2016

Accurate spectral testing without accurate instrumentation

Li Xu
Iowa State University

Recommended Citation

Xu, Li, "Accurate spectral testing without accurate instrumentation" (2016). *Graduate Theses and Dissertations*. 15165.
https://lib.dr.iastate.edu/etd/15165

Accurate spectral testing without accurate instrumentation

by

Li Xu

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
Ayman Fayed
Nathan Neihart
Akhilesh Tyagi

Iowa State University
Ames, Iowa
2016

DEDICATION

To my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my major professor Dr. Degang Chen for his guidance and support at Iowa State University. Dr.Chen greatly helps me in both technical and personal skill development. Without his guidance and support, I could not possibly complete this dissertation. His words of encouragement often inspire and motivate me. His advice on many aspects of life, research and work is the most valuable lesson to me.

I would also like to thank Dr. Randall Geiger, Dr. Nathan Neihart, Dr. Ayman Fayed and Dr. Akhilesh Tyagi for being part of my program of committee and for their inspiring teaching and research assistance.

In addition, I would like to thank all my past and present friends, colleagues and anyone who helped me with various aspects at Iowa State University.

Finally, I would like to thank my family for all of their love and support without which I could not have managed to complete my dissertation. I would like to thank my parents, Xiugang Xu and Caiyun Liu, for their unconditional love and support. I would like to thank my husband, Qilin Li, for his love, trust and encouragement. To my beloved daughter Claire Li, a source of unending joy and love, I would like to express my thanks for being such a good girl always cheering me up. My love for her is forever. I also would like to thank my in-laws, Qinghong Li and Yuanxian Qi, who have helped to care for Claire so that I would have additional time to work on my dissertation.

ABSTRACT

Analog-to-digital converters (ADCs) are becoming increasingly common in many systems in integrated circuits. Spectral testing is widely used to test the dynamic linearity performance of ADCs and waveform generators. With improvements in the performance of ADCs, it is becoming an expensive and challenging task to perform spectral testing using standard methods because of the requirement that the test instrumentation environment must satisfy several stringent conditions. In order to address these challenges and to decrease the test cost, in this dissertation, four new algorithms are proposed to perform accurate spectral testing of ADCs by relaxing three conditions required for standard spectral testing methods.

The first method developed is relaxing the requirements on precise control of coherent sampling and input signal amplitude. The efficiency and accuracy of this method is similar to the straightforward FFT, but it can simultaneously handle amplitude clipping and noncoherent sampling. By replacing a noncoherent and clipped fundamental with a coherent and unclipped one, correct spectral specifications can be obtained. Both simulation and measurement results validated the proposed method.

The second algorithm can simultaneously perform the linearity test and the spectral test with only one-time data acquisition. Targeted for realizing the cotest of linearity and spectral performance under noncoherent sampling and amplitude clipping, a new accurate method for identifying the noncoherent and clipped fundamental is introduced. The residue after removing the identified fundamental from raw data is used to obtain the linearity and spectral characterizations. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

The third method proposes an efficient and accurate jitter estimation method based on one frequency measurement. Applying a simple mathematical processing to the ADC output in time domain, the RMS of jitter and noise power are obtained. Furthermore, prior information of harmonics need not be known before the processing. The algorithm is robust enough that nonharmonic spurs do not affect the estimation result. Using the proposed algorithm, specifications of the ADC under test can be obtained without the jitter effect. Simulation results of ADCs with different resolutions show the functionality and accuracy of the method.

The last method is developed to accurately estimate the SNR with sampling clock jitter. This method does not require a precise sampling clock and thus reduces the test cost. The ADC output sequence is separated into two segments. By analyzing the difference of the two segments, the RMS of jitter and the noise power are estimated, and then the SNR is obtained. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

CHAPTER 1. INTRODUCTION

1.1 Motivation

ADC is one of the important categories of mixed-signal products. Accurate and efficient testing of high-performance ADCs is a challenging task in the modern industry as the speed and performance of the ADC increase dramatically. Spectral testing, also known as AC test, is related to the ADC's dynamic performance, including signal-to-noise ratio (SNR), total harmonic distortion (THD), spurious-free dynamic range (SFDR), and so on. In order to accurately perform the spectral test, the IEEE standards recommend the test setup to satisfy a list of stringent requirements. The test setup that satisfies all the conditions is referred to as the "ideal DFT (discrete Fourier transform) based test method" in this dissertation.

The challenges and test cost of the spectral testing come from the following aspects [1]: (1) low-distortion stimulus, (2) coherency between input frequency and sampling frequency, (3) stationarity of reference, and (4) clean sampling clock.

With the increase in the resolution and speed of ADCs, the test setup requirements for the ideal DFT-based test become much more stringent. A lot of effort has been spent on satisfying the test setup requirements, which increases the test time. Relaxing stringent conditions on the test setup not only decreases the test setup time but also facilitates the design of system-on-chips (SoCs) with on-chip test capability. Several spectral test methods that have been proposed relax one or more of the stringent requirements. With several methods proposed, it is not clear to the user which method provides the best results with minimal constraints.

1.2 Dissertation Organization

Four new test methods that address the aforementioned challenges to perform spectral testing are presented in this dissertation. As mentioned earlier, the methods can be applied to test high-performance ADCs or high-performance waveform generators. These methods can be used either in production testing or in BIST applications to decrease the test cost. The dissertation is arranged in the following order.

In Chapter 2, a spectral testing method based on fundamental identification and replacement is developed for the requirements on precise control of coherent sampling and input signal amplitude. The efficiency and accuracy of this method is similar to the straightforward FFT, but it can simultaneously handle amplitude clipping and noncoherent sampling. By replacing a noncoherent and clipped fundamental with a coherent and unclipped one, correct spectral specifications can be obtained. Both simulation and measurement results validated the proposed method.

In Chapter 3, an algorithm that can simultaneously perform the linearity test and the spectral test with one-time data acquisition is proposed. Targeted for realizing the cotest of linearity and spectral performance under noncoherent sampling and amplitude clipping, a new accurate method for identifying the noncoherent and clipped fundamental is introduced. The residue after removing the identified fundamental from raw data is used to obtain the linearity and spectral characterizations. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

Chapter 4 proposes an efficient and accurate jitter estimation method based on one frequency measurement. Applying a simple mathematical processing to the ADC output in time domain, the RMS of jitter and noise power are obtained. Furthermore, prior information of harmonics need not be known before the processing. The algorithm is robust enough that nonharmonic spurs do not affect the estimation result. Using the proposed algorithm, specifications of the ADC under test can be obtained without the jitter effect. Simulation results of ADCs with different resolutions show the functionality and accuracy of the method.

In Chapter 5, a method is developed to accurately estimate the SNR with sampling clock jitter. This method does not require a precise sampling clock and thus reduces the test cost. The ADC output sequence is separated into two segments. By analyzing the difference of the two segments, the RMS of jitter and the noise power are estimated, and then the SNR is obtained. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

Chapter 6 concludes the dissertation.

CHAPTER 2. EFFICIENT SPECTRAL TESTING WITH CLIPPED AND NONCOHERENTLY SAMPLED DATA

In a built-in self-test (BIST) environment, clipping is hard to avoid if one wants to obtain the full-range performance of an ADC. This could be due to several factors, such as circuit component mismatch, noise, clock jitter, and many others. Furthermore, coherency is a challenging task to implement in BIST as its accuracy requirement on the frequency of input signal is extremely high. This chapter proposes a new spectral testing method with the efficiency and accuracy similar to the straightforward FFT, but it can simultaneously handle amplitude clipping and noncoherent sampling. By replacing a noncoherent and clipped fundamental with a coherent and unclipped fundamental, correct spectral specifications can be obtained. Simulation results show the accurate spectral parameters of four ADCs with different resolutions. The proposed method is validated using the measurement result of a commercial 16-bit ADC. The algorithm error is quantitatively bounded using a theoretical analysis and also justified with the simulation result.

2.1 Introduction

Spectral testing is an important category in modern measurement technology. It is usually implemented by applying a pure sine wave to the device under test (DUT) and analyzing the spectrum of the output codes after a fast Fourier transform (FFT). The spectral parameters, including THD, SFDR, SNR, and effective number of bits (ENOB), can be obtained by analyzing the spectrum of the collected digitized samples. The spectral testing setup of ADC is shown in Figure 2.1. Sine waves are commonly used in ADC spectral testing because the quality is relatively easy to establish [1]. Filters are optionally used to reduce the noise and harmonic distortion in either the clock or the signal paths. To accurately test the spectral performance of an ADC, five conditions are needed for the conventional standard FFT testing method. First of all, the input sine wave should be pure enough to avoid having a distortion larger than the ADC. Second, the input signal should not exceed the full range (FR) of the ADC. If it exceeds the ADC's FR, the phenomenon clipping happens, which results in spurs in the spectrum and large errors in spectral performance testing. Third, the input signal is coherently sampled, which means that the sampled points are exactly an integer number of the input periods. Moreover, the number of sampling cycles and the number of total sampling periods are mutually prime to ensure phase distinction. The fourth condition is that the jitter of the clock should be low enough to prevent jitter artifacts from affecting the measured noise floor. Finally, the data window should be long enough, and this is easy to satisfy in most testings.

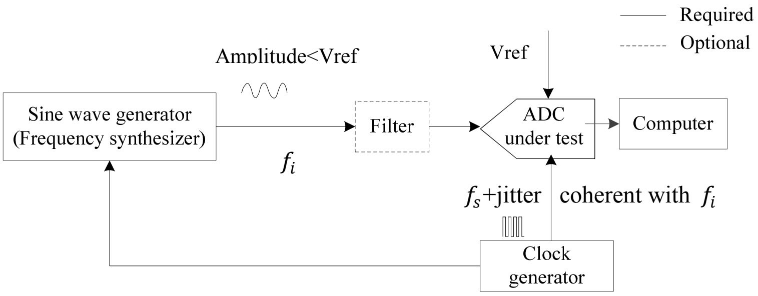


Figure 2.1 Testing setup

Built-in self-test (BIST) has been receiving attention in recent years [2–4] as it offers the stimulus and response verification capabilities for an on-chip test. This reduces the test cost and test time, as well as satisfies customers' demand for higher performances of self-calibration. In BIST ADCS, controlling the frequency and amplitude of the input sine wave accurately is challenging. Traditionally, a frequency synthesizer is used to generate a desired frequency, which in itself is a challenging task in a current on-chip design. In the other aspect, it is well-known that harmonics values are high-order nonlinear functions of the input amplitude. An ideal testing is to let the input signal cover exactly the full range of the ADC under test, and the amplitude of the input sine wave is equal to half of the ADC full scale (FS). In a bench test, precise and high-cost instrumentation is applied to adjust the input amplitude to approximately cover the full range of ADC without the clipping effect. Nevertheless, it is hard to implement on BIST. Deviation of the input amplitude and the FS of the ADC leads to either the input amplitude being much smaller than the ADC's half FS, which causes the testing result not to be accurate, or the input clipped by the ADC, which causes spurs in the spectrum or testing result to be totally incorrect. An alternative approach to get the accurate spectral testing result is to apply an input covering slightly more than the FR of the ADC and eliminate the clipping effect to get the correct spectrum.

In the current literature, several methods have been developed to get the correct spectral parameters using noncoherent sampled data. Windowing is a widely used technique to solve the noncoherency problem [5–7]. The combined windowing technique and interpolated FFT (IpFFT) was applied to low-resolution ADC testing [8]. However, the accuracy of IpFFT and windowing is not sufficient when the resolution of ADC is high or when the noncoherency is severe. In particular, the SNR error can be large. FIR was first proposed in [9]. It solved the noncoherency problem successfully, but the computing efficiency is low. [10, 11] improved the FIR method to make it more efficient and accurate. However, the noncoherent sampling methods are not intended to solve the simultaneous noncoherency and amplitude clipping problem.

The problem of clipping was discussed in the recent past. Sine wave fitting algorithms and interpolation were often used to recover the clipped signal. A four-parameter sine fitting algorithm was presented in [12] to measure the ENOB, especially when the amplitude of the sine wave was greater than the full-scale range of the DUT. [13] proposed an improved fitting of the fundamental output sine wave by eliminating clipped samples from the LS fit. The maximum likelihood method, which inherently eliminates distortions caused by clipping and incoherence, was described in [14], but its run time is much longer than that of the LS method. The efficiency of the sine fit method was improved in [15] by doing the fitting in frequency domain rather than in time domain. In [16, 17] , the four-parameter sine fitting method was used to characterize ADCs and digital oscilloscopes for THD and ENOB. The sine fitting algorithm was accurate to estimate the fundamental and high-order harmonics information of the ADC output. However, it is an iterative Newton-Gauss algorithm, which is time-consuming when the number of points used for fitting is large. And it cannot provide a correct SFDR when a nonharmonic spur contributes to SFDR. Oversampling and the interpolation method were proposed in [9, 18]. In this algorithm, the input signal was four times oversampled, and then interpolation was applied to fit and recover the clipped samples. Additional filters and clock signals were needed for another three branches, which increase both the chip area and the computing complexity.

This chapter introduces a new accurate and efficient spectral testing method to replace the clipped and noncoherent fundamental component with an unclipped and coherent one when the ADC output is clipped and sampled noncoherently. This method can be used as a replacement of the standard FFT algorithm, which the test engineers are prevalently familiar with. Simulation and experimental results of different resolution ADCs demonstrate the accuracy of this method. The error of this algorithm increases with the clipping amount, which is justified with both theoretical analysis and simulation. The algorithm error is bounded quantitatively from mathematical derivation. As clipping can be controlled in a small range, like 1% of the ADC input range, this algorithm can obtain correct spectral parameters.

This chapter is organized as follows: Section II discusses the standard spectral testing method and the effect of clipping and noncoherency. In Section III, the proposed algorithm is described in detail to solve the clipping and noncoherency problem. Section IV presents the simulation results. Experiment results are described in Section V. Section VI discusses the accuracy of the algorithm versus the clipping amount. Finally, the chapter is summarized in Section VII.

2.2 Standard Testing and the Effects of Clipping and Noncoherency

2.2.1 Standard ADC spectral testing

Let $x(t) = A\sin(2\pi ft + \varphi) + V_{os}$ be the input signal of an ADC, where *A, f,* φ , and V_{os} are the amplitude, frequency, initial phase, and offset of the input signal, respectively, and *A* is slightly less than or equal to half of the FS of the ADC. Let *y[n]* be the analog interpretation of the digital output codes converted by the ADC:

$$y[n] = A\sin\left(2\pi\frac{f}{f_s}n + \varphi\right) + \sum_{h=2}^{H} A_h \sin\left(2\pi\frac{f}{f_s}hn + \varphi_h\right) + V_{os} + w(n) \qquad n = 0,1,2...M-1 \qquad (2.1)$$

The expression f_s is the sampling frequency, $w(n)$ is the additive noise, A_h is the h^{th} harmonic of the ADC output, *H* is the total number of harmonics, and *M* is the total number of sampling points. Three assumptions in practical ADC testing exists: (1) the setup of the ADC testing must be stationary with sufficient repeatability; (2) the ADC is "continuous," meaning that $A_h = f_h(A, f, \varphi, V_{os}, f_s)$, f_h is continuous for all *h*; and (3) spectral performance is mostly dependent on *A*, *f*, and *fs*, which means $\frac{\partial f_h}{\partial A}, \frac{\partial f_h}{\partial f}, \frac{\partial f_h}{\partial f_s}$ are significant but $\frac{\partial f_h}{\partial V_{os}} \approx \frac{\partial f_h}{\partial \varphi} \approx 0$. An example spectrum of an ADC that is under standard testing is shown in Figure 2.2.

Under coherent sampling, the window captured must be an exact integer number of periods. Let *J* be the total number of sampling periods, then *M*, *f*, *f*$_s$, and *J* are related by (2.2):

$$\frac{M \times f}{f_s} = J \qquad (2.2)$$

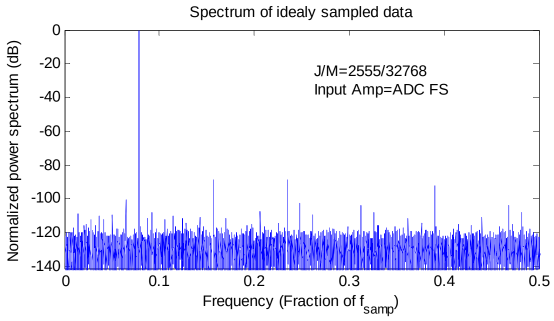


Figure 2.2 Spectrum of an ADC output under standard sampling

If *M* is chosen to be a power of 2, then any *J* that is odd is mutually prime to *M*. Equation (2.1) can be represented as

$$y[n] = A\sin\left(\frac{2\pi J}{M}n + \varphi\right) + \sum_{h=2}^{H} A_h \sin\left(\frac{2\pi hJ}{M}n + \varphi_h\right) + V_{os} + w(n) \qquad n = 0,1,2...M-1 \quad (2.3)$$

The spectral parameters can be accurately obtained by taking a discrete Fourier transform (DFT) of *M* coherently sampled data points *y[n]*.

Applying a DFT to equation (2.3), the coefficient at index *k* is given by (2.4):

$$Y[k] = \frac{1}{M}\sum_{n=0}^{M-1} y[n]e^{-j\frac{2\pi kn}{M}} \qquad k = 0,1,2...M-1 \qquad n = 0,1,2...M-1 \quad (2.4)$$

It can be seen from (2.3) and (2.4) that, for coherent sampling, the effect of noise is neglected

$$Y[J]=\frac{A}{2}e^{j\varphi}\quad Y[hJ]=\frac{A_h}{2}e^{j\varphi h} \tag{2.5}$$

The power of input signal and high-order harmonics can be estimated as

$$P_{sig}=2\left|Y[J]\right|^2 \qquad P_h=2\left|Y[hJ]\right|^2 \tag{2.6}$$

For other *k, Y[k]* represents noise as it is neither fundamental nor harmonic, the total power of noise can be calculated as

$$P_{noise}=\sum_{\substack{k=2\\ k\neq J,hJ\\ h=2,\ldots,H}}^{M}\left|Y[k]\right|^2 \tag{2.7}$$

Spectral performance parameters can be calculated as

$$SNR=10\log_{10}\left(\frac{P_{sig}}{P_{noise}}\right)\qquad THD=10\log_{10}\left(\frac{\sum_{h=2}^{H}P_h}{P_{sig}}\right)\qquad SFDR=10\log_{10}\left(\frac{P_{sig}}{2\times\max\limits_{\substack{k=2,\ldots M/2\\ k\neq J}}\left(\left|Y[k]\right|^2\right)}\right)$$
$$SINAD=10\log_{10}\left(\frac{P_{sig}}{\sum_{h=2}^{H}P_h+P_{noise}}\right)\qquad ENOB=\frac{SINAD-1.76}{6.02} \tag{2.8}$$

where SINAD is the signal-to-noise and distortion ratio. From (2.8), parameters such as SNR, THD, SFDR, and ENOB can be obtained from the spectrum of the ADC output.

2.2.2 Issues with noncoherency and clipping

1) Clipping

As mentioned in Section I, from the point of view of the amplitude requirement, ideally, we hope the input signal exactly covers the FR of the ADC or is slightly less in the bench test. However, it is extremely hard to be implemented in BIST for (1) the ADCs' full-scale range may vary slightly from one ADC to another, (2) when the sine source is fixed, the amplitude can change because of the nonstationary, and (3) the amplitude may not be equal to the designed value.

Under these limits, if we apply an input with an amplitude that is slightly less than the half FS of the ADC, as both the input amplitude and the ADC FS vary, input between the minimum possible value of the ADC FS and the maximum possible value of the ADC input could cause the input to exceed the FS range of ADC, which is shown in Figure 2.3(a).

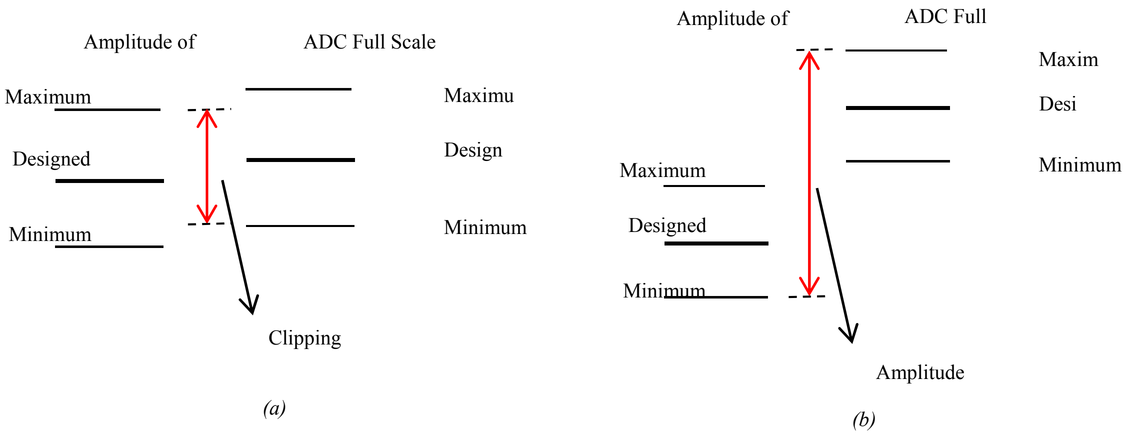


Figure 2.3 Deviation of input amplitude and ADC FS

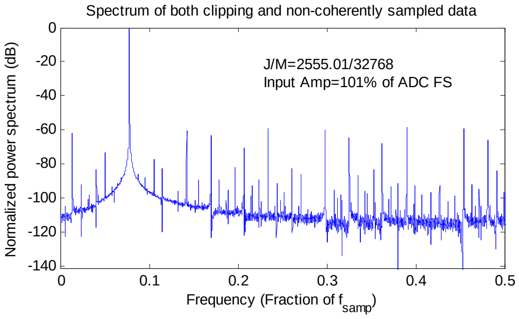


Figure 2.4 Spectrum of an ADC output that is sampled noncoherently and with clipping

In the practical testing setup of BIST, smaller input is applied to avoid the clipping effect. Nevertheless, as shown in Figure 2.3 (b), variations between the input and the FS of ADC can be large. As illustrated in Section I, harmonics are a high-order nonlinear function of the input amplitude. If the input amplitudes are different, the corresponding spectral parameters obtained are various. In this case, the nonlinearity of the gap between the input and the FS of ADC is not tested.

2) Noncoherency

In practical testing, the coherent condition is extremely hard to achieve. For instance, when f_s is 10 MHz, M is 2^14, and J is 2555, the input frequency must be 1559.4482421875 kHz. A frequency synthesizer is used to generate the accurate input frequency, which satisfies equation (2.1) for a given M, J, and sampling frequency, which is very challenging and unpractical for a BIST. If the coherent requirement is not satisfied, J is not an integer in the above equations such as (2.2) and (2.3). Let J=J_{int} + δ, where J_{int} is the integer part of J, δ is the fractional part of J, and $-0.5 < \delta \le 0.5$.

We all know that even ppm-level noncoherency can cause frequency leakage in which some fundamental energy leaks into neighbor frequencies. Figure 2. 4 shows a spectrum when there are both fundamental skirting and higher harmonics' spurs caused by both noncoherency and clipping. One cannot compute spectral performance from the spectrum directly.

As described above, for BIST, applying a relatively small input can lead the testing result to be not accurate, while a large input leads to clipping, and coherency is a challenging and costly task. In this chapter, a pure sine wave is still applied to ADC testing. The jitter noise of the clock is controlled in a certain small range, and the number of captured points is larger, but the other two requirements of coherency and unclipping are being relaxed.

2.3 The Proposed Algorithm

In this section, a new method to get accurate spectral parameters is presented when the input signal is clipped and noncoherently sampled. The proposed algorithm is targeted to provide the behavior and result that look like the standard FFT. From the discussion above, we know that clipping causes spurs in the spectrum and noncoherency results in power leakage in the fundamental harmonic, both of which conceal the high-order distortion. Our goal for this algorithm is to find an approach to remove the effect of power leakage and spurs from the spectrum. It has been proven that the error of harmonics in noncoherent sampling is mainly due to the noncoherency in the fundamental component [10]. Similarly, it can be assumed that the spurs in the clipping spectrum are mainly caused by the clipping of the fundamental component. The error due to the assumption is discussed in Section VI. In this method, the noncoherency and clipping problems are dealt with together rather than solving them one by one. In order to eliminate the effect caused by noncoherency and clipping, a clipped and noncoherent fundamental component, which is called nonideal fundamental in this chapter, is estimated from the sampled data sequence and replaced with a coherent and unclipped component, which is called ideal fundamental in this algorithm.

2.3.1 Fundamental identification

J is substituted to be $J_{int} + \delta$ in equation (2.3) under noncoherent sampling. Parameters A, V_{os}, φ, J_{int}, and δ need to be computed from the ADC output accurately to get the exact fundamentals. The procedure to estimate the five parameters is explained below.

1) Estimate A and V_{os}

In this algorithm, the amplitude and the offset of the fundamental harmonic are estimated from the number of clipping points [19]. Let K_t tell the number of clipping points at the top and K_b that at the bottom. Vref+ and Vref- are positive and negative reference voltages of the ADC. A and V_{os} can be estimated by the following equations:

$$\hat{A} = \frac{Vref^{+} - Vref^{-}}{\cos\left(\frac{(K_t - 1)\pi}{M}\right) + \cos\left(\frac{(K_b - 1)\pi}{M}\right)}$$

$$\hat{V}_{os} = \frac{1}{2}\left\{Vref^{+} + Vref^{-} + \hat{A}\cos\left[\left(\frac{(K_b - 1)\pi}{M}\right) - \cos\left(\frac{(K_t - 1)\pi}{M}\right)\right]\right]\right\} \tag{2.9}$$

2) Estimate J

After A and V$_{os}$ are obtained from the time domain data, J$_{int}$ and the first-time estimation of φ are obtained from the frequency domain. DFT is applied to the ADC output sequence y[n], and J$_{int}$ is estimated from the index of frequency bin, which contains the maximum power, excluding the DC component in half spectrum.

$$\hat{J}_{int} = \arg\max_{2 \le k \le M/2} |Y[k]| \tag{2.10}$$

φ is the initial phase of the fundamental component of the sampling sequence. From equation (2.5), and substituting J = J$_{int}$ + δ, the first-time estimation of φ can be computed as

$$\hat{\varphi}_1 = \begin{cases} \arctan\left(\frac{imag(Y[\hat{J}_{int}])}{real(Y[\hat{J}_{int}])}\right) & if\ real(Y[\hat{J}_{int}]) \ge 0 \\ \arctan\left(\frac{imag(Y[\hat{J}_{int}])}{real(Y[\hat{J}_{int}])}\right) + \pi & if\ real(Y[\hat{J}_{int}]) < 0\ and\ imag(Y[\hat{J}_{int}]) > 0 \\ \arctan\left(\frac{imag(Y[\hat{J}_{int}])}{real(Y[\hat{J}_{int}])}\right) - \pi & if\ real(Y[\hat{J}_{int}]) < 0\ and\ imag(Y[\hat{J}_{int}]) < 0 \end{cases} \tag{2.11}$$

3) Estimate δ and the fine estimation of φ

After A, V$_{os}$, J$_{int}$, and initial φ were obtained from steps 1 and 2, we apply a fine estimation of φ and δ using least squares method to zero-crossing points of the ADC output in time domain.

The approximate fundamental phase at each point can be described as

$$\Phi_{yf}[n] = \frac{2\pi \hat{J}_{\text{int}}}{M} n + \hat{\varphi}_1 \tag{2.12}$$

However, as the phase is computed directly from the noncoherent data, it does not represent the actual phase of the input sine wave or the phase of the first sampled point. In addition, δ needs to be calculated to get the actual frequency. In this method, the actual frequency and phase of the input sine wave are corrected from the set of zero-crossing points of the sampled points after the offset is removed.

There are several approaches to selecting the fitting set of points, and the one chosen in this method is defined by

$$\left| \frac{y[k]}{A} \right| < 0.1 \tag{2.13}$$

Neglecting high-order distortion and noise, the actual phase of those points can be described as

$$\Phi_{act}[k] = \frac{2\pi J}{M} k + \varphi \qquad k \in I \tag{2.14}$$

I is the set of all *k*'s that satisfy (2.13). The difference between the actual phase and the fundamental phase gives

$$\Phi_{act}[k] - \Phi_{yf}[k] = \varphi_0 + \frac{2\pi\delta}{M} k \tag{2.15}$$

where φ_0 is the difference of the actual initial phase with the first estimation and δ is the fraction of the number of periods.

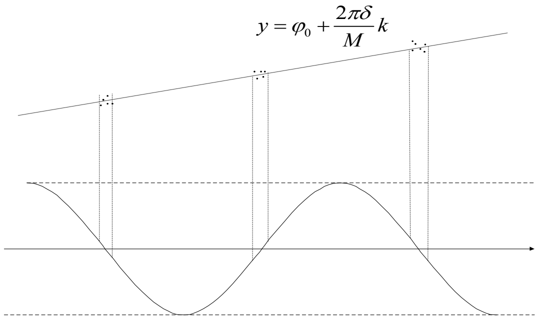


Figure 2.5 Least squares fitting line of the phase difference

Fitting $(\Phi_{act}[k]-\Phi_{yf}[k])$ with the index of k to a line using the least squares method, the estimated $\hat{\varphi}_0$ and $\hat{\delta}$ can be obtained from it. As shown in Figure 2.5, the sine wave consists of all the output points; the points on the straight line represent the phase difference of those zero-crossing points.

Then φ and J are obtained

$$\hat{\varphi}=\hat{\varphi}_1+\hat{\varphi}_0 \qquad \hat{J}=\hat{J}_{\text{int}}+\hat{\delta} \tag{2.16}$$

4) Fundamental error

From the parameters obtained from the previous the steps, the noncoherent fundamental harmonic is written as

$$yf_{nco}[n]=\hat{A}\sin\left(\frac{2\pi\hat{J}}{M}n+\hat{\varphi}\right) \tag{2.17}$$

Actually, as long as the testing is not coherent, the noncoherent fundamental component cannot be removed thoroughly as calculated in (2.17). Under this condition, there is still the fundamental effect in the residue if only (2.17) is removed from the output sequence. Assume that the error of the fundamental component can be written as

$$yfe[n] = amp_e \cdot \cos\left(\frac{2\pi\hat{J}}{M}n + \varphi_e\right) \tag{2.18}$$

The expression of amp_e and φ_e can be obtained from Ref [20] (equations (22) and (23)).

2.3.2 Clipping the fundamental component

The nonideal fundamental component is constructed of two parts: the noncoherent fundamental component and the fundamental error. Being clipped by the full scale of the ADC, the nonideal fundamental is represented as follows:

$$yf_{nid}[n] = \begin{cases} yf_{nco}[n] + yfe[n] & Vref^- < yf_{nco}[n] + yfe[n] < Vref^+ \\ Vref^+ & yf_{nco}[n] + yfe[n] \geq Vref^+ \\ Vref^- & yf_{nco}[n] + yfe[n] \leq Vref^- \end{cases} \tag{2.19}$$

2.3.3 Constructing the new sequence

When the nonideal fundamental component represented as (2.19) is subtracted from the sampling points, only the harmonics and noise distortion without noncoherence and clipping remain in the residue. We can add an ideal fundamental to the residue to get a new sequence.

The ideal fundamental component is represented as (2.20) with parameters obtained from 2.3.1.

$$yf_{id}[n] = \hat{A}\sin\left(\frac{2\pi\hat{J}_{\text{int}}}{M}n + \hat{\varphi}\right) \tag{2.20}$$

After replacing the noncoherent and clipped fundamental component with an ideal one, the new sequence is used to analyze the spectral performance of the ADC.

$$\hat{y}[n] = y[n] - yf_{nid}[n] + yf_{id}[n] \tag{2.21}$$

The proposed algorithm can be described as follows:

1) Collect M points of ADC output y[n]; M is power of 2.

2) Count the number of clipped points; estimate A and V_{os} using equation (2.9).

3) Perform DFT on y[n], get Y[k], calculate J_{int}, and initial estimate φ using (2.10) and (2.11).

4) Apply the LS method on zero-crossing points in (2.13). Get estimates of the exact J and φ from (2.16).

5) Calculate the fundamental error using equation (2.18).

6) Clip the noncoherent fundamental, and get an unideal fundamental (2.19).

7) Construct an ideal fundamental (2.20), and get the new sequence (2.21).

8) Perform DFT to the new sequence to get a spectral performance.

Remark: In the range of clipping amount discussed in this chapter, the influence of clipping on signal power and harmonics calculation is negligible, but its influence on noise is significant. The total noise power calculated from step 8 needs to be scaled up by multiplying $\frac{M}{M-(K_t+K_b)}$.

2.4 Simulation Result

2.4.1 Functionality

The proposed method of spectral testing with clipped and noncoherently sampled data has been validated to be true by simulation data generated in MATLAB. A 12-bit ADC with INL of 0.6 LSB, 14-bit with INL of 0.5 LSB, 16-bit with 0.9 LSB, and 18-bit ADC with 1.2 LSB were modeled in MATLAB, respectively. The additive noise for each ADC is at the quantization noise level. Two kinds of input sine wave were converted into digital codes by each ADC under test, the one with the amplitude slightly less than half of the FS of the ADC and other one with an amplitude that covers 1% more than the FR of the ADC. As space limitations, we only show the spectra of the testing results for the 18-bit ADC, while the test results of other ADCs are shown in Table 2.1.

Table 2.1 Spectral parameters' testing results of the proposed algorithm and standard method

	12-bit ADC	**14-bit ADC**	**16-bit ADC**	**18-bit ADC**
THD_id(dB)	–80.4	–97.59	–99.74	–118.39
THD_pro(dB)	–81.24	–98.12	–99.44	–116.22
THD_err(dB)	–88.0	–106.9	–111.2	–120.3
SFDR_id(dB)	82.95	100.11	101.25	119.54
SFDR_pro(dB)	82.92	101.39	101.99	117.13
SFDR_err(dB)	–105.2	–106.1	–109.3	–120.8
SNR_id(dB)	72.04	82.44	94.71	105.94
SNR_pro(dB)	72.36	83.10	94.06	106.01
SNR_err(dB)	–83.4	–91.0	–102.6	–124
ENOB_id(bit)	11.57	13.38	15.24	17.26
ENOB_pro(bit)	11.64	13.49	15.15	17.25
ENOB_err(bit)	0.07	0.11	–0.09	–0.01

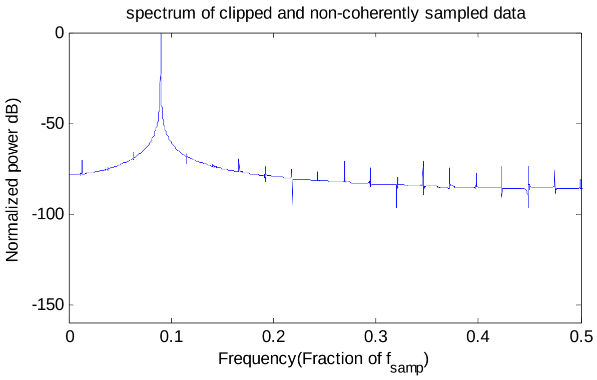


Figure 2.6 Spectrum of simulation result: direct FFT on clipped and noncoherently sampled data

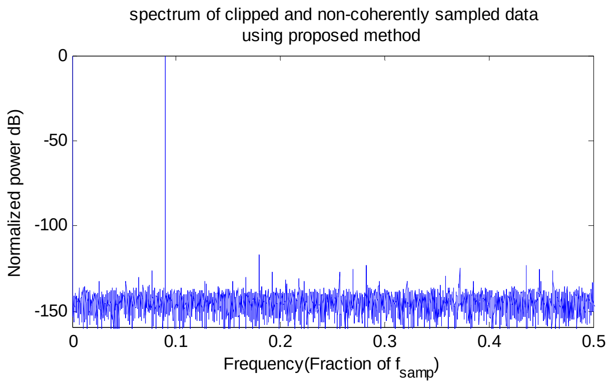


Figure 2.7 Proposed algorithm on clipped and noncoherently sampled data

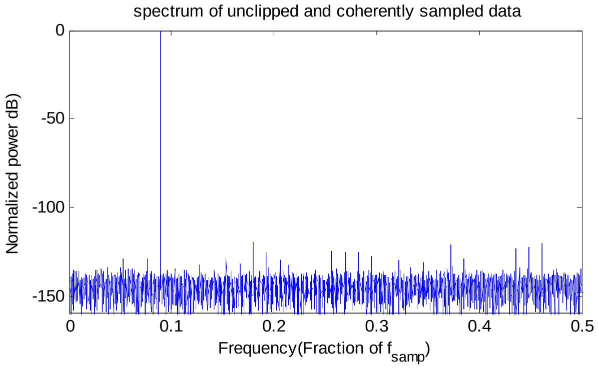


Figure 2.8 Spectrum of simulation result: direct FFT on unclipped and coherently sampled data

Figure 2.6 shows the spectrum of direct FFT applied to the clipping and noncoherently sampled data. Spectral parameter values, including SNR, ENOB, THD, and SFDR, directly obtained from this spectrum are totally incorrect, which should be discarded. The spectrum of the clipped and noncoherently sampled data using the proposed method is shown in Figure 2.7, and the spectrum of straightforward FFT applied to the standard sampling data is shown in Figure 2.8. It can be seen from Figures 2.7 and 2.8 that the proposed method is capable of removing the clipping and power leakage effect. Standard algorithm can be used to obtain ADC specifications from the spectrum recovered by the proposed method.

The measurement results of THD, SFDR, SNR, and ENOB and the corresponding errors are shown in Table 2.1. In this table, X_id is the result of the standard method when applying direct FFT to the sampled data with the input signal amplitude slightly less than the full range of the ADC under test and is coherently sampled. X_pro is the result after applying the proposed method to the clipped and noncoherent data. X_err is the error between X_pro and X_id for THD, SNR, and SFDR. It is calculated as the power error of X_pro and X_id, and then it is converted in dBs units. ENOB_err is just the difference between ENOB_pro and ENOB_id. It can be seen that when applying the proposed method to the clipped and noncoherently sampled data, specifications such as THD, SFDR, SNR, and ENOB can be accurately estimated.

2.4.2 Comparison with the state of the art

The four-parameter sine fit method is an accurate method to get the spectral performance with clipped and noncoherently sampled data [12,13]. However, all unclipped points in data are considered to perform nonlinear least squares used for fitting, and the initial guess of the fundamental frequency must be accurate enough to get the convergence. After identification of the fundamental, the three-parameter sine fit will be applied to get information of high-order harmonics. The fundamental identification of the proposed method can also be replaced by the four-parameter sine fit. However, FFT and least squares fitting are the two most time-consuming blocks in the proposed method, where FFT is computationally efficient as the total number is chosen to be a power of 2, and the LS here is a linear fitting to a small set of data, which consumes a small amount of time. In another aspect, the proposed method does not need iterations to get high-order harmonics, and in the four-parameter sine fit method, the computation time increases as the number of computed harmonics increases. The following table is a comparison between the proposed method and the four-parameter sine fit method (using MATLAB on a 64-bit Intel Core i5 CPU with 4 GB memory). The proposed method is accurate and more efficient than the four-parameter sine fitting method. The four-parameter sine fit method is also accurate on the estimation of SNR, THD, and ENOB, but it cannot estimate the correct SFDR value when SFDR is not determined by a harmonic spur.

Table 2.2 Comparison of calculation time (16-bit ADC, INL = 0.9 LSB)

M	Proposed method	Four-parameter sine fit
2^{13}	0.003s	>0.016s*

*. Time taken to identify the fundamental. The three-parameter sine fit needs to be applied to calculate each harmonic.

2.5 Experiment Results

The proposed method of spectral testing with clipped and noncoherently sampled data has been validated to be true by simulation data generated in MATLAB. But in real testing, there are more unpredictable factors that cannot be simulated, and the ultimate application of this algorithm is experimental testing. This proposed method is validated by measurement data using a 16-bit SAR ADC. Two series of data were collected in a bench testing lab at the High-Precision Analog Division of Texas Instruments. One series is collected under the standard requirement: coherently sampled, the amplitude of the input signal is slightly less than the full range of the ADC. The values of THD, SFDR, SNR, and ENOB are used as reference values to verify the proposed method. The other series of data are collected noncoherently and with clipping (δ= 0.155, clipping amount = 0.6%). Both direct FFT and the proposed method were applied to the series of data.

All three spectra are shown in Figure 2.9. The proposed method is capable of removing the skirting and clipping effect and producing a spectrum that is essentially the same as that of the standard sampling data. The parameters of THD, SFDR, SNR, and ENOB are shown in Table 2.3. It can be concluded that all errors of THD, SFDR, and SNR of the proposed method are small enough to be neglected. Hence, the condition for coherency and clipping is completely eliminated using the proposed method.

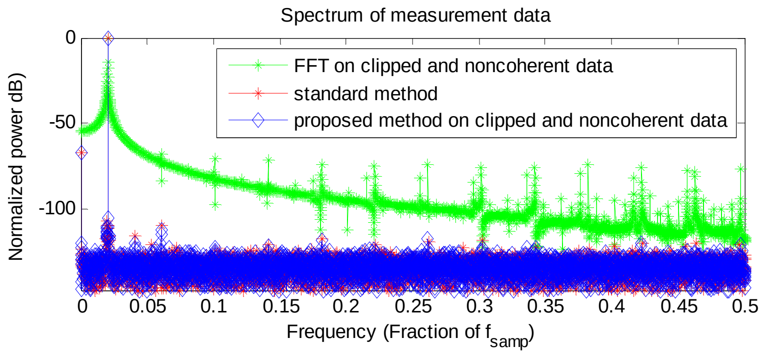


Figure 2.9 Spectrum of measurement results: (1) direct FFT on clipped and noncoherently sampled data (green), (2) proposed algorithm on clipped and noncoherently sampled data (blue), and (3) standard method (red)

Table 2.3 Testing results of a 16-bit ADC

THD_id(dB)	–105.78	SNR_id(dB)	95.84
THD_pro(dB)	–105.73	SNR_pro(dB)	95.82
THD_err(dB)	–125.14	SNR_err(dB)	–119.59
SFDR_id(dB)	109.70	ENOB_id(bit)	15.56
SFDR_pro(dB)	109.91	ENOB_pro(bit)	15.55
SFDR_err(dB)	–122.89	ENOB_err(bit)	–0.01

2.6 Error versus the Clipping Bound

In the proposed algorithm, we mentioned that a coherent and unclipped fundamental sine wave was generated to replace the noncoherent and clipped fundamental component. The method to deal with the noncoherency has been validated [12]. The relationship of the algorithm error and the clipping bound is discussed in this section. In the analysis, an input sine wave with the amplitude equal to half of the FS of the ADC is taken as the reference. The difference between the testing result of a clipped input signal and the reference is discussed with a different clipping amount.

Let *x(t)* be the input signal of an ADC, and at $t=nT_s$, the input voltage is $V_n = x(nT_s)$. The actual output code corresponding to V_n is described in (2.22), which contains two parts: the code error and the ideal output if there is no nonlinearity in the ADC under test:

$$\begin{aligned} C_n &= e(C_n) + round(G(V_n + w_n + D_{os})) \\ &= e(C_n) + G(V_n + w_n + D_{os}) + Q_n \end{aligned} \tag{2.22}$$

in which w_n is the additive noise at time nT_s, G and D_{os} are the gain and offset of the testing ADC, *round(x)* is the ideal output code of the ADC with the input voltage x, $e(C_n)$ is the code error at code C_n caused by the nonideality of the ADC, and Q_n is the quantization error. For a given input signal, we can get a sequence $e(C_n)$, and it can be assumed that the relationship between the error code and the input voltage is fixed for a certain ADC.

Assume that an input sine wave with the amplitude A equal to the ADC's half FS is taken as the reference signal. When it is unfolded to half period, the expression can be written in (2.23), and the unfolded wave is shown as the solid line in Figure 2.10.

$$V_n = A\sin(\omega n T_s) \qquad n = -\frac{M}{2}, -\frac{M}{2}+1, \dots, \frac{M}{2}-1 \tag{2.23}$$

In this section, the sample index n is from $-\frac{M}{2}$ to $\frac{M}{2}-1$ to get the phase from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ when all points are unfolded to half period. And under this condition, $\omega T_s = \frac{\pi}{M}$.

In the other case, the signal with amplitude slightly larger than the full scale of the ADC is

$$V_n' = (1+\varepsilon)A\sin(\omega nT_s) \qquad n = -\frac{M}{2}, -\frac{M}{2}+1, \ldots, \frac{M}{2}-1 \tag{2.24}$$

ε is the overdrive ratio. Plug these two equations, (2.23) and (2.24), into the representation (2.22), and we get the code error sequences in both cases, which are

$$C_n = e(C_n) + G(A\sin(\omega nT_s) + w_n + D_{os}) + Q_n \tag{2.25}$$

$$C_n' = e(C_n') + G((1+\varepsilon)A\sin(\omega nT_s) + w_n' + D_{os}) + Q_n' \tag{2.26}$$

where $e(C_n)$ and $e(C_n')$ are shown in Figure 2. 10. It can be seen that for the clipped segment, $e(C_n')$ is all 0 as the input is clipped at the top or the bottom, where the output of the ADC is either 0 or 2^N–1 (*N* is the resolution of the ADC). In the proposed algorithm, an identified and clipped sine wave is subtracted from the clipped output codes. We assume that the sine wave is identified correctly. After the subtraction, the code error is given as

$$\Delta C_n' = C_n' - G((1+\varepsilon)A\sin(\omega nT_s) + D_{os}) = e(C_n') + Gw_n' + Q_n' \tag{2.27}$$

The code error for the reference signal is

$$\Delta C_n = e(C_n) + Gw_n + Q_n \tag{2.28}$$

In order to see how much error the clipping algorithm has, one can subtract the ideal error code from the clipped error code.

$$\Delta = \Delta C_n' - \Delta C_n = (e(C_n') - e(C_n)) + G(w_n' - w_n) + (Q_n' - Q_n) \tag{2.29}$$

When we do spectrum analysis, we want to know how much harmonic does this error cause. The second and third terms in (2.29) are additive noise and quantization error, which can be assumed as white, and are in the noise floor when doing DFT, so these two parts can be ignored.

The code error can be represented into a smooth component *g(Cn)* plus a quantization component QE:

$$e(C_n) = g(C_n) + QE \tag{2.30}$$

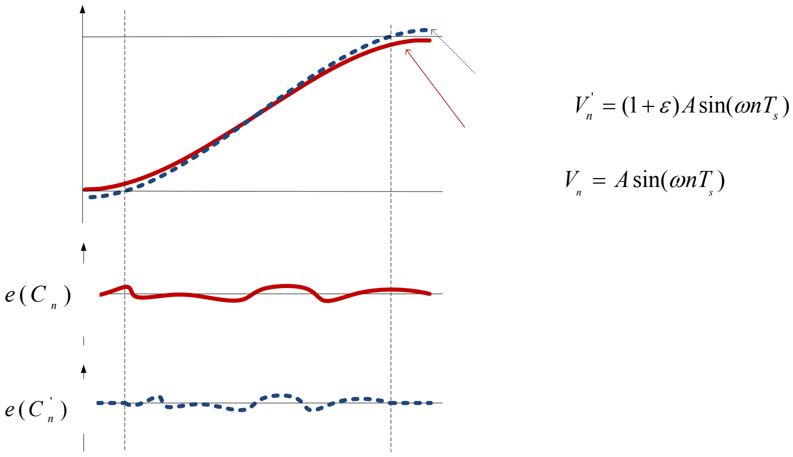
$e(C_n^{'})$ and $e(C_n)$ are similar to each other if the clipping amount is small; the Taylor series expansion to $e(C_n^{'})$ at code *Cn* is

$$e(C_n^{'}) = g(C_n) + g^{'}(\xi)\cdot(C_n^{'} - C_n) + QE \tag{2.31}$$

where ξ is a value between $C_n^{'}$ and C_n.

Plug $e(C_n^{'})$ and $e(C_n)$ to the equation of Δ, and now (2.29) can be written as

$$\Delta = \begin{cases} -e(C_n) & V_n^{'}\ is\ clipped \\ g^{'}(\xi)\cdot(C_n^{'} - C_n) & V_n^{'}\ is\ not clipped \end{cases} \tag{2.32}$$



$V_n^{'} = (1+\varepsilon)A\sin(\omega n T_s)$
$V_n = A\sin(\omega n T_s)$
$e(C_n)$
$e(C_n^{'})$

Figure 2.10 Different inputs and the code error

The power of Δ is calculated as

$$\begin{aligned} P(\Delta) = \sum_{n=0}^{M-1}\left|\Delta(n)\right|^2 &= \sum_{clipped} e^2(C_n) + \sum_{nonclipped}\left[g^{'}(\xi)\cdot(C_n^{'} - C_n)\right]^2 \\ &\leq \left[\left| e(C_n)\right|_{\max}\right]^2\cdot(K_t + K_b) + \left[\left| g^{'}(\xi)\right| \max\right]^2\cdot\sum_{n=0}^{M-1}(C_n^{'} - C_n)^2 \end{aligned} \tag{2.33}$$

where Kt and Kb are the number of clipped points at the top and bottom, respectively, as mentioned in Section III. If the clipping ratio ε is 1%, $(K_t + K_b)$ is 9% of M. As $[| e(C_n) |_{\max}]^2$ is at the THD level, the first term in (2.33) $[| e(C_n) |_{\max}]^2 \bullet (K_t + K_b)$ is about 10 dB lower than THD. The value of the first term changes as the clipping ratio varies, but the quantitative value is fixed for a given ADC and the clipping ratio.

As $g(C_n)$ is a smooth component of the code error at C_n, the magnitude of $g(C_n)$ is at THD level. The second term in (2.33) can be written as

$$\begin{aligned} \left[| g^{'}(\xi) | \max\right]^2 \bullet \sum_{n=0}^{M-1} (C_n^{'} - C_n)^2 &= \left[| g^{'}(\xi) | \max\right]^2 \cdot \sum_{n=0}^{M-1} ((1+\varepsilon) A \sin(\omega n T_s) - A \sin(\omega n T_s))^2 \\ &= \left[| g^{'}(\xi) | \max\right]^2 \cdot \sum_{n=0}^{M-1} (\varepsilon A \sin(\omega n T_s))^2 \\ &\leq THD \omega T_s \cdot \varepsilon^2 \omega^2 T_s^2 \sum_{n=0}^{M-1} n^2 \\ &\approx THD \cdot \varepsilon^2 \pi^2 \end{aligned} \qquad (2.34)$$

Take $\varepsilon = 1\%$, for example. The second term in $P(\Delta)$ is 30 dB lower than THD, which is much smaller than the first term and can be neglected. It can be concluded that the algorithm error is mostly from the clipped samples, and the error is quantitatively bounded for a given clipping amount.

This is also verified with the simulation results using the 16-bit ADC modeled in MATLAB described in Section IV. A reference input signal is generated with the amplitude slightly less than the full range of the ADC, and it is sampled coherently by the ADC. All harmonics' values are obtained by spectral analysis. Then a tested input signal with the amplitude overrange ε is sampled by the ADC. ε is from 0% to 5%. The distortion power difference between the tested signal and the reference signal are shown in Figure 2.11, where the value of the red line is obtained from equation (2.33). As the clipping amount increases, the bound of harmonics' error increases, and the error is bounded by equation (2.33). As the clipping amount can be controlled under a certain level, for example, 1%, the algorithm is accurate enough to obtain the correct spectral parameters.

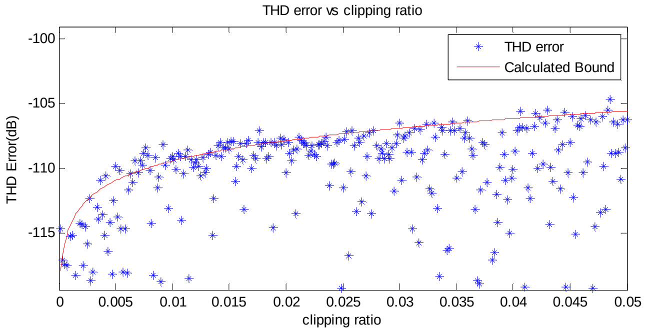


Figure 2.11 Algorithm error at different clipping amounts

2.7 Conclusion

An accurate and efficient algorithm was proposed for ADC spectral testing that allows an amplitude that is larger than a half FS and does not require coherency. This method relaxed two strict conditions in the standard spectral test and thereby makes the accurate spectral testing in BIST possible. Both simulation and experiment results showed that the proposed method is accurate when the output data are slightly clipped and noncoherently sampled. The measured parameters THD, SFDR, SNR, and ENOB were comparable to those from the standard testing. Theoretical derivation showed that the algorithm error increases with the clipping amount, and it was also validated by simulation results. Although the algorithm is not accurate when the clipping amount is large, severe clipping rarely occurs in testing. In future work, this method needs to be modified to get correct spectrum parameters when the clipping is severe.

References

[1] "IEEE Standard for Terminology and Test Methods for Analg-to-Digital Converters," IEEE Std.1241, 2010.

[2] X. Hanqing, J. Hanjun, C. Degang, and R. L. Geiger, "High-Resolution ADC Linearity Testing Using a Fully Digital-Compatible BIST Strategy," IEEE Transactions on Instrumentation and Measurement, vol. 58, pp. 2697-2705, 2009.

[3] D. Jingbo and C. Degang, "SNR measurement based on linearity test for ADC BIST," IEEE International Symposium on Circuits and Systems (ISCAS), 2011, pp. 269-272.

[4] D. Jingbo, C. Degang, and R. Geiger, "A low cost method for testing offset and gain error for ADC BIST," IEEE International Symposium on Circuits and Systems (ISCAS), 2012, pp. 2023-2026.

[5] P. Carbone, E. Nunzi, and D. Petri, "Windows for ADC dynamic testing via frequency-domain analysis," IEEE Transactions on Instrumentation and Measurement, vol. 50, pp. 1571-1576, 2001.

[6] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," Proceedings of the IEEE, vol. 66, pp. 51-83, 1978.

[7] D. Belega, M. Ciugudean, and D. Stoiciu, "Choice of the cosine-class windows for ADC dynamic testing by spectral analysis," Measurement, vol. 40, pp. 361-371, May 2007.

[8] D. K. Mishra, "ADC testing using interpolated fast Fourier transform (IFFT) technique," International Journal of Electronics, vol. 90:7, pp. 459-469, 2003.

[9] Y. Zhongjun, C. Degang, and R. Geiger, "A computationally efficient method for accurate spectral testing without requiring coherent sampling," International Test Conference(ITC) . 2004, pp. 1398-1407.

[10] S. Sudani, W. Minshun, and C. Degang, "A novel robust and accurate spectral testing method for non-coherent sampling," IEEE International Test Conference (ITC), 2011, pp. 1-10.

[11] W. Minshun and C. Degang, "A faster method for accurate spectral testing without requiring coherent sampling," IEEE Instrumentation and Measurement Technology Conference (I2MTC) , 2011.

[12] N. Giaquinto and A. Trotta, "Fast and accurate ADC testing via an enhanced sine wave fitting algorithm," IEEE Transactions on Instrumentation and Measurement, vol. 46, pp. 1020-1025, 1997.

[13] I. Kollar and J. J. Blair, "Improved determination of the best fitting sine wave in ADC testing," IEEE Transactions on Instrumentation and Measurement, vol. 54, pp. 1978-1983, 2005.

[14] J. Šaliga, I. Kollár, L. Michaeli, J. Buša, J. Lipták, and T. Virosztek, "A comparison of least squares and maximum likelihood methods using sine fitting in ADC testing," Measurement, Volume 46, Issue 10, December 2013, pp. 4362–4368.

[15] V. Palfi and I. Kollar, "Acceleration of the ADC Test With Sine-Wave Fit," IEEE Transactions on Instrumentation and Measurement, vol. 62, pp. 880-888, 2013.

[16] T. Z. Bilau, T. Megyeri, A. Sárhegyi, J. Márkus, and I. Kollár, "Four-parameter fitting of sine wave testing result: iteration and convergence," Computer Standards & Interfaces, vol. 26, pp. 51-56, 2004.

[17] C. Kui Fu, "Estimating Parameters of a Sine Wave by Separable Nonlinear Least Squares Fitting," IEEE Transactions on Instrumentation and Measurement, vol. 59, pp. 3214-3217, 2010.

[18] M. Allén, J. Marttila, and M. Valkama, "ITERATIVE SIGNAL PROCESSING FOR MITIGATION OF WIDEBAND ADC NONIDEALITIES IN COGNITIVE RADIO RECEIVER," 19th European Signal Processing Conference, 2011.

[19] J. Blair, "Histogram measurement of ADC nonlinearities using sine waves," IEEE Transactions on Instrumentation and Measurement, vol. 43, pp. 373-383, 1994.

[20] V.K.Jain, W.L.Collins, and D.C.Davis, "High-Accuracy Analog Measurements via Interpolated FFT " IEEE Transactions on Instrumentation and Measurement, vol. 28, pp. 113-122, 1979.

CHAPTER 3. FAST COTEST OF LINEARITY AND SPECTRAL PERFORMANCE WITH NONCOHERENT SAMPLED AND AMPLITUDE CLIPPED DATA

Production test is a significant contributor to the manufacturing cost for high-performance analog and mixed-signal products. The linearity test and the spectral test are two main categories in ADC testing, and the linearity test cost is usually the largest component in the test cost. For spectral testing, it is a very challenging task to precisely control the amplitude and frequency of the input sinusoidal signal. Overrange amplitude results in clipping ADC output, and noncoherent sampling results in spectral leakage. To reduce the ADC test cost dramatically, a new algorithm is proposed in this chapter. The new algorithm can simultaneously perform the linearity test and the spectral test with only one-time data acquisition. Targeted for realizing the cotest of linearity and spectral performance under noncoherent sampling and amplitude clipping, a new accurate method for identifying the noncoherent and clipped fundamental is introduced. The residue after removing the identified fundamental from raw data is used to obtain the linearity and spectral characterizations. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

3.1 Introduction

The ADC is an important category in today's semiconductor industry. Linearity testing and spectral testing of high-performance ADCs are two well-known challenging aspects in production test, and the test cost is meaningful to manufactures due to the large volume. The linearity test is related to the ADC's static characteristics, such as integral nonlinearity (INL) and differential nonlinearity (DNL), where spectral testing of the ADCs is also called AC testing and includes testing of the ADCs' dynamic specifications, such as THD, SFDR, SNR, and so on. The linearity test is conventionally tested by the histogram method using either a pure sine wave or a very linear ramp or triangle wave as stimulus; whereas spectral performance is tested by the fast Fourier transform (FFT) method using a pure sine wave as input [1].

The challenges and cost in accurate ADC testing come from two aspects: (1) testing setup and (2) data acquisition. In the testing setup, there are several challenging requirements [1, 2]: (1) the stimulus should be three to four bits more linear than the ADC under test; (2) the sampling clock timing must be precise, which means that the clock jitter should be low; (3) the sampling frequency and the stimulus frequency should be set to achieve coherency, which means that the collected data must have exactly integer cycles of input signal; and (4) the input amplitude is set to let the input signal be in the ADC's full range. Usually, it is challenging to achieve precise control over the frequency and amplitude of the tested sinusoidal input signal. Either the instrumentations are expensive for precise control, or the adjusting is time-consuming to satisfy the ideal testing requirements. And data acquisition is more time-consuming in linearity testing than in spectral testing. The histogram test is the standard and conventional test method for the linearity test [3–5] . It usually collects hundreds of hits per code to calculate INL and DNL. The data acquisition time is considerably large for high-resolution ADCs.

Researchers have worked on methods to relax the requirements described above to reduce the test cost. For spectral testing, methods have been proposed to obtain accurate spectral specifications with relaxed requirements. Several accurate and robust methods have been developed to deal with the noncoherency issue [6, 7]. Li et al. proposed an efficient spectral testing method with noncoherent and clipped data based on fundamental replacement [8, 9]. However, the accuracy and clipping range is limited. Siva et al. proposed an accurate full-spectrum test method, FERARI, which is robust to simultaneous noncoherent sampling and amplitude clipping [10]. This method is accurate and robust in spectral testing, but the estimation accuracy of the fundamental is limited, and it cannot be used in the linearity test.

Many researchers have proposed methodologies to reduce ADC linearity test time. Overall, linearity test results are obtained from combining low-code and high-code frequency tests [11]. The paper in ITC 2012 [12] proposed an efficient linearity test method that dramatically reduces the data acquisition time by more than 100 times while maintaining the same and even better test precision. This method is efficient and accurate, but it has a requirement on the data acquisition that the input signal must be sampled coherently and without amplitude clipping. It is known that coherency is a challenging condition to achieve in production test. If the input amplitude is set to be relatively small to avoid clipping, a large number of codes can be untested.

This chapter proposes an efficient cotest method for simultaneous linearity test and spectral test. Using this method, only one sequence of ADC output data needs to be collected with about one hit per code to obtain both linearity and spectral characteristics. This method also relaxes the requirements on noncoherency and clipping. With the proposed method, the noncoherency can be at any level. The input overrange is limited to 3% of the full range of ADC. This is a valid and practical limit for production test as the amplitude of the input signal to the ADC can be controlled up to 3% without any challenges. The proposed method involves accurate estimation of the fundamental component and later subtracts the estimated fundamental from the output of the ADC to obtain the residue. On the one hand, the residue of the unclipped ADC output is used to estimate the INL and DNL of the ADC. On the other hand, the residue is interpolated to a new coherent and unclipped fundamental to obtain accurate information of the ADC's harmonics and noise.

The remainder of the chapter is presented as follows: In Section II, a brief overview of an efficient linearity test algorithm and spectral testing with noncoherent sampling and clipping is presented. In Section III, a new method is proposed that can accurately estimate both linearity and spectral characteristics when the ADC output is simultaneously clipped and noncoherently sampled. In Section IV, simulations are presented showing the accuracy and robustness of the proposed method. In Section V, measurement results are shown to validate the functionality of the proposed method, and Section VI concludes the chapter.

3.2 Problem Statement

An efficient linearity test algorithm was proposed in [12], which can maintain the same and even better test precision with 100 times less data than the conventional method. The efficient linearity test algorithm is based on an assumption that for an N-bit ADC, all the INL/DNL errors are highly correlated and are deterministic functions of a much smaller number of independent errors (component size error, parasitic, voltage coefficients, etc.), which is true for most of the ADC architectures except for flash ADCs and sigma-delta ADCs. The INL curve was modeled as a "segmented nonparametric" model that contains the most significant bytes (MSB) segment, intermediate significant bytes (ISB) segment, and least significant bytes (LSB) segment. The algorithm and its limitations are discussed in the following part of this section.

The algorithm in [12] is described as follows:

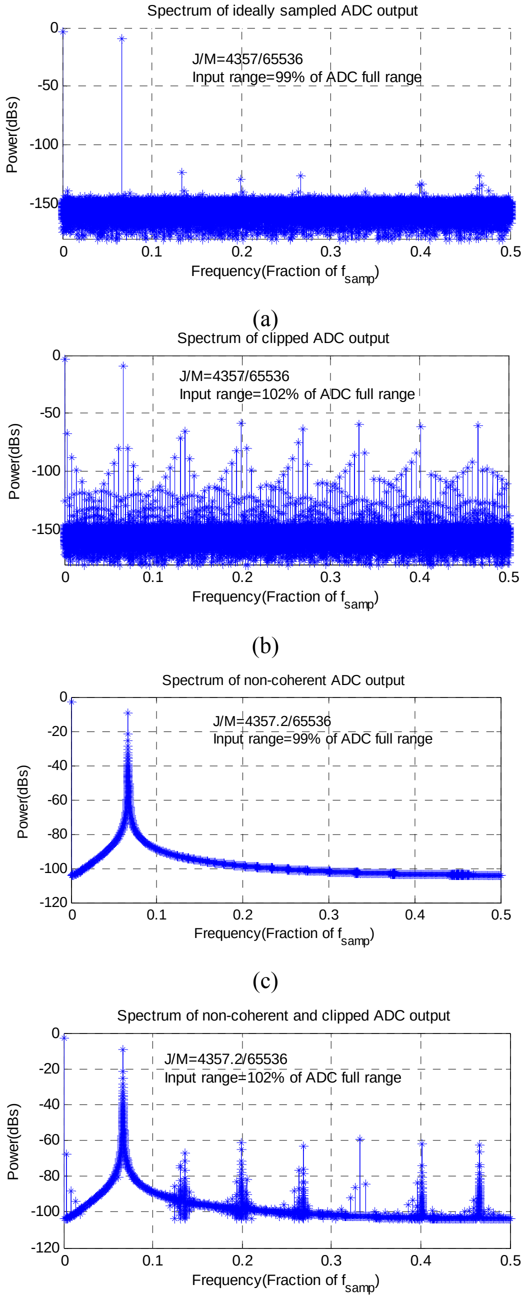
1) *With pure sine input, collect coherent ADC output data set y_n, n=1,2,3...M.*

2) *Perform FFT on y_n, obtain Y_n in the frequency domain, set the fundamental and DC component Y_{fund} and Y_{DC} to be 0, and then the frequency domain residue is Y_{en}.*
3) *Calculate the residue power P_e.*
4) *If P_e is excessive, bad ADC, stop.*
5) *If P_e is not excessive, apply IFFT to Y_{en} to obtain the time domain residue y_{en}.*
6) *At each (y_n, y_{en}) pair, write $y_{en}=E_{MSB}(C_{MSB})+E_{ISB}(C_{ISB})+E_{LSB}(C_{LSB})+noise$, where C_{MSB}, C_{ISB}, and C_{LSB} are the corresponding MSB, ISB, and LSB codes for each y_n.*
7) *Identify $E_{MSB}(C_{MSB})$, $E_{ISB}(C_{ISB})$, and $E_{LSB}(C_{LSB})$.*
8) *Construct INL(k) and DNL(k) for all code k; k is from 0 to 2^N for an N-bit ADC.*

There are two requirements on the setup of the efficient linearity test method: (1) the range of the input must be smaller than the full range of the ADC under test, and (2) the input signal must be sampled coherently. Figure 3.1 (a) shows an ADC output spectrum when the input is in the ADC's full range and sampled coherently. The total number of sampled points is M=65536, and the number of periods of the output is J=4357, which is an integer and prime to M, satisfying the coherency condition. If we follow common industry practice and select the input at about 0.25 dBFS, the input range is a little bit smaller than the ADC's full range to avoid clipping. Once the fundamental and DC the component are set to be 0 (the power is set to be 0), the rest of the spectrum contains the information of harmonics and noise. Applying steps v–viii of the efficient linearity algorithm to the spectrum residue, INL and DNL of the ADC under test can be obtained.

Although the first requirement of the amplitude was not stated in the chapter, it is actually a default requirement in the implementation of the algorithm in [12]. If the input range exceeds the ADC's full range, the ADC output can be clipped, and there are high spurs in the ADC output spectrum. However, clipping is unavoidable if one wants to test the nonlinearity for all the ADC codes. If the input range is smaller than the full range of the ADC under test, as shown in Figure

3.2(a), there are a lot of codes untested in the two ends (near code 0 and code 2^N–1 for an N-bit



(a)

(b)

(c)

(d)

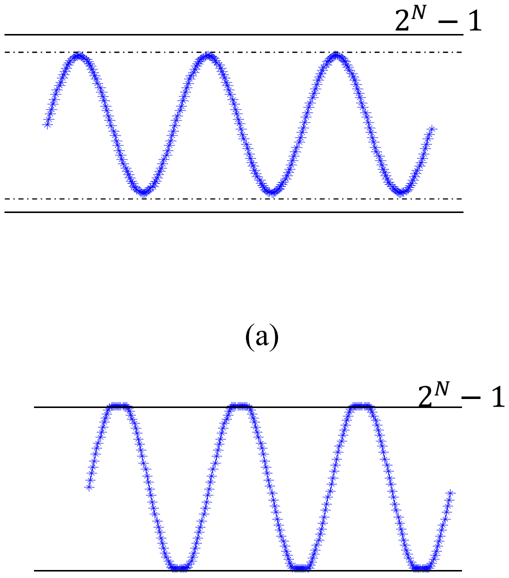
Figure 3.1 Spectra of the ideal ADC output (a), clipped ADC output (b), noncoherent output (c), and both clipped and non-coherent output (d)

ADC). Although in [12] the nonlinearity for all the codes are not necessarily required to obtain the INL and DNL curves, the nonlinearity of codes in the two ends needs to be extrapolated from the nonlinearity of codes covered by the input, which can decrease the accuracy of the method. If the amplitude of the input is increased to let the input range be slightly less than the full range of the ADC, the input signal can still be clipped as (1) the variation of the full range between ADC to ADC, (2) nonstationary of the input, and (3) the noise issue [9]. All these issues cause the problem where the amplitude is challenging to control precisely in the testing setup. Figure 3.1(b) shows a spectrum when the ADC output is clipped by the ADC's full scale (2% overrange). Compared with the correct spectrum in Figure 3.1(a), spurs are increased in the spectrum shown in Figure 3.1(b). Not only are spectrum specifications obtained from this spectrum not correct, but also INL and DNL cannot be estimated correctly from this ADC output.

It is well-known that coherency is another condition that is hard to achieve. And for high-performance ADCs, even slight noncoherency can cause the spectrum to be totally corrupted. Figure 3.1(c) shows the spectrum of the noncoherent ADC output. It can be seen that the power of the fundamental leaks to neighbor bins and masks the information of harmonics and noise. Figure 3.1(d) shows the spectrum of the ADC output with simultaneous noncoherency and amplitude clipping.

It can be concluded that noncoherency and amplitude clipping are two important issues in both ADC spectral and linearity testing. A method that can dramatically increase the linearity test efficiency was introduced, but it requires coherency and no amplitude clipping. There is a method for spectral testing that can relax the two requirements [10]. The idea of noncoherency and amplitude clipping can be applied to the linearity test if the accuracy in the fundamental estimation can be increased. And this will be discussed in the simulation result section.

The method proposed in this chapter can efficiently obtain the spectral and linearity performance using one sequence of ADC output data and can simultaneously relax the requirements on coherency and amplitude. Spectral and linearity characteristics can be obtained accurately and efficiently even using imprecise instrumentations.



(a)

(b)

Figure 3.2 ADC output when the input range is smaller(a) or lager (b) than the ADC's full range

3.3 Proposed Method

In the proposed method, the ADC under test is tested using a pure input sine wave slightly exceeding the ADC's full range, and it is sampled noncoherently. The noncoherent and clipped fundamental is first identified using four-parameter sine wave fitting (FPSF) method, which is robust and accurate. Then the noncoherent and clipped fundamental is subtracted from the ADC output, leaving the residue containing information of the harmonics and noise.

The residue of the unclipped ADC output is interpolated to an ideal fundamental, which contains the integer number of cycles, and its range is in the ADC's full range. By adding the interpolated information of harmonics and noise to the newly constructed coherent and unclipped fundamental, a new sequence is obtained. Spectrum is obtained by applying DFT to the new sequence to estimate the spectral specifications. The residue of the unclipped points is used to estimate the INL and DNL for the full codes.

Let the input of the ADC be $V_{in}(t)=A\cos(2\pi ft+\phi)+V_{os}$, where A, f, ϕ, and V_{os} are the amplitude, frequency, initial phase, and offset of the input signal, respectively. Let y[n] be the digital output codes converted by the N-bit ADC under test:

$$y[n]=\begin{cases} C_A\cos(2\pi\frac{f}{f_s}n+\phi)+h.d+w[n]+C_{os} & V_{ref}^- < V_{in} < V_{ref}^+ \\ 0 & V_{in} \le V_{ref}^+ \\ 2^N-1 & V_{in} \ge V_{ref}^- \end{cases} \tag{3.1}$$

In equation (3.1), C_A and C_{os} are the amplitude and offset of the ADC output; f_s is the sampling frequency; h.d is the high-order harmonics $\sum_{h=2}^{H} C_{Ah}\cos(2\pi\frac{f}{f_s}n+\phi_h)$; H is the number of harmonics; and w[n] is the summation of additive noise and quantization noise, n=0,1,2…M–1; and M is the total number of sampled points, which is usually selected to be the power of 2 for the efficiency of FFT. V_{ref}^+ and V_{ref}^- are the top and bottom reference voltages of the ADC under test, respectively. Let J be the number of ADC output periods and f/fs×M=J. J is an integer if the input is sampled coherently and y[M]=y[0]; otherwise, it is noncoherently sampled. In this chapter, J can be a noninteger. The following steps show the details of the proposed method to estimate linearity and spectral specifications from clipped and noncoherent y[n].

3.3.1 Fundamental Identification

The FPSF method is a standard and accurate method for fitting digitized waveform data to a sine wave [1]. In ADC testing, it can be used to identify the fundamental and high-order harmonics even under noncoherent condition. Because of space limitations, the details of the sine wave fitting method are not discussed here.

In the general sine wave fitting method, all points in the digitized waveform are used for the fitting. However, in this chapter, any ADC output that exceeds the valid code range should be excluded from the fitting. Formula (3.2) describes the index of the ADC output used for the sine wave fitting. In this equation, C_{min} is the minimum valid code. It can be 0 or hundreds to thousands LSBs larger than 0. C_{max} is the maximum valid code, and it can be 2^N–1 or hundreds to thousands LSBs less than 2^N–1. The selection of C_{min} and C_{max} depends on the architecture of the ADC under test, as long as the linearity of codes between C_{min} and C_{max} is guaranteed. If the manufacturer does not care about the linearity of codes exceeding the range of (C_{min}, C_{max}), the proposed method in this chapter does not need to include them. In this chapter, to show how to construct the linearity of the whole code range (0,2^N–1), we set C_{min}=0 and C_{max}=2^N–1, respectively.

$$n_{sf} = \{ n \mid C_{\min} < y[n] < C_{\max} \} \tag{3.2}$$

Pairs of (y[n_{sf}],n_{sf}) are used for the sine wave fitting to obtain the fundamental of the ADC output expressed as equation (3.3).

$$y_f[n] = \hat{B}\cos(\hat{\omega}n) + \hat{C}\sin(\hat{\omega}n) + \hat{D} \tag{3.3}$$

$\hat{B}$, $\hat{C}$, $\hat{\omega}$, and $\hat{D}$ in equation (3.3) are the four parameters of sine wave estimated from the FPSF method.

3.3.2 Residue Information

After the fundamental is obtained from Section III.A, it is subtracted from the ADC output codes, and the residue including harmonics and noise are left. As the clipped points do not have information of the AC property, they are excluded for the following processing. The indexes of the unclipped points are

$$n_{nc} = \left\{ n \mid 0 < y[n] < 2^N - 1 \right\} \tag{3.4}$$

And residues of the unclipped points are

$$y_e[n_{nc}] = y[n_{nc}] - y_f[n_{nc}] \tag{3.5}$$

3.3.3 Linearity Test

Now that the pairs of ADC output error and output codes $(y_e[n_{nc}],y[n_{nc}])$ are obtained, the algorithm of the efficient linearity test can be slightly adjusted to estimate the linearity of the ADC under test.

The number of MSB, ISB, and LSB segments—ms, mi, and ls—is determined first according to the principles in [12]. For example, the segments of ms-mi-ls can be 6-6-6 or 7-6-5 or other combinations for an 18-bit ADC. The MSB codes C_{MSB} are from 0 to 2^{ms}–1, ISB codes are from 0 to 2^{mi}–1, and LSB codes are from 0 to 2^{ls}–1. For an ADC output code, it can be split as the combination of the MSB code, ISB code, and LSB code. For example, the binary form of a decimal ADC output code 210823 is 110011 0111100 000111, and then $C_{MSB}=(110011)_b=51$,

$C_{ISB}=(011110)_b=30$, and $C_{LSB}=(000111)_b=7$ if ms=6, mi=6, and ls=6. As we have pairs of ($y_e[n_{nc}]$,$y[n_{nc}]$), and for each code $y[n_{nc}]$, the error can be written as

$$y_e[n_{nc}] = E_{MSB}(C_{MSB}) + E_{ISB}(C_{ISB}) + E_{LSB}(C_{LSB}) + w[n_{nc}] \tag{3.6}$$

In equation (3.6), n_{nc} is the index described in formula (3.4), and $E_{MSB}(C_{MSB})$, $E_{ISB}(C_{ISB})$, and $E_{LSB}(C_{LSB})$ are the error of the MSB, ISB, and LSB segment at C_{MSB}, C_{ISB}, and C_{LSB} of n_{nc}. Now the total number of parameters of the error term is $k_p=2^{ms}+2^{mi}+2^{ls}$, but the number of equations like (3.6) are far more than k_p. Least squares is used to solve the parameters of E_{MSB}, E_{ISB}, and E_{LSB}. Then the INL curve is constructed from them. For code C, its C_{MSB}, C_{ISB}, and C_{LSB} are first determined, and then

$$INL[C] = E_{MSB}[C_{MSB}] + E_{ISB}[C_{ISB}] + E_{LSB}[C_{ISB}] \quad C = 0,1,2...2^N - 1 \tag{3.7}$$

When the INLs of codes 0 to 2^N–1 are obtained, the INL curve can be obtained and the DNL curve can also be computed.

3.3.4 Spectral Test

In this section, we create a coherent and unclipped fundamental and interpolate the residue obtained from Section III.B to the new fundamental to get a new sequence consisting of the coherent and unclipped fundamental and interpolated error. By analyzing the spectrum of the new sequence, the specifications of the ADC under test can be obtained.

The noncoherent fundamental obtained from equation (3.3) is clipped to get the noncoherent and clipped fundamental as shown in equation (3.8):

$$y_f[n] = \begin{cases} \hat{B}\cos(\hat{\omega}n) + \hat{C}\sin(\hat{\omega}n) + \hat{D} & 0 < y_f[n] < 2^N - 1 \\ 0 & y_f[n] < 0 \\ 2^N - 1 & y_f[n] > 2^N - 1 \end{cases} \tag{3.8}$$

The unclipped part can be written as equation (3.9):

$$y_f[n] = \sqrt{\hat{B}^2 + \hat{C}^2}\,\cos\left(2\pi\frac{\hat{J}}{M}n + \phi'\right) + \hat{D} \tag{3.9}$$

In equation (3.9), $\phi' = \tan^{-1}\left(-\frac{\hat{C}}{\hat{B}}\right)$ if $\hat{C} < 0$ and $\phi' = \tan^{-1}\left(-\frac{\hat{C}}{\hat{B}}\right) \pm \pi$ if $\hat{C} > 0$, and $\hat{J}$ is the number of periods of the ADC output. It is obtained by $\hat{J} = \frac{\hat{\omega} \cdot M}{2\pi}$. As the ADC is sampled noncoherently, $\hat{J}$ is not an integer. Let the integer part of $\hat{J}$ be J$_{\text{int}}$. Next, we construct a coherent and unclipped fundamental with J$_{\text{int}}$, ϕ', and a new amplitude A'. Here A' is selected to let the range of the new fundamental be at about –0.25 dBFS of the whole ADC output code range (0,2$^{\text{N}}$–1). The new coherent and unclipped fundamental can be written as

$$y_f^{'}[n] = A'\cos\left(2\pi\frac{J_{\text{int}}}{M}n + \phi'\right) \tag{3.10}$$

The original ADC output y[n], residue y$_{\text{e}}$[n], and new fundamental $y_f^{'}[n]$ are folded into one cycle, respectively, according to their phase so that y[n] and $y_f^{'}[n]$ can be separated into points in the falling edge (marked as *f*) and rising edge (marked as *r*) as shown in Figure 3.3. The error y$_{\text{ef}}$[n] in the falling edge y$_{\text{f}}$[n] is interpolated to fundamental in the falling edge $y_{ff}^{'}[n]$ to get the new error $y_{ef}^{'}[n]$, and it is the same for the rising edge.

Equation (3.11) shows the interpolation of the new error at code $y_{ff}^{'}[n]$ at the falling edge:

$$y_{ef}^{'}[n] = y_{ef}[a] + \frac{y_{ef}[b] - y_{ef}[a]}{y_f[b] - y_f[a]}\left(y_{ff}^{'}[n] - y_f[a]\right) \tag{3.11}$$

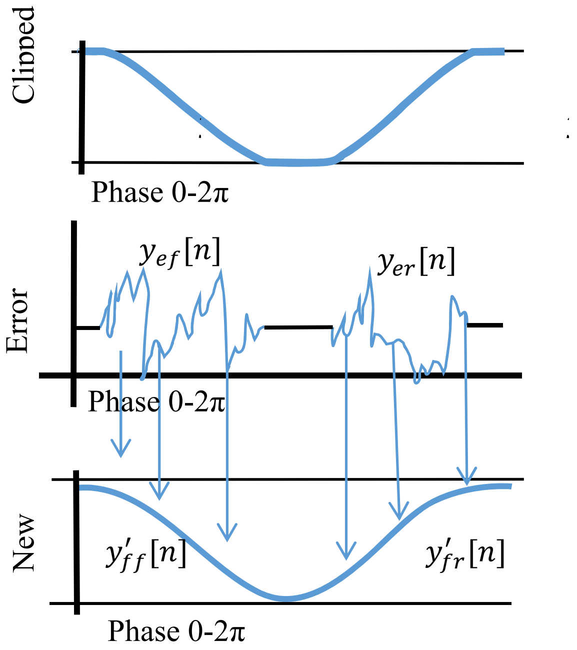


Figure 3.3 Interpolation of the unclipped residue

In equation (3.11), a and b are the indexes of the new fundamental at the falling edge. $y_f[a]$ is the minimum code, which is larger than $y'_{ff}[n]$, and $y_f[b]$ is the maximum code, which is smaller than $y'_{ff}[n]$. The error in the rising edge $y'_{er}[n]$ can also be estimated in the same way. When $y'_{ff}[n]$ and $y'_{er}[n]$ are obtained, they are unfolded to J$_{\text{int}}$ cycles according to the phase of $y'_f[n]$. When the interpolated error $y'_e[n]$ is estimated, the new sequence consists of the unclipped and coherent fundamental and interpolated error (harmonics and noise):

$$y'[n] = y'_f[n] + y'_e[n] \tag{3.12}$$

DFT is performed to the new sequence to obtain the accurate spectral characteristics.

Figure 3.4 shows the flowchart of the proposed algorithm in the cotest of linearity and spectral performance.

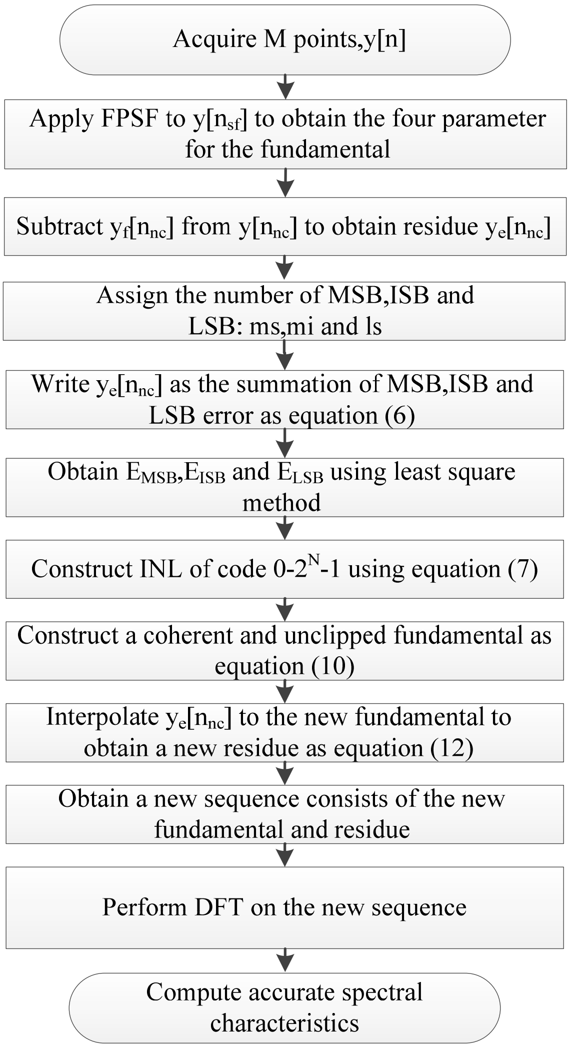

Acquire M points,y[n]
Apply FPSF to y[n_sf] to obtain the four parameter for the fundamental
Subtract y_f[n_nc] from y[n_nc] to obtain residue y_e[n_nc]
Assign the number of MSB,ISB and LSB: ms,mi and ls
Write y_e[n_nc] as the summation of MSB,ISB and LSB error as equation (6)
Obtain E_MSB,E_ISB and E_LSB using least square method
Construct INL of code 0-2^N-1 using equation (7)
Construct a coherent and unclipped fundamental as equation (10)
Interpolate y_e[n_nc] to the new fundamental to obtain a new residue as equation (12)
Obtain a new sequence consists of the new fundamental and residue
Perform DFT on the new sequence
Compute accurate spectral characteristics

Figure 3.4 Flowchart of the proposed method

3.4 Simulation Results

The proposed method of the cotest of spectrum and linearity with clipped and noncoherently sampled data is validated by simulation data generated in MATLAB. An 18-bit SAR ADC was modeled with mismatches in the charge capacitors in MATLAB as the capacitor mismatch is the main error source in the SAR ADC. And the error source from other aspects is covered in the measurement result. The additive noise to the input is 1 LSB. Two sets of input sine wave were converted into digital codes by the ADC under test: one in the full range of the ADC and coherently sampled is set as the reference, and the other one with an amplitude that covers 2% more than the full range of the ADC is noncoherently sampled. The total number of sampled points is 262,144 for each set, which is approximately one hit per code.

It needs to be stated how we select the test frequency for the cotest of linearity and spectral performance in the proposed method. In common practice in the industry, the linearity is usually tested at a low frequency. The spectral characteristics are usually tested at one typical frequency, which is usually around 1/10 of the Nyquist frequency. So in the simulation section, we select the input signal with a frequency at about 1/20–1/10 of the Nyquist frequency to test the linearity and spectral performance simultaneously.

3.4.1 Spectral Test

The spectra of the raw ADC output data from the two inputs are shown in Figure 3.5 (the blue spectrum is that of the reference, and the green one is the spectrum of the noncoherent and clipped ADC output). It can be seen that there are power leakage and increased spurs in the green spectrum. Spectral parameter values, including SNR, THD, and SFDR, directly obtained from this spectrum are totally incorrect, which should be discarded. Spectral parameters obtained from the reference spectrum (blue) are shown in Table 3.1.

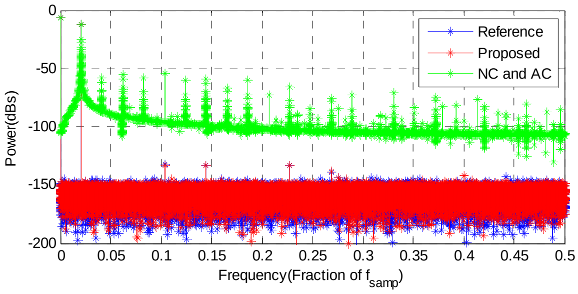


Figure 3.5 Spectra of the coherent and unclipped ADC output (blue), the noncoherent and clipped output (green), and the proposed noncoherent and clipped output (red)

Table 3.1 Spectral performance from the reference method and the proposed method (unit dBs)

	SNR	THD	SFDR
Reference	98.87	–117.15	120.22
Proposed	98.67	–117.67	120.65

Applying the proposed method to the clipped and noncoherent ADC output, the new spectrum is shown as the red one in Figure 3.5. It can be seen that the power leakage and spurs are totally removed and the spectrum is identical with the reference one. It can be concluded that the proposed method is capable of removing the clipping and power leakage effect. The spectral parameters estimated from the proposed method are also shown in Table 3.1. It can be seen that when applying the proposed method to the clipped and noncoherently sampled data, specifications such as THD, SFDR, and SNR can be accurately estimated.

3.4.2 Linearity Test

The true INL of the ADC extracted from the ADC model is used as the reference. Once the ADC is generated, each and every code transition voltage is found using a binary search, similar to the standard servo loop method. From the true transition voltages, the true INL/DNL can be

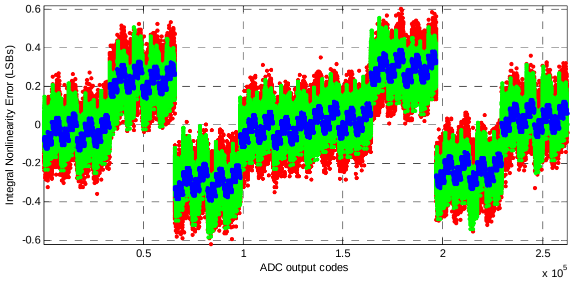
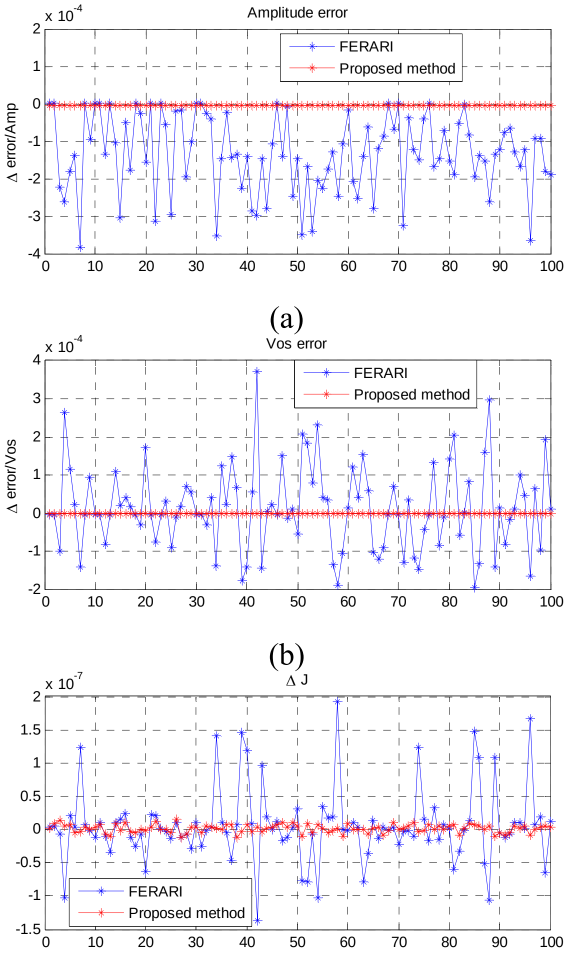


Figure 3.6 True INL (blue), INL from the histogram method (red), and INL from the proposed method (green)



(a)

(b)

(c)

Figure 3.7 Accuracy of amplitude (a), offset (b), and frequency(c)

computed. Also, the histogram method is used to test the INL for comparison. In this simulation, the number of hits per code of histogram is set to be 128. Figure 3.6 shows the INL of the ADC under test: true INL (blue), INL from the histogram method (red), and INL from the proposed method (green). It can be seen that the true INL lies in the middle of the INL from histogram and the proposed method, and the noise band of the proposed method is smaller than that of histogram, but it only uses data 128 times less than that of the histogram method.

3.4.3 Estimation Accuracy

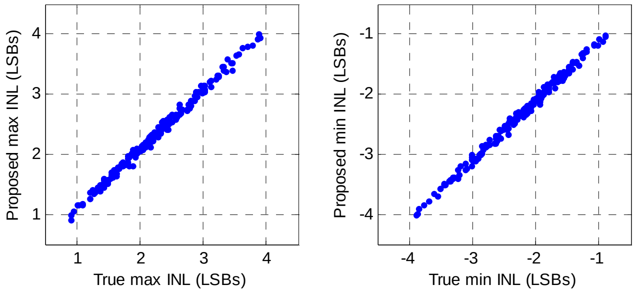
The accuracy of the proposed method in fundamental estimation is compared with that of the FERARI method. One hundred sets of sine wave with randomly generated amplitude, offset, and frequency are used, converted by the ADC modeled in the beginning of this section. These amplitude, offset, and frequency values are recorded as the true values for reference. Parameters obtained from the proposed method and the FERARI method are compared in Figure3.7. The error is calculated as $\frac{X_{estimated} - X_{true}}{X_{true}}$, where X is the amplitude, offset, or J (total number of periods, can represent input frequency). Based on the test result, the accuracy of the proposed method is several orders higher than that of the FERARI method.

3.4.4 Robustness Test

The robustness of the proposed method is shown with respect to noncoherent sampling and amplitude clipping up to 3%. Two hundred 18-bit SAR ADCs were generated in MATLAB with random capacitor mismatch. The true INL of each ADC were obtained from the mismatch of the capacitors, and the true THD and SFDR were obtained using the standard DFT method under coherent sampling and no amplitude clipping, whereas the proposed INL, THD, and SFDR of each

ADC were obtained using input signal with randomly selected values of δ and overrange. The values of δ vary from –0.5 to 0.5 (total range), and overrange percentage was in the range 0 to 3. The data record length for each run was 65536.

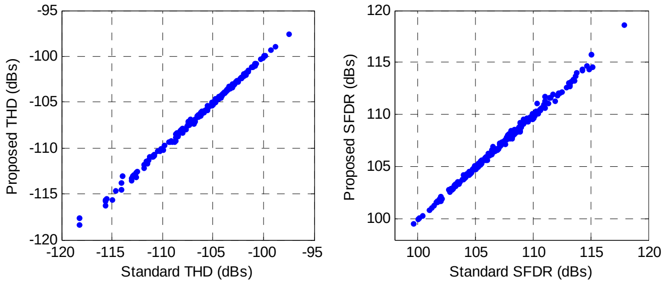
Figure 3.8 shows the relationship between INL obtained from the proposed method and the true INL of the ADC under test. Figure 3.9 shows the relationship of the proposed THD and SFDR and these parameters from the standard DFT method. The fitting line of the proposed parameter (INL, THD, or SFDR) versus the true parameter is a straight line with a slope of 1. This shows that the method is accurate and robust to both noncoherent sampling and amplitude overrange up to 3%.



(a) (b)

Figure 3.8 The proposed maximum INL versus the true maximum INL (a)

The proposed minimum INL versus the true minimum INL (b)



(a) (b)

Figure 3.9 The proposed THD versus the true THD (a), the proposed SFDR versus the true SFDR (b)

3.5 Measurement Results

The proposed method of the cotest with clipped and noncoherently sampled data has been validated to be true by simulation data generated in MATLAB. But in real testing, there are more unpredictable factors that cannot be simulated, and the ultimate application of this algorithm is experimental testing. This proposed method is validated by measurement data of an unreleased part (18-bit ADC) of the new ADC product from an R&D group in a leading semiconductor company. Three sets of data are taken from the R&D group: (1) INL curve tested using the histogram method with 128 hits per code; (2) a sequence of ADC output with coherent sampling and amplitude clipping, the total data length being 262,144; (3) a sequence of ADC output with noncoherent sampling and amplitude clipping, the total data length being 262,144.

Table 3.2 Test result of the measurement result (unit: dBs)

	SNR	SFDR	THD
Reference	97.24	118.35	– 116.20
Proposed	97.13	116.28	– 115.85

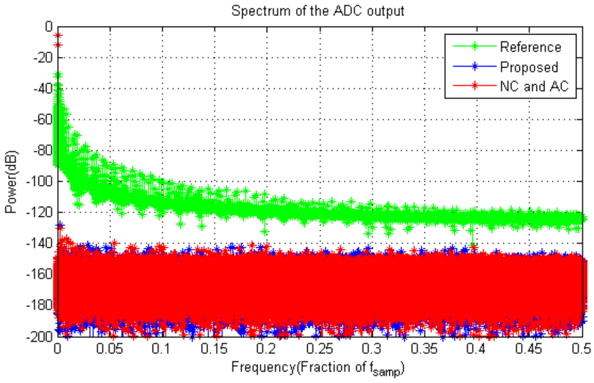


Figure 3.10 Spectra of the reference (blue), noncoherent and clipped output (green), and the proposed method on noncoherent and clipped output (red)

3.5.1 Spectral Test

Applying the FERARI method to data sequence 2 described in the last paragraph, the spectrum and spectral performance are taken as a reference for spectral testing. Figure 3.10 shows the spectra of reference, noncoherent and clipped data, and application of the proposed method to the noncoherent and clipped data. The noncoherency and clipping effect has been removed by the proposed method. Table 3.2 shows the spectral performance using the proposed method on noncoherent and clipped data, and from the reference method, it can be seen that the estimation of spectral performance is accurate.

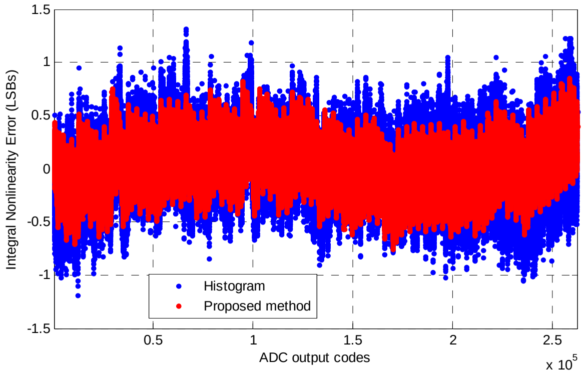


Figure 3.11 Measurement results comparing the INL of an 18-bit SAR ADC using the ramp histogram method with 128 hits/code against the INL of the proposed method using an average of 1 hit/code

3.5.2 Linearity Test

The reference INL of the ADC under test is obtained from the ramp histogram test with 128 hits per code, which is shown as the blue dots in Figure 3.11. The red dots are the INL obtained from the proposed method with one hit per code and with noncoherency and amplitude clipping. It can be seen that the red dots lie right near the center of the blue dots' noise band, and the proposed method has better precision than the histogram method.

3.6 Conclusion

A new test method that efficiently and accurately estimates the linearity and spectral characteristics of an ADC with noncoherent and overrange input was proposed. This method greatly reduces the data acquisition time and makes the full-code linearity test possible. It relaxes the requirement on precise control over frequency and amplitude of the input signal for spectral testing and makes it possible for the cotest of linearity and spectrum from one sequence of data. To identify the noncoherently sampled and overrange fundamental, a new and accurate fundamental identification method was proposed. The residue obtained after subtracting the estimated fundamental from the collected data was used to calculate the linearity characterization and interpolated to an ideal fundamental to obtain accurate spectral results. The accuracy and robustness of the proposed method was validated by both simulation results and measurement results. The proposed method is especially suitable for applications where it is challenging to obtain precise control over frequency and amplitude of test signal. Accordingly, the test cost and test time are reduced dramatically.

References

[1] "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," IEEE Std.1241, 2010.

[2] "IEEE standard for Digitizing Waveform Recorders, " 2007.

[3] J. Doernberg, H. S. Lee, and D. A. Hodges, "Full-Speed Testing of a/D Converters," IEEE Journal of Solid-State Circuits, vol. 19, pp. 820-827, 1984.

[4] H. W. Ting, B. D. Liu, and S. J. Chang, "A histogram-based testing method for estimating A/D converter performance," IEEE Transactions on Instrumentation and Measurement, vol. 57, pp. 420-427, Feb 2008.

[5] F. Alegria, P. Arpaia, A. M. D. Serra, and P. Daponte, "Performance analysis of an ADC

histogram test using small triangular waves," IEEE Transactions on Instrumentation and Measurement, vol. 51, pp. 723-729, Aug 2002.

[6] S. K. Sudani and D. Chen, "FIRE: A Fundamental Identification and Replacement Method for Accurate Spectral Test Without Requiring Coherency," IEEE Transactions on Instrumentation and Measurement, vol. 62, pp. 3015-3025, 2013.

[7] Y. Zhongjun, C. Degang, and R. Geiger, "A computationally efficient method for accurate spectral testing without requiring coherent sampling," International Test Conference, 2004, pp. 1398-1407.

[8] L. Xu and D. Chen, "ADC spectral testing allowing amplitude clipping," 2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1526-1529.

[9] X. Li and C. Degang, "Efficient Spectral Testing With Clipped and Noncoherently Sampled Data," Transactions on Instrumentation and Measurement, 2013.

[10] S. Sudani, L. Xu, and D. Chen, "Accurate Full Spectrum Test Robust to Simultaneous Non-coherent sampling and amplitude clipping," IEEE International Test Conference (ITC), 2013.

[11] A. C. Serra, M. F. da Silva, P. M. Ramos, R. C. Martins, L. Michaeli, and J. Saliga, "Combined spectral and histogram analysis for fast ADC testing," IEEE Transactions on Instrumentation and Measurement, vol. 54, pp. 1617-1623, Aug 2005.

[12] Z. J. Yu and D. G. Chen, "Algorithm for Dramatically Improved Efficiency in ADC Linearity Test," Proceedings International Test Conference, 2012.

CHAPTER 4. ACCURATE AND EFFICIENT METHOD OF JITTER AND NOISE SEPARATION AND ITS APPLICATION TO ADC TESTING

Jitter is a crucial factor in high-speed and high-performance ADC testing. This chapter proposes an efficient and accurate jitter estimation method based on one frequency measurement. Applying a simple mathematical processing to the ADC output in time domain, the RMS of jitter and noise power are obtained. Furthermore, prior information of harmonics need not be known before the processing. The algorithm is robust enough that nonharmonic spurs do not affect the estimation result. Using the proposed algorithm, specifications of the ADC under test can be obtained without the jitter effect. Simulation results of ADCs with different resolutions show the functionality and accuracy of the method.

4.1 Introduction

The ADC is one of the important categories of mixed-signal products. Accurate and efficient testing of high-performance ADCs is a challenging task in the modern industry as the speed and performance of the ADC are increased dramatically. Spectral testing, also known as AC test, is related to the ADC's dynamic performance, including SNR, THD, SFDR, and so on. The challenges and test cost of spectral testing come from the following aspects [1]: (1) low-distortion stimulus, (2) coherency between input frequency and sampling frequency, (3) stationarity of reference, and (4) clean sampling clock.

Jitter, which is defined as the variation in the sampling instant, is an important specification in high-speed analog-mixed signal devices. Deterministic jitter (DJ) and random jitter are two categories of jitter [2]. As the increase of frequency and data rate, jitter can be the ultimate limit of the performance in some applications. In ADC testing, jitter plays a crucial role as the SNR decreases as the sampling rates or input frequency increases with uncertainty in the sampling clock. Clock jitter, as well as the ADC's intrinsic noise, increases the noise floor of the ADC output spectrum. It is necessary to accurately separate jitter from the ADC intrinsic noise to get the true ADC performance.

Many researchers have proposed methodologies to test the ADC accurately and efficiently even under a nonideal testing environment. With a known impure source, high-resolution ADC spectral specifications were obtained under a noncoherent sampling condition [3]. An accurate ADC testing method with noncoherently sampled data was proposed in [4]. A testing method was introduced in [5] when the input amplitude is clipped. Conventional jitter measurement methods apply two inputs with sufficient separate frequencies with the ADC under test to calculate the jitter information [1,6,7]. The dual-frequency method increases the test cost as the requirement of the signal generator and synthesizers are high for the ATE test. And for the SoC test, the two frequency methods increase the test cost because the low-frequency on-chip test needs a large die area for capacitors. An FFT-based jitter separation method was proposed to separate random jitter and deterministic jitter [2]. An analytical method was proposed to extract the instantaneous and RMS sinusoidal jitter from phase-locked loops (PLL) output [8]. In ADC testing, the summation of quantization noise is the noise that needs to be estimated in calculating the specification SNR. Random jitter affects the ADC output as well as the noise, and it cannot be estimated from the methods listed above. A fast and accurate jitter and noise measurement method with one frequency test signal was proposed in [9]. By setting a certain number of harmonics of the ADC output to be zero in the frequency domain, the residue of the ADC output was separated to be two sets with different jitter powers. The RMS of jitter was obtained by processing the two sets of data. This method is accurate and efficient in jitter estimation. However, it requires knowing the number of harmonics before setting them to be zero, and nonharmonic spurs affect the test result.

This chapter proposes a new method to accurately measure jitter and noise power based on one frequency measurement, and it does not require prior knowledge of harmonics. Moreover, compared with [9], the accuracy of the proposed method is not affected by nonharmonic spurs. And this algorithm can separate jitter and noise through a simple mathematical processing, which is much more efficient than any other method. With this method, the correct spectral performance of the ADC under test can be obtained even using a sampling clock with random jitter. This method relaxes the requirement of the sampling clock and then reduces the test cost, making it possible to test an ADC using an imprecise instrument.

This chapter is organized as follows: In Section II, the jitter effect on ADC spectral testing is introduced. Section III proposes the new method to estimate jitter noise. Section IV shows the simulation results and accuracy of the method. The last section concludes the result.

4.2 Problem Statement

The ADC spectral testing is usually implemented by applying a pure sine wave to the device under test (DUT) and using FFT to analyze the output codes. Parameters such as THD, SFDR, and SNR can be obtained from the FFT spectrum. The model of ADC spectral testing with a clock jitter is shown in Figure 4.1. A pure input sine wave is applied to the ADC, where the additive noise of the ADC is modeled as random voltage added to the input signal. Jitter is modeled as random clock variation to the ideal clock instant. A data sequence of the ADC output x_n is collected to analyze the spectral specifications.

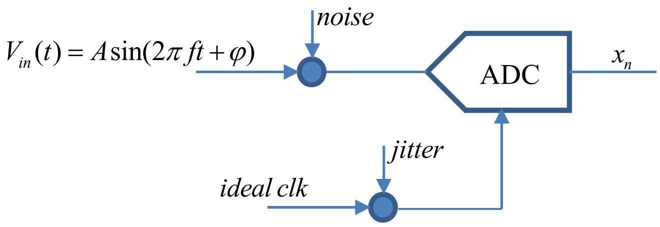


Figure 4.1 ADC test configuration

Assume that the input signal of the ADC under test is a pure sine wave, which can be written as

$$V_{in}(t) = A\sin(2\pi ft + \varphi) \tag{4.1}$$

where *A, f*, and φ are amplitude, frequency, and phase of the input signal, respectively. One set of the ADC output with M sampling points are collected, and the analog representation of the output is

$$x_n = V_{in}(nT_s + \delta_n) + h.d + w_n \qquad n = 0,1....M-1 \tag{4.2}$$

where *h.d* is high-order distortions, w_n is the summation of additive and quantization noise, δ_n is the clock jitter at each sampling instant, and T_s is the sampling period. Here, both jitter and noise are modeled as random variables following the Gaussian distribution: $\delta_n \sim N\ (0,\sigma_1^2)$ and $w_n{\sim}N\ (0,\sigma_2^2)$. This ADC output set is coherently sampled, meaning that it contains an integer number of cycles, and the phase of each sampling point is distinct.

The jitter effect in ADC testing is shown in Figure 4.2. The input should be sampled at the rising edge of the sampling clock, but the uncertainty in clock instant δ_n causes uncertainty in the output ΔV. And the value of the uncertainty at the output is proportional to the input slope. In ADC spectral testing, if the clock jitter is random and white, the error induced by jitter varies with the input phase: it is larger near zero-crossing and smaller at the top and bottom of the sine wave, which is also validated by the simulation result in [9].

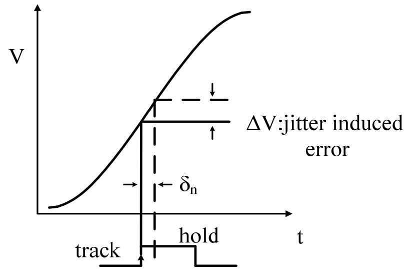
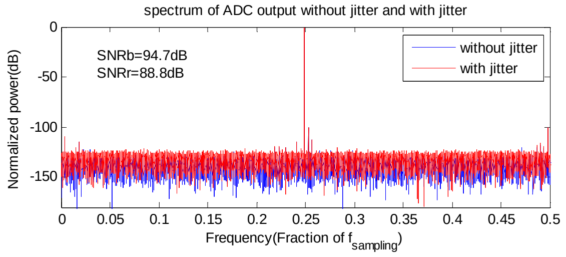
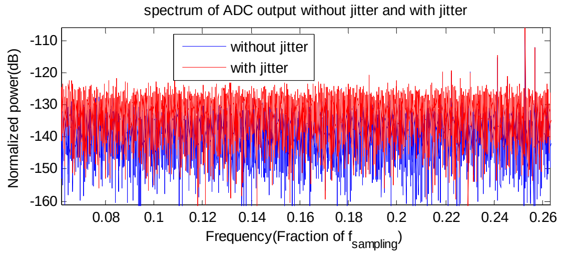


Figure 4.2 Jitter effect on ADC testing



(a)Spectrum of a 16-bit ADC



(b) Portion of the spectrum in (a)

Figure 4.3 Spectra of ADC output with and without the jitter effect

In the frequency domain, random jitter increases the noise floor as well as the additive noise in the ADC output spectrum, which can be seen in Figure 4.3 (a) shows spectra of a 16-bit ADC with and without the jitter effect. Except for jitter information, the other conditions of the two spectra are the same. The blue curve is the spectrum when the ADC is sampled with a clean clock, and the red curve is the spectrum when the ADC is sampled with a clock jitter. Figure 4.3 (b) shows a portion of the spectra in (a), and it can be seen that the noise floor of the spectrum with jitter is higher than that without jitter. It can also be seen from the specification SNR that the SNR of the red curve is lower than that of the blue curve.

The summation of the quantization error and additive noise is what we are interested in as it is important to get the specifications of SNR. For convenience, the summation of the quantization error and additive noise is called noise in the rest of the chapter. If one directly calculates SNR from the ADC output spectrum with jitter, it will be less than the true SNR as the jitter is also treated as noise, which is illustrated in Figure 4.3. Jitter cannot be separated from noise by a simple FFT method [6] as it has the same effect on the noise floor. The following section will discuss the property of jitter in ADC testing and separate jitter and noise information using the time domain ADC output.

4.3 Proposed Method

This section proposes a new method to separate jitter and noise using the time domain ADC output data. Two segments of the ADC output sampled under a coherent condition are collected. By subtracting the two data sets, the residue containing information of jitter and noise is obtained. Applying some mathematical processing to the residue according the characteristics of jitter and noise, the power of jitter and noise can be separated. The processing is based on assumptions that the two data sets have the same fundamental and harmonic information, the jitter effect on the harmonics can be ignored, and the two data sets are collected coherently. These assumptions will be explained in the following paragraphs.

The ADC output data sequence consisting of M points is described as equation (4.2) in Section II. As the RMS of jitter is usually small, we apply Taylor expansion to equation (4.2), and the expression of x_n can be written as (4.3)

$$x_n \approx A\sin(2\pi fnT_s+\varphi)+2\pi fA\cos(2\pi fnT_s+\varphi)\delta_n+h.d+w_n \tag{4.3}$$

It can be seen from equation (4.3) that the error of the ADC output caused by the jitter effect ($2\pi fA(cos(2\pi fnT_s+\varphi)\delta_n$) is proportional to the slope and frequency of the input signal. For the same clock jitter, the error caused by jitter is larger at a higher input frequency. So jitter is usually measured when the input frequency is a little bit less than half of the sampling frequency.

Moreover, as shown in equation (4.3), we just consider the effect of jitter on the fundamental and neglect its effect on harmonics because the jitter effect on harmonics is very small. Taking an 18-bit ADC, for example, the THD of an 18-bit ADC is at –110 dB, and jitter is usually at *ps* level. Then the error of harmonics caused by jitter is lower than –200 dB, which is far below the noise floor and obviously can be neglected.

If we continuously collect another set of output of the ADC under test with the same data length of x_n, we can have the expression of the new data sequence as (4.4):

$$x_n^{'} \approx A\sin(2\pi fnT_s+\varphi)+2\pi fA\cos(2\pi fnT_s+\varphi)\delta_n^{'}+h.d^{'}+w_n^{'} \tag{4.4}$$

as $x_n^{'}$ is right following x_n, and also coherently sampled, it has the same initial phase with that of x_n. The fundamentals and high-order of harmonics of the two sets are identical. Then the values of x_n

and x_n' are identical for each sampling instant, except for the influence of jitter and additive noise. That is why jitter and noise in equation (4.4) are δ_n' and w_n' rather than δ_n and w_n.

Subtracting (4.4) from (4.3), the residue can be written as equation (4.5):

$$\begin{aligned} e_n = x_n - x_n' &\approx 2\pi fA\cos(2\pi fnT_s+\varphi)(\delta_n-\delta_n')+(w_n-w_n') \\ &= 2\pi fA\cos(2\pi fnT_s+\varphi)\Delta_n + N_n \end{aligned} \tag{4.5}$$

where $\Delta_n = \delta_n - \delta_n'$ and $N_n = w_n - w_n'$. As illustrated previously, the jitter effect on high-order harmonics can be neglected, and the result of *(h.d-h.d')* is almost 0, which is not shown in equation (4.5). And if $\delta_n \sim N(0,\sigma_1^2)$ and $w_n \sim N(0,\sigma_2^2)$, then δ_n' and w_n' have the same distribution as δ_n and w_n, respectively, as the two sets of data are collected continuously in the same environment. Based on basic statistics knowledge, the distribution of Δ_n and N_n can be obtained as $\Delta_n \sim N(0,2\sigma_1^2)$ and $N_n \sim N(0,2\sigma_2^2)$.

Under the conditions of coherent sampling and long data length, several mathematical processings are applied to the residue data described in equation (4.5) to separate jitter and ADC noise. We first get the summation of the squared residue:

$$\begin{aligned} \sum_{n=0}^{M-1} e_n^2 &= \sum_{n=0}^{M-1}(2\pi fA)^2\cos^2(2\pi fnT_s+\varphi)\Delta_n^2 + \sum_{n=0}^{M-1} N_n^2 + \sum_{n=0}^{M-1} 4\pi fA\cos(2\pi fnT_s+\varphi)\Delta_n N_n \\ &\approx 4M(\pi fA)^2\sigma_1^2 + 2M\sigma_2^2 \end{aligned} \tag{4.6}$$

Now one relationship between jitter and noise has been obtained as equation (4.6).

Next, each value of the residue is multiplied by $cos(2\pi fnT_s)$, and we get equation (4.7):

$$\begin{aligned} e_n\cos(2\pi fnT_s) &= 2\pi fA\cos^2(2\pi fnT_s+\varphi)\Delta_n + \cos(2\pi fnT_s+\varphi)N_n \\ &= 2\pi fA\left(\frac{1}{2}+\frac{1}{2}\cos\left(2\pi fnT+\varphi\right)\right)\Delta_n + \cos(2\pi fnT_s+\varphi)N_n \end{aligned} \tag{4.7}$$

The total M points' summation of (4.7) is

$$\sum_{n=0}^{M-1}\left(e_n\cos(2\pi fnT_s)\right)^2 \approx 3M\left(\pi fA\right)^2\sigma_1^2 + M\sigma_2^2 \tag{4.8}$$

Another relationship between jitter and noise has been obtained as (4.8), and it is different from (4.7). Combining equations (4.7) and (4.8), the variance of jitter and noise can be estimated as

$$\sigma_1^2 = \frac{2\times\sum_{n=0}^{M-1}\left(e_n\cos(2\pi fnT_s+\varphi)\right)^2 - \sum_{n=0}^{M-1}e_n^2}{2M(\pi fA)^2} \tag{4.9}$$

$$\sigma_2^2 = \frac{3\times\sum_{n=0}^{M-1}e_n^2 - 4\times\sum_{n=0}^{M-1}\left(e_n\cos(2\pi fnT_s+\varphi)\right)^2}{2M} \tag{4.10}$$

The RMS of jitter and the power of intrinsic noise then can be calculated as

$$RMS_{jitter} = \sqrt{\sigma_1^2} \tag{4.11}$$

$$P_{noise} = \sigma_2^2 \tag{4.12}$$

Now jitter and noise are separated. Then P_{noise} can be used to calculate the ADC's specification of SNR.

The algorithm to accurately estimate jitter and noise in time domain using the proposed method for single-frequency testing is given as the following:

1) Apply a pure sine input to the ADC under test

2) Collect two sets of ADC output coherently with the same sample length: x_n *and* $x_n^{'}$.

3) Subtract the two sets; get the residue en.

4) Apply the two processes to en as (4.6) and (4.8).

5) Get the variance (RMS) of jitter and noise.

6) Calculate the true ADC specifications.

4.4 Simulation Results

In this section, the proposed jitter measurement method is validated by simulation data generated in MATLAB. An 18-bit ADC is modeled in MATLAB as a set of transition levels. The nonlinearity error is modeled as a set of Gaussian distributed variables with zero mean and a standard deviation of 0.003 LSB. The total sample points of each segment is 2^15. The sampling frequency is set to be 2 MHz. The amplitude of the input sine wave is selected to cover 99% of the full range of the ADC, and the input frequency is a little smaller than half of the sampling frequency, which is chosen carefully to satisfy the coherent sampling condition. The additive noise is introduced with the input. It is Gaussian distributed random noise with zero mean and a standard deviation of 1 LSB. The jitter is modeled as a random error added to the ideal sampling instant with zero mean and a standard deviation of 3 ps.

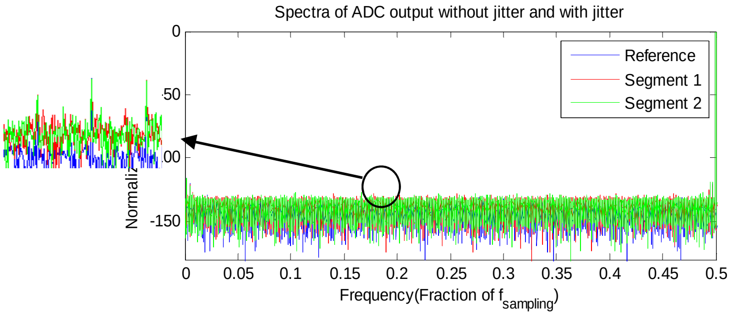


Figure 4.4 Spectra of ADC output with and without jitter

The true specifications of the ADC are obtained by sending a pure input sine wave to the ADC and sampling the input with an ideal clock to estimate the measurement error of the proposed method. This set of ADC output data without jitter was collected as reference. Then another two sets of ADC output were collected continuously and coherently with clock jitter. The jitter and additive noise added to the two input signal follow the same distribution. It can be seen from Figure

4.4 that the noise floors of spectra with jitter are at the same level as and higher than that of the reference spectrum. Although the jitter increases the noise floor, we select the number of total sampling points to be large enough that the noise floor is not that high to affect the harmonics.

As shown in Figure 4.4 the harmonics value of the red (segment 1 with jitter), green (segment 2 with jitter), and red (reference) spectra are at the same height. It means that ADC specifications such as THD and SFDR are not affected by jitter, and we do not compare them in this chapter. However, SNR is decreased as jitter increases the noise power.

Table 4.1 Testing result of the proposed method

Jitter added	3 ps	SNR true	99.01 dB
Estimated	3.02 ps	SNR estimated	99.09 dB

The SNR obtained from the ADC output with jitter is 93 dB, whereas the true SNR of the reference output is 99.01 dB. As specifications of SFDR and THD are not affected by jitter, they are not compared here. The residue of the two sets of data collected with jitter is shown in Figure 4.5. All information of fundamental and harmonics have been removed; only noise and jitter effect are left. After applying the proposed algorithm to the residue, the estimated jitter and SNR are shown in Table 4.1. It can be seen from the table that the estimation of jitter is almost identical to the jitter value added to the ADC, and the jitter effect on SNR has been thoroughly removed. It can be concluded that the proposed method can separate jitter and noise in ADC testing successfully. And after jitter and noise separation, correct ADC specifications can be obtained. Simulations of ADCs with different resolutions are performed to demonstrate the functionality and accuracy of the proposed method, as shown in Table 4.2. It can be said that the proposed method estimates jitter and SNR accurately.

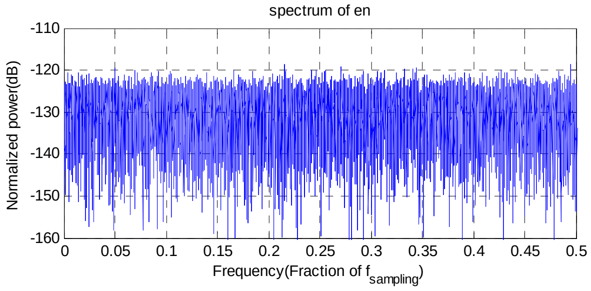


Figure 4.5 Spectrum of residue en

Table 4.2 Testing result of ADCs with different resolutions

	12-bit	14-bit	16-bit
Sampling frequency	10 MHz	5 MHz	2 MHz
M	2^13	2^13	2^14
THD	–81.24	–91.1 dB	–105 dB
Jitter added	5 ps	5 ps	3 ps
Jitter Estimated	5.0 ps	4.99 ps	3.01
SNR true	72.22 dB	82.78 dB	94.81 dB
SNR estimated	72.8 dB	82.62 dB	94.92 dB

In the ADC output spectrum, except for harmonic distortions, there are usually nonharmonic spurs generated by periodic jitter, time interleaving, and other factors. All the nonharmonic spurs affect the jitter testing result if they are not set to be zero before the processing in reference [9]. This requires prior knowledge of those spurs. However, the proposed method in this chapter is robust to nonharmonic spurs; testers do not need to know the number of harmonics and the locations of nonharmonic spurs of the ADC under test.

A 16-bit ADC modeled in MATLAB is used to verify the functionality of the proposed method in dealing with nonharmonic spurs. The nonharmonic spurs in the spectra shown in Figure 4.6 are modeled as periodic jitter added to the sampling clock. The reference ADC output is sampled with an ideal clock, and segment 1 and 2 are sampled with both random jitter and periodic jitter. The RMS of random jitter is 3 ps. The random jitter increases noise floor in the spectra, and periodic jitter generates nonharmonic spurs in the spectra.

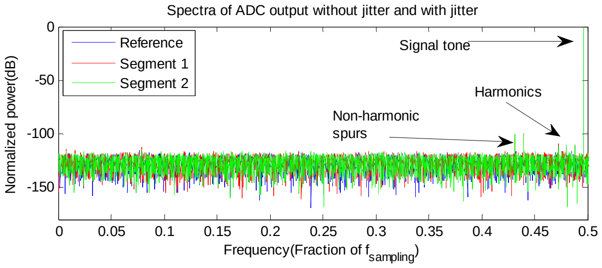


Figure 4.6 Spectra of a 16-bit ADC output without jitter (blue), with random and periodic jitter (red and green)

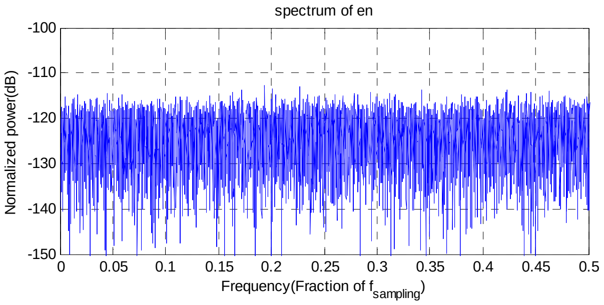


Figure 4.7 Spectrum of residue en

Figure 4.7 shows the spectrum of residue after the subtraction of the two segments of the ADC output with jitter. It can be seen that there is no information of harmonics and nonharmonic spurs left. The jitter estimation results of different methods are shown in Table 4.3. The estimation result of the proposed method is 2.98 ps, and the method from Reference [9] is 2.97 ps when the nonharmonic spurs are excluded. But if the nonharmonic spurs are not excluded, the estimation result error is large. From the table, it can be concluded that if nonharmonic spurs are not set to be zero, they can affect the jitter estimation result. However, the proposed method is accurate whether there are nonharmonic spurs or none.

Table 4.3 Jitter estimation results

Method	RMS of jitter
Proposed method	2.98 ps
Ref [9] with spurs exclusion	2.97 ps
Ref [9] without spurs exclusion	3.63 ps

4.5 Conclusion

A simple, accurate, and efficient method that simultaneously extracts the clock jitter and the additive noise of ADC testing was presented. Compared with conventional standard jitter estimation methods, this new algorithm only needs one frequency input applied to the ADC test. So it decreases the test cost both in hardware and in testing time. Compared with the conventional one-frequency method, the method in this chapter does not need prior information of harmonics and nonharmonic spurs, which increases the accuracy of the estimation. Moreover, for jitter and noise separation, it does not require FFT in the calculation. Simulation results validated the functionality and accuracy of the proposed method. In future work, the method needs to be justified by measurement results, and the algorithm can be modified to work under a noncoherent sampling condition.

References

[1] "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," IEEE Std.1241, 2010.

[2] T. J. Yamaguchi, H. X. Hou, K. Takayama, D. Armstrong, M. Ishida, and M. Soma, "An FFT-based jitter separation method for high-frequency jitter testing with a 10x reduction in test time," IEEE International Test Conference, 2007, pp. 1-8.

[3] S. Sudani, C. Degang, and R. Geiger, "High resolution ADC spectral test with known impure source and non-coherent sampling,"2013 IEEE International Symposium on Circuits and Systems (ISCAS), 2013, pp. 2674-2677.

[4] S. K. Sudani and D. Chen, "FIRE: A Fundamental Identification and Replacement Method for Accurate Spectral Test Without Requiring Coherency," IEEE Transactions on Instrumentation and Measurement, vol. 62, pp. 3015-3025, 2013.

[5] L. Xu and D. Chen, "ADC spectral testing allowing amplitude clipping,"2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1526-1529.

[6] S. Shariat-Panahi, F. A. C. Alegria, A. Manuel, and A. M. D. Serra, "IEEE 1057 Jitter Test of Waveform Recorders," IEEE Transactions on Instrumentation and Measurement, vol. 58, pp. 2234-2244, Jul 2009.

[7] W. Kester, "The Data Conversion Handbook," Analog Device, Inc, pp. 5.73-5.75, 2004.

[8] T. J. Yamaguchi, M. Soma, M. Ishida, T. Watanabe, and T. Ohmi, "Extraction of instantaneous and RMS sinusoidal jitter using an analytic signal method," IEEE Transactions on Circuits and Systems Ii-Analog and Digital Signal Processing, vol. 50, pp. 288-298, Jun 2003.

[9] W. Minshun, C. Degang, and D. Jingbo, "Fast & accurate algorithm for jitter test with a single frequency test signal," in 2011 IEEE International Conference on Electro/Information Technology (EIT), 2011, pp. 1-5.

CHAPTER 5. AN SNR ESTIMATION METHOD FOR ADC SPECTRAL TESTING WITH CLOCK JITTER

Spectral testing in an important category in ADC testing. The sampling clock quality is a crucial factor in ADC spectral testing. The accumulated clock jitter of the sampling clock generates power leakage in the fundamental component of the ADC output spectrum. The random clock jitter increases the noise floor of the ADC output spectrum. The two kinds of jitter thus decrease the SNR of the ADC under test. A new algorithm is proposed in this chapter to accurately estimate the SNR with sampling clock jitter. This method does not require precise a sampling clock and thus reduces the test cost. The ADC output sequence is separated into small segments. Segment pairs are found through initial phase matching. By analyzing the difference of the segments pairs, the noise power is estimated, and then SNR is obtained. Simulation and measurement results against the standard test methods collaborate to validate the accuracy and robustness of the new solution.

5.1 Introduction

With technological development, more complex circuits such as SoCs are designed. Although this approach decreases the design cost by embedding more circuits on a single chip, it increases the cost associated with testing such systems. The ADC is one of the important building blocks in modern mixed-signal products. It is one of the most widely used integrated circuits (IC) in SoCs. As speed and performance increase, the testing of the ADC is becoming challenging. Spectral testing, also called AC testing, includes testing of ADCs' dynamic specifications, such as THD, SFDR, SNR, and so on [1]. Spectral performance is usually tested by the DFT method using a pure sine wave as input. For SoCs, as all blocks are built in a single chip, the precision of the sampling clock is a crucial issue in the testing.

Clock jitter, defined as the difference between the real sampling instant and the ideal sampling instant, is an important specification in high-speed analog-mixed signal devices. In ADC testing, jitter plays a crucial role. Jitter can be classified as random clock jitter and accumulated clock jitter in this chapter.

Random clock jitter, usually Gaussian distributed, increases the noise floor of the ADC output spectrum, especially when the signal frequency is high. There is degradation in the estimated SNR due to the random clock jitter. If the sampling clock is on-chip, the stability cannot be guaranteed that it can generate an accumulated clock jitter except for random clock jitter.

Accumulated jitter in sampling clock can also be called frequency drift as it causes the sampling frequency to deviate from the intended one. The accumulated clock jitter generates the power leakage in the fundamental component of the ADC output spectrum. This effect is similar to the power leakage caused by noncoherency. However, traditional methods that are effective in solving the noncoherency problem cannot remove the effect caused by accumulated jitter. These methods include windowing [1] and FIRE [2].

Many methods have been proposed to measure or characterize clock jitter, especially for high-speed application [3,4]. These methods use precise and expensive instruments to analyze the clock signal to obtain the jitter information. For ADC application, many methods have been proposed to test ADC in the presence of clock jitter. Dual-frequency methods are the conventional methods in ADC testing with a random clock jitter. It applies two inputs with sufficient separate frequencies to the ADC under test to calculate the SNR degradation at a higher frequency [5,6]. This method increases the test cost as the requirement of the signal generator and synthesizers are high for the ATE test.

Methods [7–9] are proposed to test the correct ADC SNR in the presence of a random clock jitter. These methods can estimate the correct SNR when the random clock jitter is not large. However, the major limitation of these methods is that they only consider random clock jitter; they model the clock jitter as a variable with zero mean Gaussian distribution. Accumulated clock jitter is not taken into consideration.

The effect of sampling clock jitter on the acquired samples is analyzed in [9]. This paper proposed two methods to estimate jitter for superheterodyne receiver architectures and cognitive radio architectures at high sampling rates. A method to compensate for the jitter is also proposed. The methods are tested and validated through computer simulations and theoretical analysis.

The effective impact of jitter on the SNR of the ADC process [10] is evaluated when the observation interval is limited to a finite number of samples. It will be shown that, in this case, the jitter constraints on the sampling clock can be more relaxed.

The effect of sampling clock jitter on the acquired samples in the midst of quantization noise and random Gaussian noise is analyzed in [11]. The paper proposes a method for estimating jitter for cognitive radio architectures at high sampling rates. The paper also examines the fixed-point implementation of the algorithm and its theoretical performance. However, it only includes theoretical and simulation; there are no measurement results to verify the method.

A modified way of calculating timing jitter using phase modulation (PM) noise measurements of high-speed digital clocks is presented in [12], which considers the frequency response of the jitter analyzer, providing a more accurate map.

Sampling clock jitter effects in digital-to-analog converters are described in [13]. A formula for the output error power because of the sampling clock jitter for a sinusoidal input is derived and verified by numerical simulations, and its spectrum characteristics are shown. Also, its effects on ADC SNR are clarified by numerical simulation as follows: (1) When the total noise power outside as well as inside the signal band is taken into account, the ADC SNR remains almost constant regardless of the sampling jitter. (2) However, when an analog low-pass filter follows the ADC and only the noise power inside the signal band is considered, the ADC SNR degrades as jitter increases and the input signal frequency becomes higher. Thus the sampling clock jitter is serious for the high-speed ADC.

The effect of accumulated jitter on the estimation of SNR is analyzed in frequency domain [14] and time domain [15]. Both theoretical analysis and simulation show the accumulated jitter on SNR estimation. However, none of the papers mentioned provide a validation of measurement results.

It can be seen that many methods were proposed to investigate clock jitter and its effect on ADC testing. The methods in the literature mentioned above suffer from one or more of the following issues: (1) the model of clock jitter is not realistic; (2) only theoretical derivation and simulation were given, and none of those papers applied measurement results to verify their analysis; and (3) methods to estimate the correct SNR, which means to recover the correct SNR from the effect of clock jitter, were not given. So it is required to develop a test method that can obtain the correct SNR in the presence of clock jitter. Moreover, the jitter model must be realistic, and the method should be verified by measurement results and can be applied for ADC testing in the industry.

In this chapter, a test method that can estimate the SNR with clock jitter is presented. The model of the clock jitter is given as the presence of both accumulated jitter and random clock jitter. The jitter properties and their effect on ADC testing are investigated. For the first time, the restriction on the sampling clock is removed so that the sampling clock can be accumulated jitter or random jitter or the superposition of the two kinds of jitter. Both simulation results and measurement data from the industry validate the functionality of the proposed method. This makes the testing of ADC using a low quality of sampling clock possible.

The remaining chapter is arranged as follows: Section 2 describes clock jitter and its effect on ADC testing. The new test method is presented in Section 3. Sections 4 and 5 provide simulation and measurement results. Section 6 concludes the chapter.

5.2 Problem Statement

In ADC spectral testing, a pure sine wave is fed into the ADC under test. DFT is performed to the ADC output, and then the spectrum is analyzed to obtain the spectral performance.

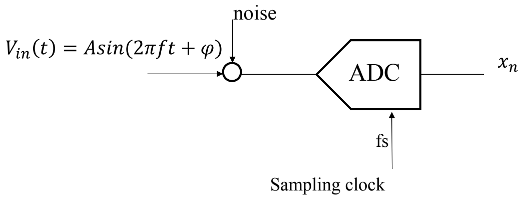


Figure 5.1 ADC test setup

Figure 5.1 shows the test setup for the ideal spectral test method [16]. Here, *ideal* means the test setup satisfies the ADC spectral test requirements in IEEE standards [1]. $V_{in}(t)$ is the input signal, and noise is the additive noise in Figure 5.1.

The expression of the pure sine wave input with amplitude *A*, offset V_{os}, frequency *f*, and initial phase φ is shown in equation (5.1):

$$V_{in}(t) = V_{os} + A\sin\left(2\pi ft + \varphi\right) \tag{5.1}$$

Let us suppose that ADC under test is N bit and the sampling rate is $f_s=1/T_s$. If the clock is ideal, the sampling instant at the nth sample is

$$t_n = nT_s \qquad n = 0,1,2...M-1 \tag{5.2}$$

Here, M is the number of sampled points that are usually collected for the spectral testing of an N-bit ADC. The analog interpretation of the ADC output can be expressed as

$$x[n] = V_{os} + A\sin\left(2\pi fnT_s + \varphi\right) + h.d + w[n] \tag{5.3}$$

w[n] is the superposition of additive noise and quantization noise at the nth sample, and $h.d = \sum_{h=2}^{H} A_h \cos(2\pi hnT_s + \varphi_h)$ is high-order harmonics where A_h and φ_h, respectively, contain the information of amplitude and phase of h^th harmonic of ADC.

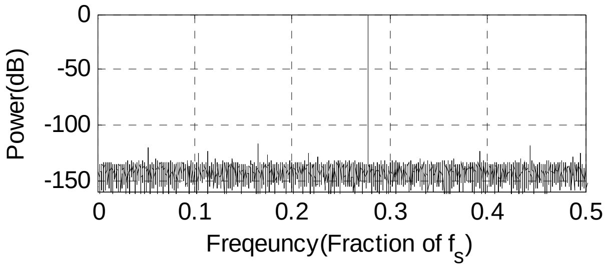


Figure 5.2 Spectrum of a coherent ADC output

If the input signal is sampled coherently, which means that the data record length contains exactly an integer number of cycles, the spectrum of the ADC output is like the one shown in Figure 5.2.

In reality, the clock may be unstable or the sampling frequency can drift because of temperature fluctuation. Clock jitter is defined as the difference between the real clock edge and the ideal clock edge. As shown in Figure 5.3, the ideal clock period is T_s, and the sampling instant is at time 0, T_s, $2T_s$, and so on. The clock is however affected by some amount of jitter. Then there are deviations between the real sampling edges and the ideal ones. Let δ_n be the clock jitter at time nT_s, and the deviations δ_0, δ_1, and δ_2 shown in Figure 5.3 are the jitter of the clock. Because of that jitter, the sampling instants are then shifted to $0+\delta_0$, $T_s+\delta_1$, and $2T_s+\delta_2$.

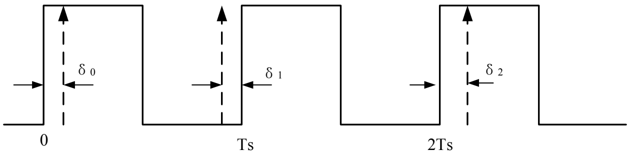


Figure 5.3 Clock jitter

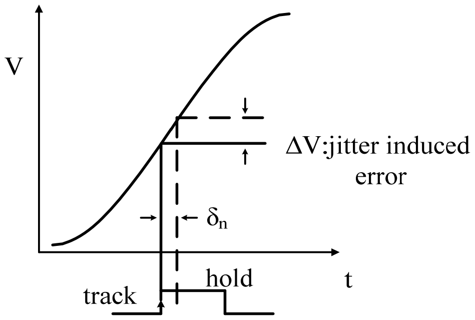


Figure 5.4 Clock jitter on ADC output

The jitter effect on the ADC sampled data will be shown in the following section. For ADC, the sample is taken when the ADC's track and hold goes into the hold state. As shown in Figure 5.4, the input sine wave should be sampled at the rising edge of the sampling clock. Assuming that the sampling instant is nT_s, the uncertainty in clock shifts the sampling instant to $nT_s+\delta_n$, causing uncertainty ΔV in the input. Thus, there is uncertainty in the ADC output, and the value of the uncertainty at the output is proportional to the input slope.

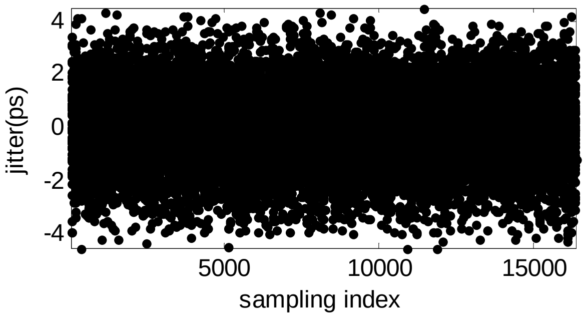


Figure 5.5 Example of random clock jitter

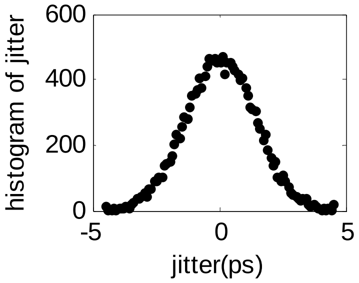


Figure 5.6 Histogram of the random clock jitter in Figure 5.5

5.2.1. Random Clock Jitter

Random clock jitter usually follows Gaussian distribution. In this chapter, random clock jitter is modeled as a variable with zero mean and a certain value variance. Figure 5.5 shows an example of random clock jitter generated in MATLAB: 16,384 points with a mean of zero and an RMS of 1.7 ps. Figure 5.6 shows the histogram of the clock jitter shown in Figure 5.6. It can be seen from the histogram that the generated random clock jitter roughly follows Gaussian distribution. As the random clock jitter is Gaussian distributed, its spectrum is like the one of white noise, as shown in Figure 5.7.

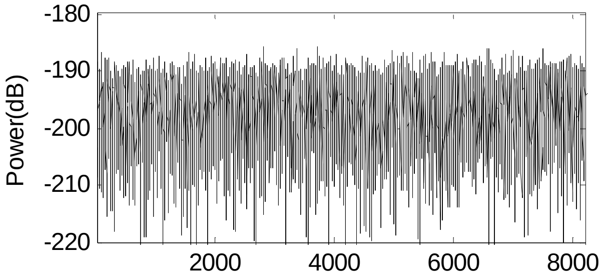


Figure 5.7 Spectrum of the random clock jitter shown in Figure 5.5 (relative to s^2)

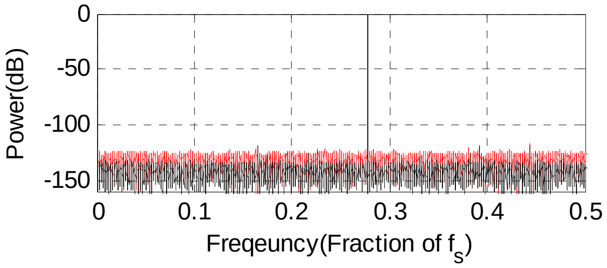


Figure 5.8 Spectrum of an ADC output without random clock jitter (black) and spectrum of the ADC output with random clock jitter (red)

If an ADC is tested using a clean sampling clock, the spectrum of the ADC output can be seen as the black spectrum in Figure 5.8. While there is random clock jitter in the sampling clock, the spectrum of the ADC output can be seen as the red one in Figure 5.8. Except for the clock jitter, the other settings of the red spectrum are the same as those of the black spectrum, such as input amplitude and frequency, sampling frequency, ADC performance, and so on. It can be seen that the harmonic values of the two spectra are the same. The only difference between them is that the noise floor of the red one is higher than that of the black one. One can also analyze the expression of DFT on ADC output data with clock jitter, and it can be proved that random clock jitter increases the noise floor of the ADC output spectrum. If the SNR is calculated directly from the red spectrum, the value is smaller as jitter is treated as noise in this case.

5.2.2. Accumulated Clock Jitter

Accumulated clock jitter is an accumulation of random clock jitter. As it is an accumulation, it is much more difficult to analyze the property in the means of distribution. Figure 5.9 shows an example of accumulation clock jitter. It is the accumulation of random clock jitter shown in Figure 5.5, which passes through a low-pass filter. It can be seen that the accumulated clock jitter is much larger than the random clock jitter as it accumulates in time domain. Figure 5.10 shows the histogram of the clock jitter shown in Figure 5.9. It cannot be concluded what kind of distribution an accumulated jitter follows. For another set of a given random clock jitter used for accumulation, the shape of accumulated clock jitter and its histogram will be totally different. In frequency domain, the spectrum of the accumulated jitter shown in Figure 5.9 is also shown in Figure 5.11. It can be seen that the spectrum is not white and there is power leakage caused by accumulation. The effect of accumulated clock jitter on the ADC output spectrum is shown in Figure 5.12. The black one is the spectrum of the ADC output with a clean clock, and the red one is that with accumulated clock jitter. It can be seen that there is power leakage in both the fundamental and the noise increasing in the noise floor.

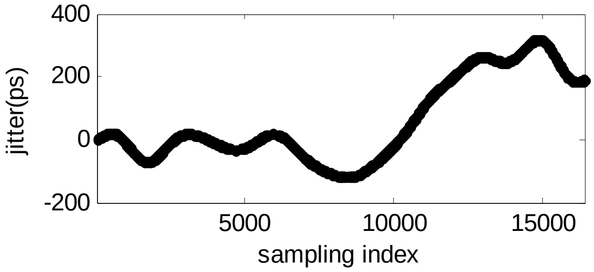


Figure 5.9 Example of accumulated clock jitter

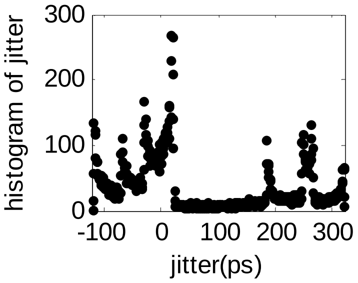


Figure 5.10 Histogram of accumulated clock jitter in Figure 5.9

5.2.3 Jitter Effect versus Noncoherency in ADC Output

Because of the increasing frequency and data rate, jitter can be the ultimate limit of the performance in some applications. Random clock jitter increases the noise floor and then reduces the estimated SNR value. When accumulated jitter dominates the clock jitter, even the collected ADC output samples are set to be coherent, and there is still power leakage as shown in the red spectrum in Figure 5.12. The power leakage looks like the one generated as noncoherency. However, the leakage cannot be removed by traditional methods, which are effective in noncoherency.

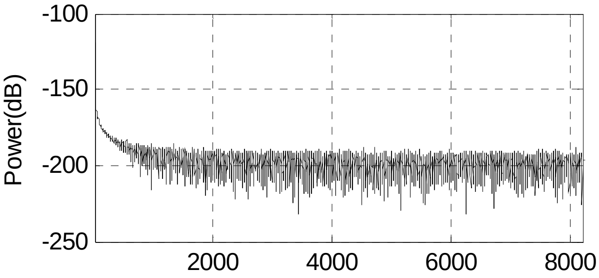


Figure 5.11 Spectrum of accumulated clock jitter in Figure 9 (relative to s^2)

FIRE [2], a state-of-the-art method used for dealing with the noncoherency problem, is applied to the ADC output with the spectrum shown as the red one in Figure 5.12. The new spectrum after FIRE is shown as the blue spectrum in Figure 5.13, while the red one is the same as that in Figure 5.12. It can be seen that the power leakage caused by clock jitter cannot be removed by the conventional noncoherency method. Then the SNR calculated from this spectrum is not correct. A method needs to be developed to calculate an SNR without the jitter effect.

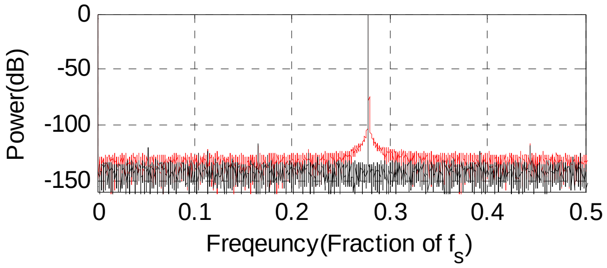


Figure 5.12 Spectrum of an ADC output without accumulated clock jitter (black) and spectrum of an ADC output with accumulated clock jitter (red)

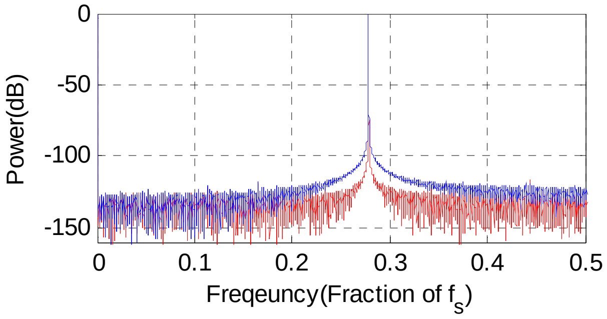


Figure 5.13 Spectrum of an ADC output with accumulated clock jitter (red) and spectrum of an ADC output with accumulated clock jitter + FIRE

5.2.4 Jitter Effect in Time-Domain ADC Output

From the previous analysis, it can be seen that random clock jitter increases the noise floor of the ADC output and accumulated clock jitter induces power leakage that cannot be removed through conventional methods. Both types of jitter reduce the estimated value of SNR. In ADC testing, one can tell that there is clock jitter through the SNR estimation results the data collected in the ADC output. This section briefly presents the jitter effect on the time-domain ADC output.

In paper [7], it is known that if we have two segments of the ADC output with the same conditions, which means the input with the identical initial phase, and we collect the same number of points for the two segments, then the deterministic terms of the two segments, such as fundamental and harmonics, are the same. The only difference is the random terms, such as additive noise and clock jitter. If we take subtraction of the two segments point by point, the difference of the two segments can be shown as in Figure 5.14 and Figure 5.15. The identical deterministic terms are removed, leaving the noise and clock jitter effect.

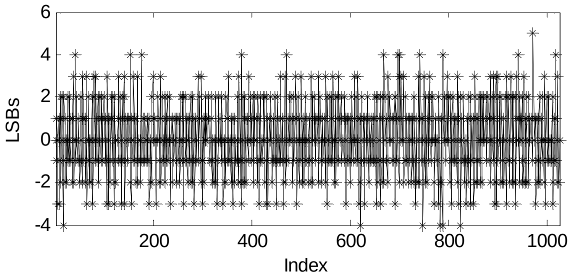
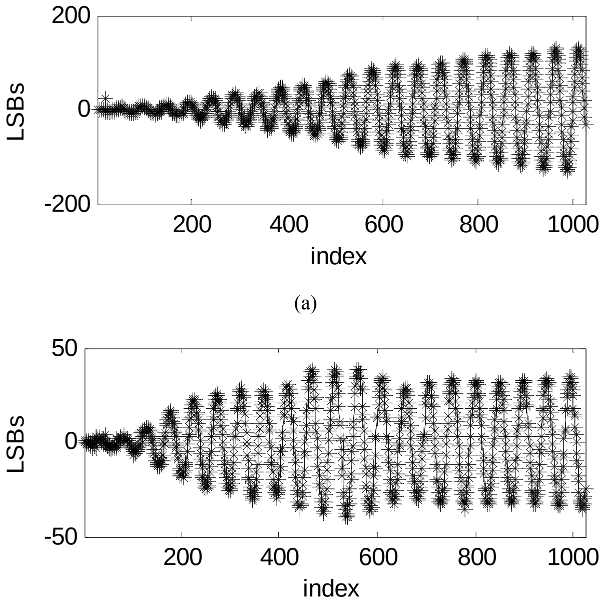


Figure 5.14 Difference of two segments with accumulated clock jitter



(a)

(b)

Figure 5.15 Difference of two segments with accumulated clock jitter

In Figure 5.14, as there is only random clock jitter, the difference of the two segments looks like white. The values of the difference are around several LSBs (least significant bits). Figure 5.15 shows two cases of two segments' difference with accumulated clock jitter. As jitter may accumulate in a different trend for two segments, the difference may grow up and down in different shapes. This will be explained in detail in Section 3.

5.3 Proposed Method

In the proposed method, the data record length of the ADC output is set to be M, where M is the number of points usually used to test the spectral performance for an N-bit ADC. Then the first half points of the ADC output are broken into small segments. Other sets of small segments are searched in the second half points to be matched best with segments in the first half points. Subtraction is applied between the matched pairs. The fundamental of the segments' difference is identified and removed. Jitter and noise separation is applied to the residue, and noise power is calculated to compute the SNR. The detail of the proposed method is shown as follows:

Equation (5.3) shows the expression of the ADC output when the clock is clean. When there is clock jitter, (5.3) can be modified as

$$x[\mathrm{n}] = V_{os} + A\sin\left(2\pi f\left(nT_s + \delta_n\right) + \varphi\right) + h.d + w[\mathrm{n}] \qquad (5.4)$$

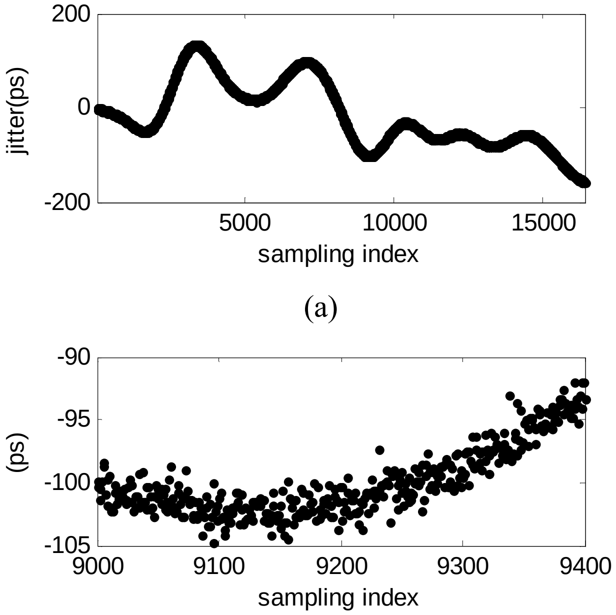
where δ_n is the clock jitter at time nT_s.

5.3.1 Jitter Model

As accumulation jitter accumulates in an unpredictable trend, defining it quantitatively is hard. So RMS alone is not enough to describe accumulative jitter. In this chapter, we use segmentation to express the jitter model. It must be stated that, in reality, the jitter sequence cannot be observed from the clock. All we can see is the actual sampling instant, which is $nT_s+\delta_n$. In time domain, the jitter drifts as a fluctuation. We model the jitter as the superposition of accumulated clock jitter and random clock jitter, as shown in equation (5.5):

$$\delta[\mathrm{n}] = \delta_c[\mathrm{n}] + \delta_r[\mathrm{n}] \qquad (5.5)$$

Here, δ_r[n] is the random jitter at sampling instance nT_s, and δ_c[n] is the accumulated clock jitter at sampling instance nT_s. Although the drifting of accumulated jitter cannot be predicted, we can model the accumulated jitter by small segments.



(a)

(b)

Figure 5.16 Time domain clock jitter (a) and a portion of the curve in (a)

Figure 5. 16(a) shows an example of δ_n. If we zoom a small segment of the curve, it can be shown in Figure 5. 16(b). It is a curve with an initial value, slope and some random terms. However, we do not know the exact value of these parameters. We can break the long jitter sequence into small segments. For each small segment, the expression of the accumulated jitter can be expressed as these three parameters as shown in equation (5.6).

$$\delta_c[\mathrm{n}] = \delta_{cc} + \delta_{cl} \times n + \delta_{cr}[\mathrm{n}] \qquad (5.6)$$

In equation (5.6), the accumulated jitter is modeled as three terms: constant offset δ_{cc}, slope δ_{cl}, and local random term $\delta_{cr}[n]$. The constant offset is the initial value of each segment. The slope is the slope of the endpoint line of the segment. The local random term is the distance of the accumulated jitter from the endpoint line. Simply, the small segmented accumulated jitter can be modeled as a straight line plus some random term.

This chapter aims to estimate the correct spectral performance without the jitter effect, so these jitter parameters are not estimated here. This model helps remove the jitter effect from the ADC output, and then the correct noise power of the ADC under test can be calculated.

5.3.2 Segmentation Pairs

The first M/2 points in (5.4) are broken into small segments—$x_1[n]$, $x_2[n]$,…,$x_s[n]$—each segment has m points, as shown in Figure 5.17. Here, m is a small number, and it can be set to 64 or 128 in this chapter. For each segment $x_i[n]$, i=1,2,…,s, find another segment $x'_i[n]$ in the second M/2 points, letting the initial phase of $x'_i[n]$ match that of the $x_i[n]$ best.

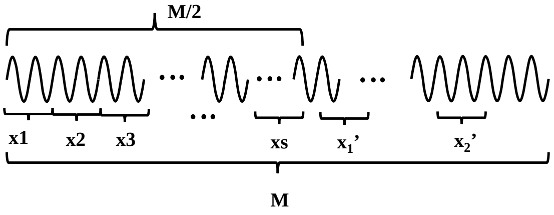


Figure 5.17 Time-domain ADC output

The expression of $x_i[n]$ is shown in (5.7):

$$x_i[\text{n}] = V_{os} + A\sin\left(2\pi f\left(nT_s + \delta[\text{n}]\right) + \varphi\right) + h.d + w[\text{n}] \qquad \text{n=0,1,2,\ldots m–1} \qquad (5.7)$$

Compared with the value of sampling instance nT_s, the value of clock jitter $\delta[n]$ is a relatively small value. We can apply Taylor expansion to the fundamental part in (5.7). Then equation (5.7) can be written as

$$x_i[\mathrm{n}] = V_{os} + A\sin\left(2\pi fnT_s + \varphi\right) + 2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\delta[\mathrm{n}] + h.d + w[\mathrm{n}] \qquad (5.8)$$

And x’ᵢ[n] can be written as

$$x_i^{'}[\mathrm{n}] = V_{os} + A\sin\left(2\pi fnT_s + \varphi^{'}\right)2\pi fA\cos\left(2\pi fnT_s + \varphi^{'}\right)\delta_n^{'} + h.d^{'} + w^{'}[\mathrm{n}] \qquad (5.9)$$

If we subtract xi’[n] from x[n] point by point, it can be seen that the offset V_{os} is canceled. As the initial phases of the two segments are nearly identical, the fundamentals of the two segments are approximately equal to each other, and the harmonic difference of the two segments (h.d-h.d’) can be ignored. Even if the initial phases of the two segments are not the same, there is a small amount fundamental residue in the difference. The difference of the two segments can be expressed as

$$x_i[\mathrm{n}] - x_i^{'}[\mathrm{n}] \approx F_{res} + 2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\left(\delta[\mathrm{n}] - \delta^{'}[\mathrm{n}]\right) + N[\mathrm{n}] \qquad (5.10)$$

where F_{res} is the fundamental residue, $A\sin\left(2\pi fnT_s + \varphi\right) - A^{'}\sin\left(2\pi fnT_s + \varphi^{'}\right)$, and N[n] is the difference of the noise (N[n]=w[n]-w[n]’) with zero mean, and the variance is twice that of w[n].

5.3.3 Fundamental Removal for Segments

With the jitter model shown in (5.5) and (5.6), equation (5.10) can be rewritten as

$$x[\mathrm{n}] - x^{'}[\mathrm{n}] \approx F_{res} + 2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\left(\delta_{cc} + \delta_{cl}\times n + \delta_{cr}[\mathrm{n}] - \left(\delta_{cc}^{'} + \delta_{cl}^{'}\times n + \delta_{cr}^{'}[\mathrm{n}]\right)\right) + N[\mathrm{n}] \qquad (5.11)$$

As δ_{cc} is a constant value for each small segment, the difference of δ_{cc} and $\delta_{cc}^{'}$ is also constant for each segment pair. This difference is modulated on the fundamental, making $2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\left(\delta_{cc} - \delta_{cc}^{'}\right)$ also a fundamental term. Working with F_{res}, it forms a new fundamental residue, F'_{res}. Let $\Delta_{cr}[\mathrm{n}] = \delta_{cr}[\mathrm{n}] - \delta_{cr}^{'}[\mathrm{n}]$ and $\Delta_{cl}[\mathrm{n}] = \delta_{cl}[\mathrm{n}] - \delta_{cl}^{'}[\mathrm{n}]$. Then (5.11) can be written as

$$x[\mathrm{n}] - x^{'}[\mathrm{n}] \approx F_{res}^{'} + 2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\Delta_{cr}[\mathrm{n}] + 2\pi fA\cos\left(2\pi fnT_s + \varphi\right)\Delta_{cl}[\mathrm{n}] + N[\mathrm{n}] \qquad (5.12)$$

Then the fundamental in (5.12) is identified and removed, and the left residue ei[n] only contains jitter and noise information as shown in (5.13):

$$e_i[\text{n}]= 2\pi fA\cos\left(2\pi fnT_s+\varphi\right)\Delta_{cr}[\text{n}]+2\pi fA\cos\left(2\pi fnT_s+\varphi\right)\Delta_{cl}[\text{n}]+N[\text{n}] \qquad (5.13)$$

We know that the clock jitter is a slow drifted variable. If m is large, the clock jitter will accumulate. Taking m=1024, for example, the $e_i[n]$ is shown in Figure 5.15. The horn shape is mainly caused by the accumulated jitter, which is a slow drift term. However, if we set m to be a small number, the time for data acquisition of $x_i[n]$ is too small that the accumulation of the clock jitter is not serious. Taking M=1024 and m=64, for example, combining the s (s=M/m=16) segments and $e_i[n]$ (i=1,2,…,16) together, the new e[n] is shown in Figure 5.18.

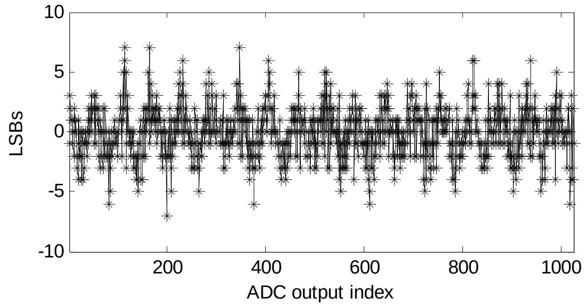


Figure 5.18 e[n] when m=64, M=1024

5.3.4 Jitter and Noise Separation

For this step, the residue only contains noise information of the ADC under test and some random clock jitter residue. $N'[\text{n}]=2\pi fA\cos(2\pi fnT_s+\varphi)\Delta_{cl}[\text{n}]+N[\text{n}]$ is a random term with variance proportional to the m length, and the residue of all the segment pairs can be written as

$$e[\text{n}]= 2\pi fA\cos\left(2\pi fnT_s+\varphi\right)\Delta_{cr}[\text{n}]+N'[\text{n}] \qquad \text{n=0,2,…M–1} \qquad (5.14)$$

Now, the residue contains a random jitter modulated with fundamental and random noise. Then the jitter and noise separation method in [7] can be used to solve noise power and then calculate the value of SNR without the jitter effect. The variance of $N'[\text{n}]$ can be calculated as

$$P_{n1} = \frac{3 \times \sum_{n=0}^{M-1} e^{2}[\mathrm{n}] - 4 \times \sum_{n=0}^{M-1} \left(e[\mathrm{n}]\cos(2\pi f n T_{s} + \varphi) \right)^{2}}{2M} \qquad (5.15)$$

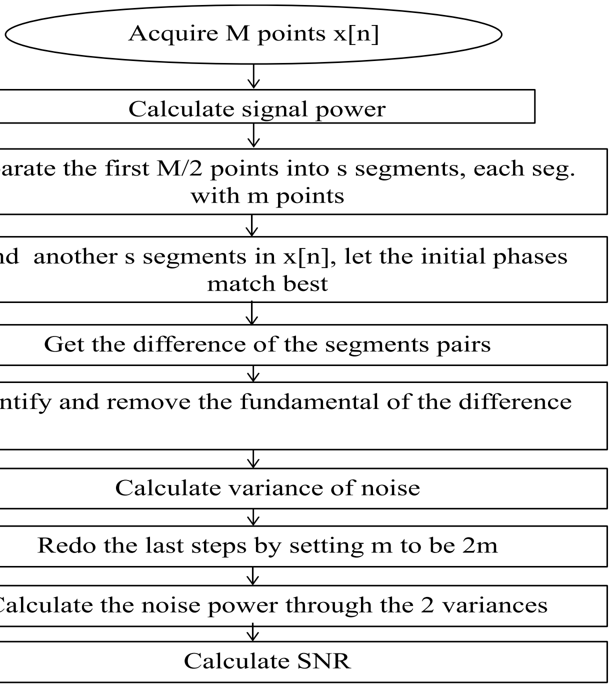

Acquire M points x[n]
Calculate signal power
Separate the first M/2 points into s segments, each seg. with m points
Find another s segments in x[n], let the initial phases match best
Get the difference of the segments pairs
Identify and remove the fundamental of the difference
Calculate variance of noise
Redo the last steps by setting m to be 2m
Calculate the noise power through the 2 variances
Calculate SNR

Figure 5.19 Algorithm flowchart

5.3.5 Noise Power Correction

It must be stated that the variance of $N'[\mathrm{n}]$ varies with the m value as the accumulated jitter is like a random walk, which is also called the Wiener process. Then the variance of $N'[\mathrm{n}]$ calculated in the equation is different when m is different. As segment size doubles，the variance of $N'[\mathrm{n}]$ doubles. Another variance P_{n2} is obtained when m is doubled. However, the variance of N[n] should be the same as it is twice that of the ADC’s noise, which should be constant. Then the noise power P_n, which is also the variance of N[n], can be calculated as $2P_{n1}–2P_{n2}$.

The SNR is calculated as

$$SNR = 10\log_{10}\frac{A^2/2}{P_n} \tag{5.16}$$

Now the correct SNR of the ADC under test in the presence of a clock jitter is obtained. The algorithm for this method is described in Figure 5.19.

5.4 Simulation Results

In this section, the proposed SNR estimation method is validated by the simulation data generated in MATLAB. A 14-bit ADC is modeled in MATLAB as a set of transition levels. The nonlinearity error is modeled as a set of Gaussian distributed variables with zero mean and a certain standard deviation. The additive noise is introduced with the input. It is Gaussian distributed random noise with zero mean and a standard deviation of 0.75 LSB. The ADC was first tested with a sine wave that was coherently sampled, and the sampling clock is modeled as ideal. The value of the SNR obtained is considered the reference value. The same ADC is later fed with the same input signal. However, the sampling clock is modeled as clock jitter added to the ideal sampling instant. The sampling clock jitter is modeled as the superposition of an accumulated Gaussian distributed term, and another Gaussian distributed term a random error. Table 5.1 shows the parameter setting. And the input amplitude is set around –0.05 dB to almost cover the full scale of the ADC under test.

Table 5.1 Parameter setting of the simulation

f	0.9 MHz	M	2^15
fs	1 MHz	m	128

The amount of jitter added to the clock is relative to the ADC's quantization noise power.

The quantization noise power is calculated as

$$P_{qu} = \frac{q^2}{12} \tag{5.17}$$

In equation (5.17), q is 1 LSB.

For random clock jitter, let σr be the standard deviation, and the jitter-induced power for the ADC is calculated as

$$P_r = \frac{(2\pi fA\sigma_r)^2}{2} \tag{5.18}$$

While the random clock jitter–induced power is given as (5.18), it is much more difficult to determine the accumulated jitter power. As the accumulated jitter is obtained through the accumulation of a Gaussian distributed random noise, we use the power of the Gaussian term to indicate the amount of accumulated jitter in this chapter. It must be pointed out that the actual accumulated jitter power is far more than the Gaussian term power. Let the RMS of the Gaussian noise used for accumulation be σc, and the accumulated jitter power is given as

$$P_c = \frac{(2\pi fA\sigma_c)^2}{2} \tag{5.19}$$

5.4.1 Simulation Test Results

We add both random clock jitter and accumulated clock jitter to the sampling clock. The simulation results of SNR estimation are shown in Figure 5.20. The horizontal axes are the variance of the random jitter σ_r^2 and the Gaussian term σ_c^2 used to accumulate. As the jitter can accumulate, the actual jitter power is far larger than the one shown in the horizontal axes.

The red lines are the reference SNR of the ADC under test, and they are obtained when the input signal is sampled coherently and there is no clock jitter. The black lines are the SNR values

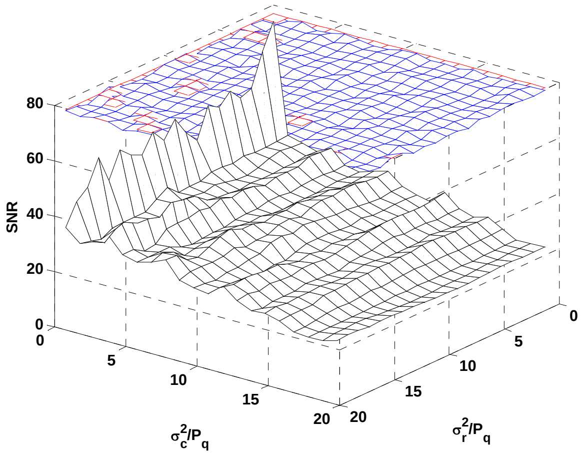


Figure 5.20 SNR simulation results in the presence of both types of jitter

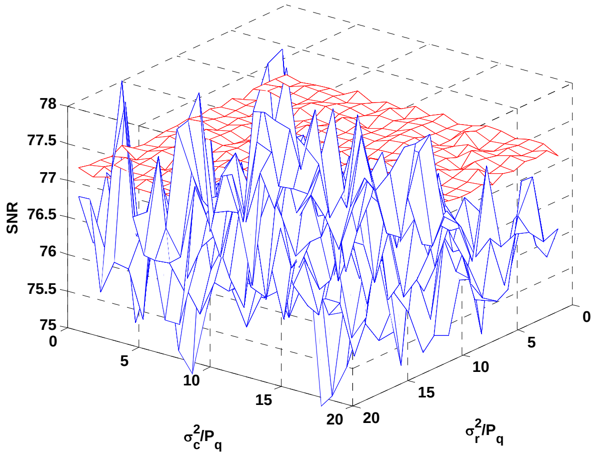


Figure 5.21 The reference SNR and the SNR using the proposed method in Figure 5.20

obtained from direct FFT when there is clock jitter. The proposed method, the estimated SNR with clock jitter, is shown as blue lines in the figure. It can be seen from the figure that as jitter amount increases, the direct estimated SNR decreases as jitter amount affects the noise power calculation. The effect of accumulated jitter is much more than that of random jitter. The SNR using the proposed method is close to the reference SNR. These two SNRs are shown in Figure 5.21. It can be seen that the reference SNR is around 77 dB, and the proposed SNR is 75 to 78 dB, which is much more accurate compared with the direct estimation.

5.4.2 Compared with the VTS14' Method

We first include only random clock jitter in the sampling clock to verify the validity of the proposed method on random jitter and compare it with the VTS14' method. And then both random jitter and accumulated jitter are added to compare the proposed methods with the VTS14' method.

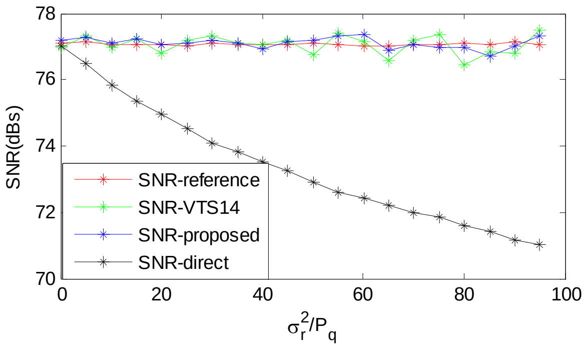


Figure 5.22 SNR simulation results with random clock jitter

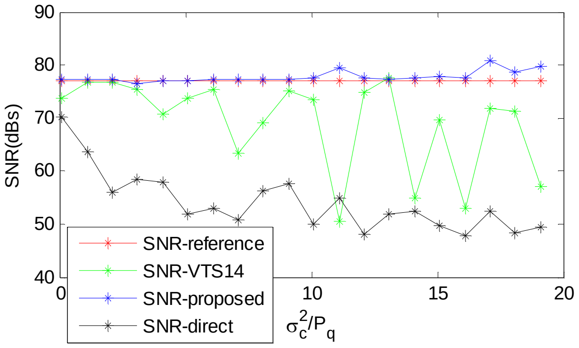


Figure 5.23 SNR simulation results with both random jitter and accumulated jitter

Figure 5.22 shows the simulation results when there is only random clock jitter. Twenty simulations are run with different amounts of jitter added. The horizontal axis shows the ratio of the Gaussian jitter power to the quantization noise power. There is only random jitter in the clock and no accumulated clock jitter. The red dots in Figure 5.22 are SNR estimated using a clean clock, which can be treated as a reference. The black dots are the SNR estimated directly from DFT when there is random clock jitter. It can be seen that as jitter amount increases, the SNR value decreases. The green ones are obtained from the VTS14' method, and the blue ones are obtained using the proposed method. It can be seen from the figure that both the VTS14' method and the proposed method work when there is only random jitter. And the SNR estimation is accurate under different jitter amounts using the proposed method. It must be stated that the power ratio 100 is so large that is not realistic. The simulation here shows that the proposed method works even if the jitter is that large.

Then the random jitter power is set to be 20 times that of the quantization power, and the accumulated jitter is increased from 0 to 20 times that of the quantization power to show the validity of the two methods. Figure 5.23 shows the simulation results: it can be seen that with clock jitter, the direct estimated SNR is far from accurate. When the jitter amount is small, the VTS14' method works, but the results are bad when the accumulated jitter amount increases, whereas the proposed method is accurate regardless of jitter amount.

5.5 Measurement Results

In this section, silicon measurement results are presented to validate the effectiveness of the proposed SNR estimation method. A commercially available 14-bit 1 Msps SAR ADC with an integrated 25 MHz RC oscillator is chosen for this experiment. The ADC input signal frequency is around 19 kHz, and it is fine-tuned to achieve coherent sampling. First, ADC SNR is measured with an “ideal” off-chip 25 MHz clock source. This is denoted as the “reference SNR.” Next, ADC SNR is measured with the on-chip jittery RC oscillator, which is denoted as the “direct estimated SNR.”

Then the proposed method is applied to the ADC output with the on-chip RC oscillator and the true ADC SNR is estimated, which is denoted as the “proposed estimated SNR.” This experiment is repeated on four devices, and the measurement results are shown in Table 5.2. It is clearly evident from the results that the proposed method provides an accurate estimation of the true ADC SNR in the presence of clock jitter.

Table 5.2 Measurement Results (unit: dBs)

	Reference SNR	Direct estimated SNR	Proposed estimated SNR
ADC_1	77.4	22.1	75.4
ADC_2	77.7	13.5	76.5
ADC_3	77.8	13.2	76.0
ADC_4	78.2	13.6	76.7

5.6 Conclusion

A new test method that accurately estimates the SNR of an ADC with sampling clock jitter was proposed. This relaxed the requirement to have a precise sampling clock for spectral testing. A new segment matching method was proposed to remove the accumulated jitter and random clock jitter. Correct noise power can be calculated without the effect of jitter. A more realistic jitter model was proposed and analyzed in this chapter. The collected ADC output was separated into small segments. By analyzing the difference of the matched segments pairs, correct noise information was calculated. Then the SNR was estimated without the jitter effect. The accurate functionality of the proposed method was presented using simulation results on 14-bit ADCs. This method was also compared with literature, proving that it also works even when there is only random clock jitter. The proposed method was also verified using commercially available ADC testing results. For the first time, there is a method working on accumulated clock jitter using industry data to prove the validation. This method can be readily used in applications where obtaining precise control over a sampling clock, such as BIST ADCs, is challenging.

References

[1] IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters, IEEE Std. 1241, 2010.

[2] S.K.Sudani, D.Chen, "FIRE: A Fundamental Identification and Replacement Method for Accurate Spectral Test Without Requiring Coherency," IEEE Transactions on Instrumentation and Measurement, vol.62, no.11, pp.3015-3025, Nov. 2013

[3] T. J. Yamaguchi, K. Ichiyama, H. X. Hou and M. Ishida, "A robust method for identifying a deterministic jitter model in a total jitter distribution," International Test Conference, Austin, TX, 2009, pp. 1-10.

[4] J. C. Hsu and C. Su, "BIST for Measuring Clock Jitter of Charge-Pump Phase-Locked Loops," IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 2, pp. 276-285, Feb. 2008.

[5] S. Shariat-Panahi, F. A. C. Alegria, A. Manuel, and A. M. D. Serra, "IEEE 1057 Jitter Test of Waveform Recorders," IEEE Transactions on Instrumentation and Measurement, vol. 58, pp. 2234-2244, Jul 2009.

[6] W. Kester, "The Data Conversion Handbook," Analog Device, Inc, pp. 5.73-5.75, 2004.

[7] L. Xu, D. Chen, "Accurate and efficient method of jitter and noise separation and its application to ADC testing," IEEE VLSI Test Symposium (VTS), pp.1-5, 13-17 April 2014.

[8] W. Minshun, C. Degang, and D. Jingbo, "Fast & accurate algorithm for jitter test with a single frequency test signal," in 2011 IEEE International Conference on Electrol Information Technology (EIT), 2011, pp. 1-5.

[9] Towfic, Z.J.; Shang-Kee Ting; Sayed, A.H., "Sampling clock jitter estimation and compensation in ADC circuits," in Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp.829-832, May 30 2010-June 2 2010

[10] Da Dalt, Nicola. "Effect of jitter on asynchronous sampling with finite number of samples." IEEE Transactions on Circuits and Systems II: Express Briefs, 51.12 (2004), pp. 660-664.

[11] Z. J. Towfic and A. H. Sayed, "Clock jitter estimation in noise," 2011 IEEE International Symposium on Circuits and Systems (ISCAS), Rio de Janeiro, 2011, pp. 1251-1254.

[12] Howe, D. A., and T. N. Tasset. "Clock jitter estimation based on PM noise measurements." Frequency Control Symposium and PDA Exhibition Jointly with the 17th European Frequency and Time Forum, 2003. Proceedings of the 2003 IEEE International. IEEE, 2003.

[13] Kurosawa, Naoki, et al. "Sampling clock jitter effects in digital-to-analog converters." Measurement 31.3 (2002), pp. 187-199.

[14] S.S.Awad. "The effects of accumulated timing jitter on some sine wave measurements." IEEE Transactions on Instrumentation and Measurement, 44.5 (1995), pp. 945-951.

[15] S.S.Awad, "Analysis of accumulated timing-jitter in the time domain," IEEE Transactions on Instrumentation and Measurement , vol.47, no.1, pp.69-73, Feb 1998.

[16] S. K. Sudani, L. Xu and D. Chen, "A Comparative Study of State-of-the-Art High-Performance Spectral Test Methods," IEEE Design & Test, vol. 32, no. 1, pp. 26-35, Feb. 2015.

CHAPTER 6. SUMMARY

In this dissertation, challenges in standard high-performance ADC spectral testing were discussed. Algorithms of accurate spectral testing without accurate instrumentation were presented. The first method eliminated the requirements of coherent sampling and in-range control of input amplitude of ADC spectral testing. The second method relaxed the conditions of precise control over amplitude and frequency. Furthermore, it obtained linearity performance in the meantime without additional consumption. The third method allowed spectral testing with an imprecise sampling clock. The method provided noise and jitter separation, which can be used for low-cost jitter characterization. The fourth method was proposed to test ADC spectral performance when there is accumulated clock jitter. This method can be used in on-chip ADC testing and ADC testing with low-cost instrumentations. Both simulation and measurement results were used to verify these methods. All of the proposed algorithms aimed to relax the stringent requirements of spectral testing setup. Making the setup less stringent can reduce the requirements and the cost of the testing instruments. Furthermore, the algorithms can be used for BIST applications.