Built-In Self-Test for the Analysis of Mixed-Signal Systems

By

George Joseph Starr

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
May 14, 2010

Keywords: Built-In Self-Test, Mixed-Signal, Selective Spectrum Analysis, Multiplier, CORDIC, Test Pattern Generator, Output Response Analyzer

Copyright 2010 by George Joseph Starr

Approved by

Charles E. Stroud, Chair, Professor of Electrical and Computer Engineering
Victor P. Nelson, Professor of Electrical and Computer Engineering
Foster F. Dai, Professor of Electrical and Computer Engineering

Abstract

A new Built-In Self-Test technique called Selective Spectrum Analysis has been developed for the measurement of analog characteristics in mixed-signal circuits. This design utilizes digital components to generate and collect analog waveforms to measure both signal strength and phase shift through an analog circuit. These measurements can be used to determine the performance of the analog circuit. However, this design still requires a practical approach to be able to quickly and accurately obtain measurements on-chip.

This thesis proposes improvements to the implementation of this Built-In Self-Test design. These improvements allow for the calculation of the results of these analog measurements on-chip. This thesis also introduces a program to automatically generate the hardware description language model required to fully implement this design in any Application Specific Integrated Circuit or Field Programmable Gate Array.

Acknowledgements

I would like to thank Dr. Charles Stroud for his constant support and advice throughout my Master's studies. His guidance helped me become a better engineer by teaching me how to logically analyze an unknown behavior and turn it into a known behavior. The knowledge he imparted to me will be invaluable in my future career and for this, I am most grateful. I would also like to thank Dr. Nelson for his instruction in digital design and ASIC fabrication. I would like to thank Dr. Dai for his support in this project and his expertise in the field of analog electronics. I would like to thank these three professors for serving as my committee members. Without their contributions, the work like that presented in this thesis would not be possible.

I would also like to especially acknowledge my colleague Jie Qin whose continued research in Built-In Self-Test acts as a pillar of inspiration to future BIST researchers. I would also like to thank Bradley Dutton and Joseph Cali for their assistance with both the realization of this research and their support throughout my Masters degree. I would also like to thank Mary Pulukuri for her experience in BIST research. I would also like to thank our undergraduate researchers Neil, Alex and Eric for their contributions to my research. Most of all, I would like to thank my wife, Kim, whose constant support, sacrifice and encouragement allowed me realize a dream and pursue graduate school.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog to Digital Converter
ASIC	Application Specific Integrated Circuit
ATE	Automated Test Equipment
BIST	Built-In Self-Test
CCU	Coordinate Rotation Digital Computer Calculation Unit
CORDIC	Coordinate Rotation Digital Computer
DAC	Digital to Analog Converter
DDS	Direct Digital Synthesis
DFT	Design For Testability
DUT	Device Under Test
FFT	Fast Fourier Transform
GCD	Greatest Common Denominator
HDL	Hardware Description Language
IC	Integrated Circuit
IIP3	Input Referred I Third Order Intercept Point
IM3	Third Order Intermodulation
IMD	Intermodulation Distortion
IMP	Integer Multiple Period
IP3	Third Order Intercept Point
LSB	Least Significant Bit

LUT	Look-Up Table
MAC	Multiplier Accumulator
MSB	Most Significant Bit
NCO	Numerically Controlled Oscillator
NF	Noise Figure
OIP3	Output Referred Third Order Intercept Point
ORA	Output Response Analyzer
PAC	Phase to Amplitude Converter
PCB	Printed Circuit Board
SINAD	Signal to Noise and Distortion
SNR	Signal to Noise Ratio
SSA	Selective Spectrum Analysis
THD	Total Harmonic Distortion
TPG	Test Pattern Generator
VHDL	Very High Speed Integrated Circuit Hardware Description Language

Chapter One

Introduction

The construction of high frequency analog circuits is riddled with difficulties. There are a variety of factors that can determine the quality of an analog Integrated Circuit (IC). These factors can include defects and non-uniform densities in the silicon wafer that cause transistors to operate differently as well as poor design and component choice [1]. Only some of these factors are predictable, therefore there is a need to test the specification of an analog IC to guarantee that it will operate as expected [2].

In digital electronics, there is an expanding field of techniques dedicated to developing self-testing and self-correcting algorithms for digital devices. The most effective of these derived methods is a technique known as Built-In Self-Test, where the testing components are located on-chip, eliminating the need for bulky and expensive test equipment [3]. A mixed-signal Built-In Self-Test approach has been proposed to measure the analog characteristics of a circuit [4]-[12]. This technique utilizes Direct Digital Synthesis (DDS) to generate test patterns and Selective Spectrum Analysis to collect measurement results. Discussed in this thesis are additions and improvements to this Built-In Self-Test (BIST) approach to measure the characteristics of an analog circuit incorporated within a mixed-signal design.

1.1 Built-In Self-Test

To keep up with the ever increasing complexity of electronics, the field of circuit testing has had to evolve. Design for Testability (DFT) techniques were developed "to improve the testability of a circuit by including additional circuitry that improves controllability and observability of the circuit under test" [3]. Traditionally, circuits were tested using Automated Testing Equipment (ATE) which would apply a series of inputs to the IC's pins and monitor the output pins for the correct response. This technique of fault testing limits the observable faults to only those that can be directly stimulated from the IC's pins. This shortcoming led to the idea of placing the testing equipment within the silicon device to perform a more thorough and sometimes shorter test sequence [3].

Built-In Self-Test (BIST) is any method for a circuit to test itself to detect faults or invalid responses without the use of external test equipment [3]. The typical BIST architecture contains two essential components used for testing and two additional features used to facilitate testing. The two essential components are the Test Pattern Generator (TPG) and the Output Response Analyzer (ORA). The TPG produces a sequence of test patterns to stimulate the Device Under Test (DUT). The ORA collects the output response of the DUT and translates it into a pass/fail indication [3]. The two secondary components used to facilitate testing are the BIST controller and isolation circuitry. The BIST controller acts to orchestrate the testing procedure by controlling the testing sequence. The isolation circuitry is used to isolate the DUT from the normal inputs so that the TPG can stimulate the DUT with a specific test sequence [3]. This approach is illustrated in Figure 1-1.

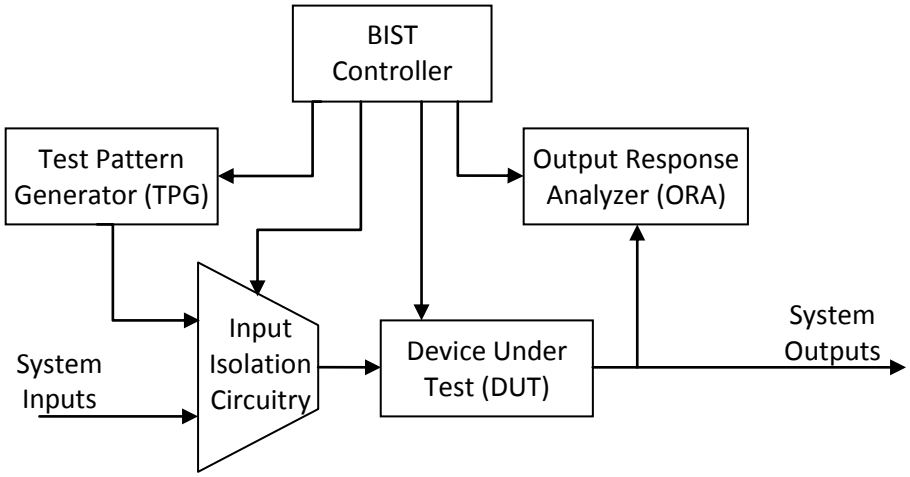

Figure 1-1: Basic BIST Architecture [3]

1.2 Built-In Self-Test for Analog Components in Mixed-Signal Systems

A mixed-signal IC allows developers to combine many different analog and digital systems into one IC. This increases the simplicity of integrating the original design, but the DFT process becomes more difficult. More components are being placed on ICs as the area required for circuit design is shrinking due to improved fabrication techniques, but each added component may require a different DFT technique to verify functionality [1]. These difficulties become even more exaggerated when one considers the introduction of mixed-signal architecture into the IC design. Traditional digital BIST architectures are designed to generate test patterns of distinct ones and zeros to apply to the inputs of the DUT. The output of the circuit is then monitored and compared to the expected digital result [3]. Analog circuits do not operate under such simplified binary principles; therefore a modified approach is required to verify the proper operation of the device.

To perform the test procedure, a TPG is used to produce a series of test patterns that are applied to the DUT. These test patterns are designed to exercise specific aspects of the DUT to test for fault conditions [3]. When testing an analog circuit, the TPG needs to be able to create an analog waveform to test the analog characteristics of a circuit. One method for easily achieving this waveform is a procedure called Direct Digital Synthesis (DDS) [17].

In DDS, a Numerically Controlled Oscillator (NCO) is used in conjunction with a Digital to Analog Converter (DAC) to generate a waveform from a frequency control value and starting phase. The NCO consists of a phase accumulator and a Phase to Amplitude Converter (PAC) [17]. The phase accumulator continuously accumulates the frequency control value at a periodic interval defined by a system clock frequency. This accumulated value is passed to the PAC, which converts the accumulator value to a magnitude value that corresponds to a phase location in the sinusoidal waveform. The output of the PAC is then transmitted to the DAC to generate the analog waveform for testing. Since the phase information is updated every clock cycle, the waveform will appear quantized. An additional filter is needed to smooth this quantized waveform. The illustration in Figure 1-2 shows a visual representation of how a TPG is formed using this method [17]. An illustrated example of the output of each stage is also represented in this figure.

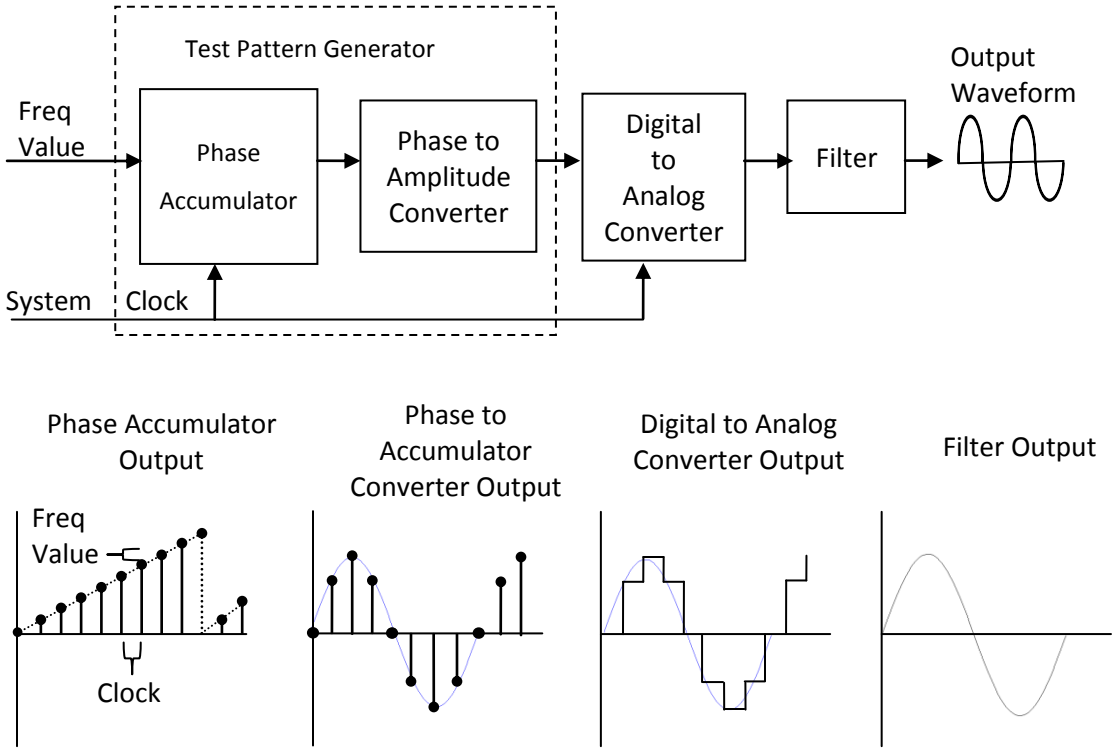

Figure 1-2: DDS-based Analog Test Pattern Generator [17]

The goal of the ORA is to collect the response from the DUT after it has been stimulated with the output of the TPG. After the response has been collected, it can be interpreted into a pass/fail indicator [3]. For analog BIST measurements however, the results will not be identical with every test. Noise, temperature, humidity, and external electromagnetic radiation sources will affect the performance and response of an analog system [20]. Therefore it becomes necessary to interpret the results and compare these observed measurements to a goal rather than a distinct binary response to determine a pass/fail indicator.

To convert this analog waveform into a digital signal, an Analog to Digital Converter (ADC) can be used. To collect the data points from the digitized input, the ORA implements two multiplier/accumulators (MAC) pairs. These two MACs can be

used to extract the in-phase and out-of-phase components of a single frequency within this measured signal [4]. Once this data has been collected, the result can be compared to a set of specifications for that test to generate a pass/fail indicator. This approach to the ORA is illustrated in Figure 1-3.

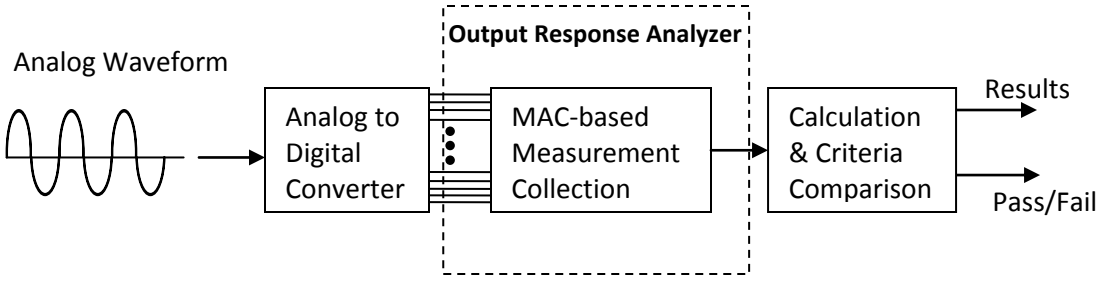

Figure 1-3: MAC-based Analog Output Response Analyzer Approach

1.3 Selective Spectrum Analysis

To perform a test on an analog circuit, a tone must be driven into the circuit and the response measured. To measure this response, a traditional spectrum analysis can be performed by a Fast Fourier Transform (FFT) processor [18]. These FFT engines capture a large amount of data points to perform analysis on a broad frequency spectrum to calculate the phase and magnitude for many distinct frequencies simultaneously using a derivative of the Fourier Transform algorithm [18]. These FFT processors require a large amount of area overhead and power consumption to implement [5]. This is not desirable for a BIST approach where the goal is to use minimal overhead to test.

As an alternative solution, this BIST model applies a different approach to measure the phase and magnitude response of the DUT. Once the analog waveform generated by the TPG has stimulated the DUT, the DUT's response is collected by the ORA. To collect the spectral content, the ORA employs two pairs of MACs to capture

and accumulate the results. The MAC acts to filters out a single frequency of interest from the measured response. In this BIST approach, only one frequency can be analyzed at a time, rather than a broad spectrum approach performed by the FFT processor [5]. This allows a circuit with lower area overhead and faster calculation time to be developed. This process of measuring one exclusive frequency at a time is referred to as Selective Spectrum Analysis (SSA).

This SSA approach is not without limitations. To implement this circuit still requires resource overhead to generate tones and collect the DUT response; though the required area is a minimal when compared to that implemented by traditional FFT processors [5]. In addition, the test time to obtain accurate measurements is a function of the frequency words used and how many data points are collected. Therefore, a poor choice in frequency words or performing a test that involves the capturing of multiple frequency points can potentially result in a long test time. This proposed BIST approach includes a mathematical basis for calculating the phase and magnitude from the output of the ORA, but does not include a practical implementation for this design [4][5].

1.4 Thesis Statement

The goal of this thesis is to present additions and improvement to the design for the SSA Built-In Self-Test model for mixed-signal systems presented in [4]-[12]. This thesis will present a method for performing common analog tests on-chip. In addition, this thesis will also present improvements to the MAC-based ORA by introducing an improved multiplier design. This thesis will also present a technique for automatically generating the complete hardware descriptive model for multiple platforms of implementation.

Chapter 2 will present a detailed overview of the Built-In Self-Test model for mixed-signal systems. Background mathematics and procedures of measurement will also be elaborated upon. Chapter 3 will present the work completed to obtain an automated implementation of the Built-In Self-Test mixed-signal design. This includes improvements in the TPG using minimized NCOs and the ORA using an improved multiplier. Chapter 4 will discuss the improvements to the overall approach with the addition of on-chip interpretation of the ORA measurements. The thesis will be summarized in Chapter 5 and provide suggestions for future areas of research and development.

Chapter Two

Theory of Operation and Background

Complex mixed-signal electronic systems are being extensively developed. Consequently, the increasing cost associated with testing these complex systems has motivated research efforts to explore more efficient testing methodologies. This chapter will present an introduction to Built-In Self-Test using Selective Spectrum Analysis to evaluate the performance of analog devices in mixed-signal systems. This will cover the mechanisms of measurement which are used to collect test information. In addition, a description of previous analog testing techniques will also be presented. This chapter will also cover a brief overview of multiplier architectures and the Coordinate Rotation Digital Computing algorithm as an introduction to improvements to this existing Built-In Self-Test approach. Finally, this chapter concludes with a re-statement of the thesis goals.

2.1 Analog Characteristics

In the measurement of electronic circuits, there are key performance parameters that can be used to determine if the circuit has met design specifications. Critical parameters that define an analog circuit include its response to direct stimulus, distortion from the input waveform, and noise within the system. To measure each of these characteristics, the presented mixed-signal BIST model is capable of measuring linearity, frequency response, and a variety of noise measurements for an analog circuit [7].

Linearity is a very important characteristic of electronic amplifiers and filters. Linearity refers to the characteristic behavior of a device to have an output which is linearly proportional to the input [20]. This is considered synonymous with having low distortion and preserving the fidelity of analog signals. When two tones are applied to the input of an analog circuit which exhibits a poor linearity response, the output of the circuit experiences Intermodulation Distortion (IMD). This results in the formation of frequencies which are not integer multiple harmonic frequencies of the originally applied two tones [21]. IMD represents a serious hindrance to circuit performance as these unintentionally generated tones may result in signals which prove extremely difficult to filter out by normal means, causing an indeterminate response from the circuit. Illustrated in Figure 2-1 is an example of IMD. Two tones at frequencies 7 and 8 are supplied to an analog amplifier. If this amplifier was linear, the output would contain spectral content at only the two input tones and harmonics. Because this amplifier is nonlinear, the output contains the original frequencies, harmonics of the input tones, and intermodulation between the original frequencies and harmonics tones [6].

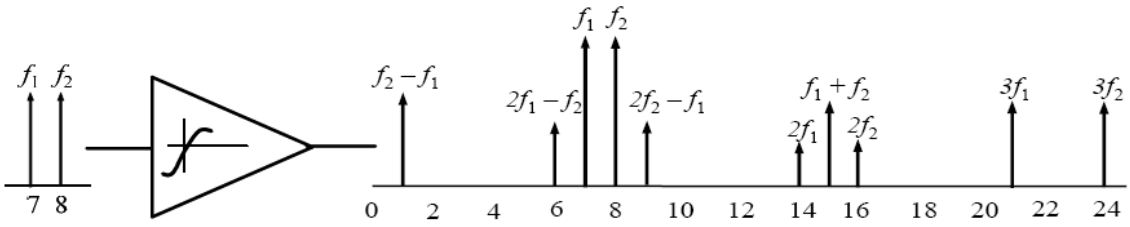

Figure 2-1: Illustration of Analog Distortion from Two Tone Input [6]

Frequency response is another important measurement when attempting to characterize an analog system [7]. Frequency response can be defined as the comparison between a system's output response and an input signal. Frequency response can be separated into two separate types of measurements: magnitude and phase response.

Magnitude response is the difference between the magnitudes of the incident waveform into the device and the magnitude of the output waveform. This is usually expressed graphically against the frequency of that waveform. This is illustrated in Figure 2-2, where the magnitude is recorded as the frequency is swept from low to high. Magnitude response is useful when characterizing the gain delivered by an amplifier or the cutoff point and slope of a filter [8]. Phase response serves an equally important role in characterizing the analog circuit. Phase response measures the phase delay of the output waveform which occurs due to the analog circuit.

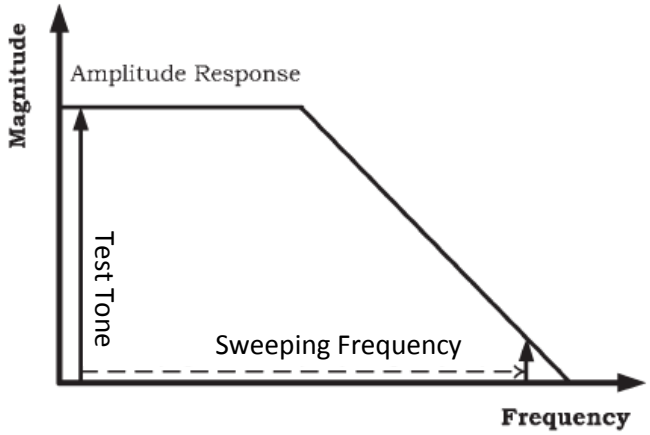

Figure 2-2: Illustration of Frequency Response Plot Showing the Magnitude Response [8]

In any electronic circuit, noise plays a considerable factor in determining the quality of the circuit. There are many sources which may generate noise in analog circuits including thermal noise, shot noise, flicker noise, and avalanche noise [20]. This noise appears in tandem with the signal of interest, thus excessive noise increases the difficultly of accurately interpreting the signal. This makes preserving the signal clarity a priority in analog design and testing. To characterize the noise within a system there are two important parameters. The first is the signal to noise ratio (SNR). This measurement is a

direct comparison between the power within the signal of interest and the noise generated by the system [9]. This is defined as

$$SNR = \frac{Signal\ \ Power}{Average\ \ Noise\ Power\ \ (within\ specified\ \ Bandwidth)} \quad (2\text{-}1)$$

The noise power within an SNR measurement must be defined to a particular bandwidth of interest to contain statistically significant information. SNR can be further subdivided into three equally important measurements based on how the noise power bandwidth is defined. The first subdivision is known as Total Harmonic Distortion (THD) [20]. In a THD measurement, the noise in question is only measured at frequencies where harmonic tones of the original signal exist. This gives a direct comparison between the signal of interest and the generated harmonics of that signal. The next subdivision of noise bandwidth measurement involves measuring only background noise while avoiding all harmonic tones in the measurement. This is how the traditional SNR measurement is performed. The final form of SNR subdivision is known as the Signal to Noise and Distortion (SINAD). The SINAD measurement is a combination of the previous two measurements where both background noise and harmonic distortion are combined as noise and compared to the signal of interest [20]. All three of these noise measurement techniques are illustrated in Figure 2-3. In this illustration each bullet represents a point of measurement.

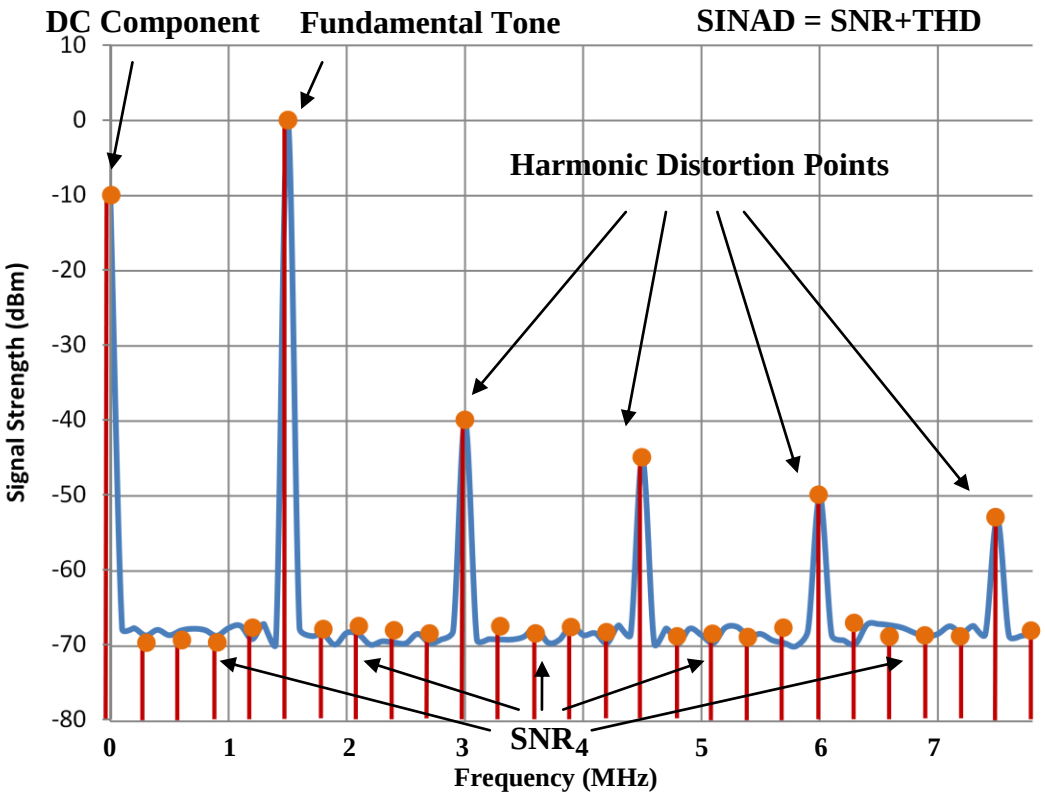

Figure 2-3: Illustration of SNR, THD, and SINAD Measurement Points in Frequency Sweep

The second critical parameter when measuring noise is called the Noise Figure (NF). This parameter is defined as the ratio between the SNR before the influence of the DUT and the SNR after the DUT has processed the signal [9]. This characteristic is important as it represents the degradation of the SNR due to interference within the DUT [20]. The formal definition for NF is shown in Equation 2-2.

$$NF = \frac{SNR_{IN}}{SNR_{OUT}} \tag{2-2}$$

2.2 Built-In Self-Test Using Selective Spectrum Analysis

Analog performance testing is a challenging task even when performed by an experienced engineer with the correct equipment. After the test data is collected, the test results of an analog circuit are then compared to a given set of system specifications to

verify performance [3]. In order to perform a myriad of analog tests such as linearity, frequency response, and noise measurements within a mixed-signal IC, it becomes necessary to implement a BIST approach. This allows for specification testing without the use of expensive and bulky external equipment.

To obtain analog measurements within this application, the BIST needs to collect the frequency spectrum content coming from the DUT. To collect this data, the BIST implements an ORA. Many techniques have been proposed to implement an ORA for analog testing in [21]-[25], but these designs, unfortunately, are incapable of measuring each of the desired analog characteristics of linearity, frequency response, and noise measurements within one unified model. Some designers utilize FFT to perform broad spectrum signal analysis, but the overhead requirements of these techniques limit their utilization in commercial applications [18].

A solution to this overall BIST approach was originally proposed in [6]. Within this design is a digital TPG that is constructed utilizing DDS to generate the tones required for stimulating the DUT and internal reference. The response of the DUT is collected by a digital ORA. To capture these measurements, this ORA will implement two pairs of multipliers/accumulators [4].

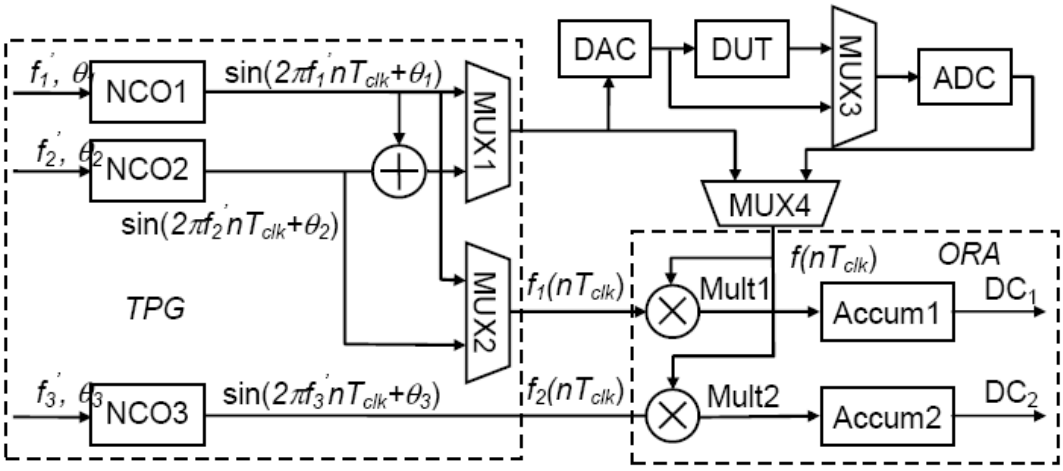

Figure 2-4: Illustration on Mixed-Signal BIST Architecture [11]

This mixed-signal BIST architecture, illustrated in Figure 2-4, is capable of performing measurements on the DUT depicted in the illustration. In addition, this approach utilizes an analog multiplexer, depicted as MUX3, which allows the DUT to be bypassed and the performance of the remaining system elements to be measured. The BIST method for analog measurement relies primarily on two elements to test an analog DUT. These are the DDS-based TPG and the MAC-based ORA [4].

2.2.1 Test Pattern Generator

As shown in Figure 2-4, the DDS-based TPG of is made of three numerically controlled oscillators (NCOs). In this design, each NCO consists of a phase accumulator, phase truncation stage, and phase to amplitude converter. The phase accumulator is capable of accepting both a frequency control word and a starting phase word. In the phase accumulator, the frequency control word, *f*, is continuously accumulated. This acts to control the frequency of the generated waveform. The phase word, θ, is used to define a starting offset for the accumulator, which will correspond to a phase offset for the

generated waveform. The phase truncation stage removes unused Least Significant Bits (LSBs) and formats the data so that it will properly interact with the PAC. This allows for a large accumulator to accumulate precise phase information but implement the conversion between phase and amplitude with a less precise PAC to reduce area overhead [4]. The PAC design is usually achieved using a Look-Up Table (LUT). The output of the LUT represents the magnitude component of a sinusoidal waveform and can be converted to an analog signal with a DAC or used as a digital reference tone [17]. The resulting waveform and frequency from each NCO are expressed in Equations 2-3 through 2-5.

$$f' = \frac{f \cdot f_{CLK}}{2^N} \tag{2-3}$$

$$\theta' = \frac{\theta * 2\pi}{2^N} \tag{2-4}$$

$$NCO = \sin(2\pi \cdot f' \cdot n \cdot T_{CLK} + \theta') \tag{2-5}$$

In these equations, the frequency control word and phase control word are represented by f and θ, respectively. The size of the phase accumulators in which these control words are accumulated, can be represented as *N*-bits wide. Finally the rate of accumulation can be represented as the system clock frequency, $f_{CLK} = 1/T_{CLK}$. The accumulator effectively acts to divide f_{CLK} by $2^N/f$, creating the observed frequency and phase of f' and θ' [4]. The final waveform output is expressed in Equation 2-5 where variable *n* is equivalent to the number of clock cycles which have elapsed. This NCO design within the TPG of this BIST is illustrated in Figure 2-5.

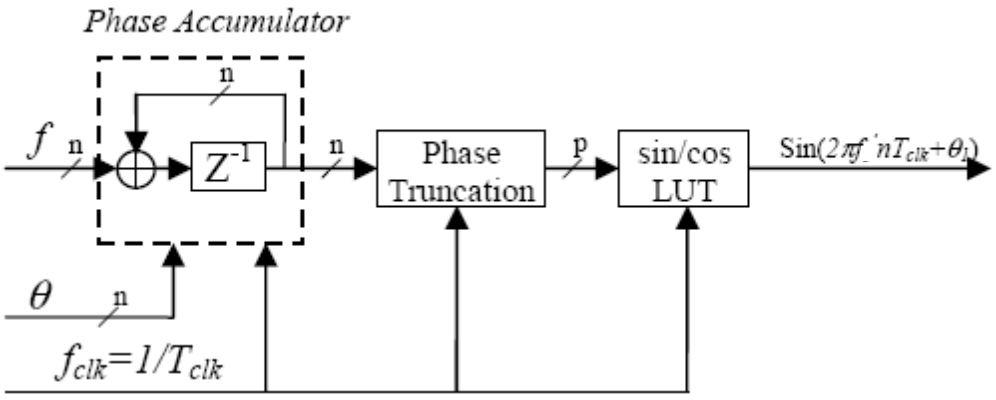

Figure 2-5: NCO Used in Mixed-Signal BIST Design [4]

In the BIST architecture in Figure 2-4, NCO1 and NCO2 are used to generate the test pattern waveform [4]. Through the multiplexer labeled MUX1, the model is capable of selecting a waveform generated by NCO1 or the combination of NCO1 and NCO2. This allows either a one-tone or two-tone waveform to be generated. The output of this multiplexer is connected to the DAC. By combining the NCOs with the DAC, the DDS model is completed, and it becomes possible to generate one or two-tone analog waveforms. The output of the multiplexer MUX1, which contains the test pattern, is also attached to the input of the ORA by multiplexor MUX4 [4]. This allows the BIST to bypass all analog components and test the functionality of the digital components. To capture each measurement, the ORA requires reference tones. These can be generated from additional NCOs in the TPG [11].

This TPG approach has two disadvantages due to using DDS to generate waveforms. The first problem with using DDS is the generation of image tones around the clock frequency. The lower half of the frequency band, 0Hz to f_{CLK}/2, contains the tones producible by the DDS, while the upper half of the frequency band, f_{CLK}/2 to f_{CLK}, contains images of these tones centered around f_{CLK} [17]. This divides the usable

spectrum in which the DDS can generate tones to half the system clock frequency to avoid these image tones. The second disadvantage occurs as the DDS generates tones close to the Nyquist bandwidth of half the clock frequency, $f_{CLK}/2$. At this point, the tone generated, f, and the tone's image around the clock, $f_{CLK} - f$, will appear very close to each other. This case also results in stronger image tones as quantization effects become more pronounced when the phase accumulator uses larger increments. This results in difficulties distinguishing the tone from the image. To reduce these images, sharp filters are typically applied to remove frequency components at and above $f_{CLK}/2$. This reduces the strength of images and quantization noise, but also limits the usable range of the DDS to less than 40 percent of f_{CLK} [17].

2.2.2 Output Response Analyzer

The second essential element of the mixed-signal BIST approach is the MAC-based ORA. Its function is to collect the response signal of the DUT. The ORA operates on the principle of SSA, where the objective is to measure the magnitude and phase of one specific tone.

The principles of operation for this ORA are straightforward. The ORA consists of two pairs of MACs. Each multiplier has one input attached to the multiplexer MUX4, which selects the source of the test data as shown in Figure 2-4. When undergoing a test, the waveform being sampled by the ADC can be represented as $f(nT_{CLK})$ [11]. The second input on each multiplier, signals f_1 and f_2, connect to a digital NCO which is generating a reference waveform. If signals $f_1(nT_{CLK})$ and $f_2(nT_{CLK})$ are set to

$\cos(\omega n T_{CLK})$ and $\sin(\omega n T_{CLK})$ respectively, then the resulting values of DC_1 and DC_2, which represent the results of each MAC, are as shown in Equations 2-6 and 2-7 [4].

$$DC_1(\omega) = \sum_n f(nT_{CLK}) \cdot cos(\omega n T_{CLK}) \tag{2-6}$$

$$DC_2(\omega) = \sum_n f(nT_{CLK}) \cdot sin(\omega n T_{CLK}) \tag{2-7}$$

Equations 2-6 and 2-7 show that DC_1 and DC_2 each represent the in-phase and out-of-phase components of $f(nT_{CLK})$ at frequency ω. By substituting Equations 2-6 and 2-7 for the in-phase and out-of-phase components of the Fourier transformation equation $F(\omega)$, the resulting identity can be obtained [4].

$$F(\omega) = \sum_n f(nT_{CLK}) e^{-j\omega n T_{CLK}} = DC_1(\omega) - j \cdot DC_2(\omega) \tag{2-8}$$

Equation 2-8 shows that the frequency content of the sampled waveform can be expressed by the in-phase and out-of-phase components DC_1 and DC_2, respectively. To selectively examine a single frequency component of the response waveform, the reference tones being mixed with the sampled waveform need to be modified to the desired test frequency in this MAC-based ORA design. To obtain the magnitude and phase of each frequency component, it becomes convenient to express the frequency measurements as shown in Equation 2-9 [4].

$$F(\omega) = DC_1(\omega) - j \cdot DC_2(\omega) = A(\omega) e^{+j\,\Delta\varphi(\omega)} \tag{2-9}$$

In this representation of $F(\omega)$, the component $A(\omega)$ represents the magnitude of the measurement for frequency ω. The second component $\Delta\varphi(\omega)$ represents the relative

phase delay of the signal of frequency ω with respect to the phase of the reference tones. These two parameters, $A(\omega)$ and $\Delta\varphi(\omega)$, are used to determine the results of each of the analog test measurements. The measurement of the relative phase shift can be represented as shown in Equation 2-10 [4].

$$\Delta\varphi(\omega) = tan^{-1}\left(\frac{DC_2}{DC_1}\right) \tag{2-10}$$

This represents phase measurement only within the limited output of the arctangent range of -90º to +90º. To express the absolute phase shift ($\Delta\varphi_0(\omega)$), the magnitude and sign convention of DC_1 and DC_2 must be factored into the measurement. This measurement of absolute phase is summarized in Table 2-1.

Table 2-1: Relationship Between $\Delta\varphi(\omega)$ and $\Delta\varphi_0(\omega)$ [4]

	|DC1|>|DC2|	|DC1|<|DC2|
DC1>0, DC2>0	$\Delta\varphi(\omega) = \Delta\varphi_0(\omega)$	$\Delta\varphi(\omega) = 90º - \Delta\varphi_0(\omega)$
DC1>0, DC2<0	$\Delta\varphi(\omega) = 360º - \Delta\varphi_0(\omega)$	$\Delta\varphi(\omega) = 270º + \Delta\varphi_0(\omega)$
DC1<0, DC2>0	$\Delta\varphi(\omega) = 180º - \Delta\varphi_0(\omega)$	$\Delta\varphi(\omega) = 90º + \Delta\varphi_0(\omega)$
DC1<0, DC2<0	$\Delta\varphi(\omega) = 180º + \Delta\varphi_0(\omega)$	$\Delta\varphi(\omega) = 270º - \Delta\varphi_0(\omega)$

To solve for the $A(\omega)$ parameter, three approaches are suggested in [4]. The first approach demonstrates that it is possible to adjust the phase of the testing waveform used to stimulate the DUT, such that the DUT response becomes in-phase with the reference tone. If the device is stimulated and the phase shift is measured, the inverse of this phase shift can be applied to the reference source in the TPG. This will shift the in-phase measurement to match the DUT phase response. Therefore the value within the in-phase MAC will directly represent the magnitude of the measurement, while the out-of-phase MAC will contain a value of zero. This magnitude is expressed in Equation 2-11 [4].

$$\begin{aligned} A(\omega) &= F(\omega)e^{-j\Delta\varphi(\omega)} \\ &= \sum_n f(nT_{CLK})e^{j[\omega n T_{CLK} - \Delta\varphi(\omega)]} \\ &= \sum_n f(nT_{CLK})\cos(\omega n T_{CLK} - \Delta\varphi(\omega)) \end{aligned} \quad (2\text{-}11)$$

This is very beneficial as there is no calculation circuitry need to directly obtain $A(\omega)$ from DC_1 and DC_2. However circuitry would be need to obtain the phase delay and rerun the test with the modified phase to obtain a measurement. This approach would not be valid if measuring noise since the noise phase could change for each subsequent measurement [4].

The second approach to obtaining magnitude relies on factoring the phase out from the observed measurement. If the phase shift of the observed waveform can be obtained, then it can simply be divided out of either the in-phase or out-of-phase component of measurement as shown in Equation 2-12 [4].

$$A(\omega) = \frac{DC_1}{\cos(\Delta\varphi(\omega))} = \frac{DC_2}{\sin(\Delta\varphi(\omega))} \quad (2\text{-}12)$$

Unlike the first approach, this approach does not require the test to be rerun to calculate the magnitude; therefore it can be used to perform noise measurements. However, there is additional hardware required to calculate phase, convert the phase shift through a cosine or sine LUT, and perform the division [4].

There is one common factor between the first two approaches. Both approaches utilize phase shift to calculate magnitude. The third approach does not utilize phase shift to calculate the magnitude, but instead relies on the values of DC_1 and DC_2 directly. The third option presents the most direct method to implement as it only depends on the raw values of DC_1 and DC_2. This is described in Equation 2-13 [4].

$$A(\omega) = \sqrt{{DC_1}^2 + {DC_2}^2} \quad (2\text{-}13)$$

Each of these three options represents a valid method of calculating the magnitude for the SSA response in this MAC-based ORA. Now that the ORA is capable of collecting phase and magnitude utilizing two pairs of MACs, it requires a method for controlling the length of accumulation, *N*. When the ORA is attempting to accumulate the response of a single tone from the DUT, the accumulator outputs DC_1 and DC_2 are expressed as shown in Equations 2-14 and 2-15 [4].

$$\begin{aligned} DC_1(N) &= \sum_{n=1}^{N} A\cos(2\pi f n T_{CLK} + \Delta\varphi) \cdot \cos(2\pi f n T_{CLK}) \\ &= \frac{AN}{2}\cos(\Delta\varphi) + \frac{A}{2}\sum_{n=1}^{N} cos(4\pi f n T_{CLK} + \Delta\varphi) \end{aligned} \tag{2-14}$$

$$\begin{aligned} DC_2(N) &= \sum_{n=1}^{N} A\cos(2\pi f n T_{CLK} + \Delta\varphi) \cdot \sin(2\pi f n T_{CLK}) \\ &= \frac{AN}{2}\sin(\Delta\varphi) + \frac{A}{2}\sum_{n=1}^{N} sin(4\pi f n T_{CLK} + \Delta\varphi) \end{aligned} \tag{2-15}$$

DC_1 and DC_2 contain the in-phase and out-of-phase components necessary for calculation, represented as $A \cdot \cos(\Delta\varphi)$ and $A \cdot \sin(\Delta\varphi)$, respectively. An additional component of $\frac{N}{2}$ is factored into the result as a component of accumulation. Because this component exists equally in both DC_1 and DC_2 it can be factored away if the number of accumulation cycles is known. Unfortunately, these terms also contain a summation of cosine and sine which act to skew the correct answer based on when the accumulation is halted. The resulting DC_1 and DC_2 terms will accumulate over time as shown in Figure 2-6 [4].

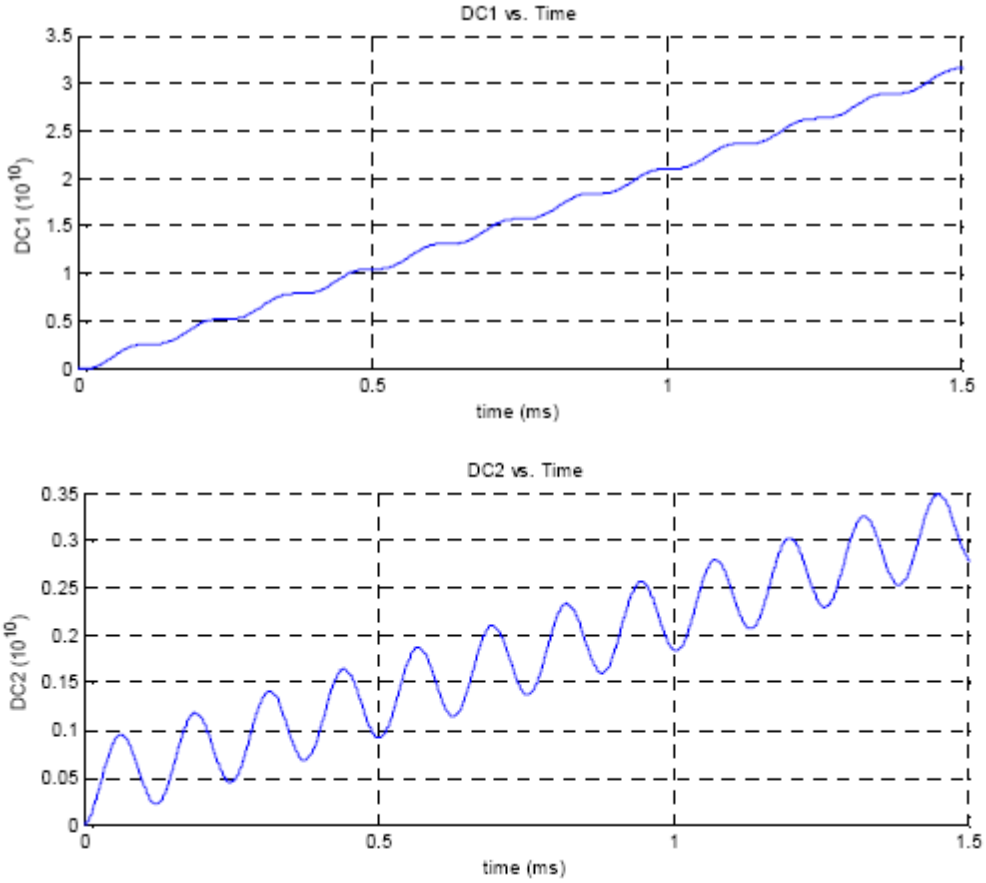

Figure 2-6: DC_1 and DC2 Versus Time [11]

The slope of each line represents the DC_1 in-phase accumulated value of $\frac{AN}{2}\cos(\Delta\varphi)$ and the DC_2 out-of-phase accumulated value of $\frac{AN}{2}\sin(\Delta\varphi)$. The oscillation in each waveform is a result of the additional erroneous sinusoidal component. In order to obtain the correct measurement, this term must be negated from the calculations. One method for removing this term is to utilize the identity that the summation of all points in the unit circle will equate to zero as shown in Equation 2-16. Therefore, there must be a number of discrete summations *N* of this sinusoidal error factor which result in canceling out this term from both DC_1 and DC_2.

$$0 = \int_0^{2\pi} cosine(x)\delta x = \int_0^{2\pi} sine(x)\delta x \tag{2-16}$$

The number of samples *N* which relates to the canceling of this term occurs at the Integer Multiple Period (IMP) of the waveform. This IMP occurs when the waveform crosses the axis after each period. The waveforms are generated using NCOs in the TPG which utilize a phase accumulator. When these accumulators reach their limit and overflow the waveform crosses the origin. The point where the waveform crosses the origin corresponds with an IMP [630]. However, if an overflow occurs and the accumulator contains any value other than zero, then the discrete measured value will contain a measurement past the origin instead of at the origin. The overflow flag of the accumulator as well as the value of zero within the accumulator can be used to signal that an IMP has occurred; a simple circuit can be constructed as shown in Figure 2-7.

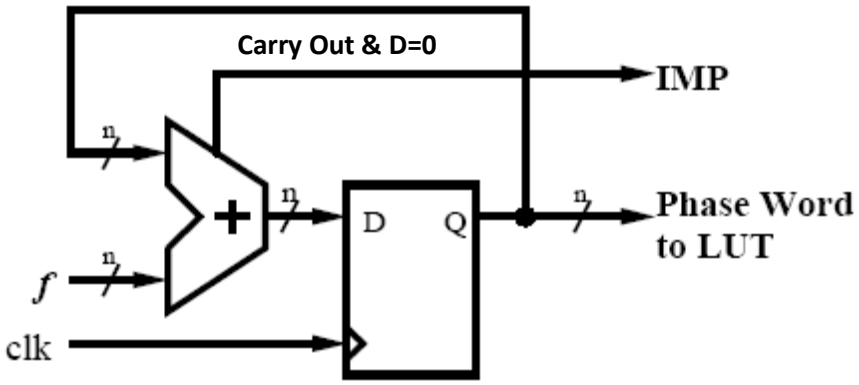

Figure 2-7: IMP Detection Circuit for Single Tone Measurement [630]

For tests that require two tones, a slightly modified approach must be taken to obtain the IMP. In a two tone measurement, the results stored in DC_1 and DC_2 contain error based on both the fundamental waveforms and the intermodulation of these waveforms. Therefore the true IMP which will cancel out these terms will occur only when the IMPs of both the fundamental and intermodulation tone occur simultaneously [30].

2.2.3 Analog Functional Measurements Approach

This BIST architecture, utilizing a DDS-based TPG and MAC-based ORA, is capable of performing linearity, frequency response, and noise measurements. These tests allow the performance of the analog circuit to be characterized and compared to design specifications [6]. An overview of how these measurements are conducted within the BIST circuitry is given within this section.

The first of these testable measurements is linearity. The linearity response of a real system is defined by severity of non-linearity, as all real devices inherently contain some degree of non-linearity. The test for linearity involves measuring the Third Order Intercept Point (IP3) under a two tone test input [6]. This product is generated from higher order harmonics of the input tones intermodulating with the original tones to create new frequency content. This is illustrated in the Figure 2-1. Specific points of interest are the frequencies $2f_2$-f_1 and $2f_1$-f_2. These represent the two artificially generated products of the IMD which appear closest to the original tones. These two tones in particular are known as the Third Order Intermodulation (IM3) components [4]. The IM3 tones are chosen to define linearity because their proximity to the fundamental tones makes them extremely difficult to filter away. Using these IM3 terms, the Input Referred Intercept Point (IIP3) is defined by Equation 2-17 [4].

$$IIP_3[dBm] = \frac{\Delta P[dB]}{2} + P_{in}[dBm] \tag{2-17}$$

The ΔP term represents the difference in power between these IM3 tones and the fundamental tones, and P_{in} is the signal power at the input of the device. The intermodulation components of this device can be theoretically calculated using the

Taylor Series approximations. Given that the linearity of interest is that of a non-dynamic input, Taylor Series approximation becomes an ideal representation for nonlinear time-invariant systems. Using the Taylor Series method, the input to analog system, *x(t)*, and the transfer function of the system, *y(t)*, are represented as shown in the following two Equations 2-18 and 2-19 [6].

$$x(t) = A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t) \tag{2-18}$$

$$y(t) = \alpha_0 + \alpha_1 x(t) + \alpha_2 x(t)^2 + \alpha_3 x(t)^3 + \cdots \tag{2-19}$$

In this case, A_1 and A_2 represent the magnitude of the tones while ω_1 and ω_2 represent the two separate frequencies. The symbols $\alpha_{\#}$ represent the time-invariant coefficients of the Taylor Series expansion [4].

$$\begin{aligned} y(t) = \quad & \tfrac{1}{2}\alpha_2\left(A_1{}^2 + A_2{}^2\right) \\ & + \left[\alpha_1 A_1 + \tfrac{3}{4}\alpha_3 A_1\left(A_1{}^2 + 2A_2{}^2\right)\right]\cos(\omega_1 t) \\ & + \left[\alpha_1 A_2 + \tfrac{3}{4}\alpha_3 A_2\left(2A_1{}^2 + A_2{}^2\right)\right]\cos(\omega_2 t) \\ & + \tfrac{1}{2}\alpha_2\left(A_1{}^2\cos(2\omega_1 t) + A_2{}^2\cos(2\omega_2 t)\right) \\ & + \alpha_2 A_1 A_2[\cos([\omega_1 + \omega_2]t) + \cos([\omega_1 - \omega_2]t)] \\ & + \tfrac{1}{4}\alpha_3\left[A_1{}^3\cos(3\omega_1 t) + A_2{}^3\cos(3\omega_2 t)\right] \\ & + \tfrac{3}{4}\alpha_3 \begin{Bmatrix} A_1{}^2 A_2[\cos([2\omega_1 + \omega_2]t) + \cos([2\omega_1 - \omega_2]t)] \\ + A_1 A_2{}^2[\cos([2\omega_2 + \omega_1]t) + \cos([2\omega_2 - \omega_1]t)] \end{Bmatrix} \end{aligned} \tag{2-20}$$

After solving this expression for the component of the fundament term at ω_2 and the IM3 term at $2\omega_2 - \omega_1$ and assuming that A_1=A_2, then the IIP3 and Output Referred Intercept Point (OIP3) can be obtained from the Taylor Series variables as shown in Equations 2-21 and 2-22 [6].

$$\text{IIP}_3 \approx \sqrt{\tfrac{4}{3}\left|\tfrac{\alpha_1}{\alpha_3}\right|} \quad \text{, when} \quad \alpha_1 \gg \tfrac{9}{4}\alpha_3 A^2 \tag{2-21}$$

$$\text{OIP}_3 \approx \alpha_1 IIP_3 \tag{2-22}$$

This technique of measuring linearity relies on the assumption that the test tones and output of the device are relatively small in magnitude such that the device does not become damaged or desensitized [6]. The illustration below in Figure 2-8 shows the relationship between the fundamental tone, IM3 point, and ΔP.

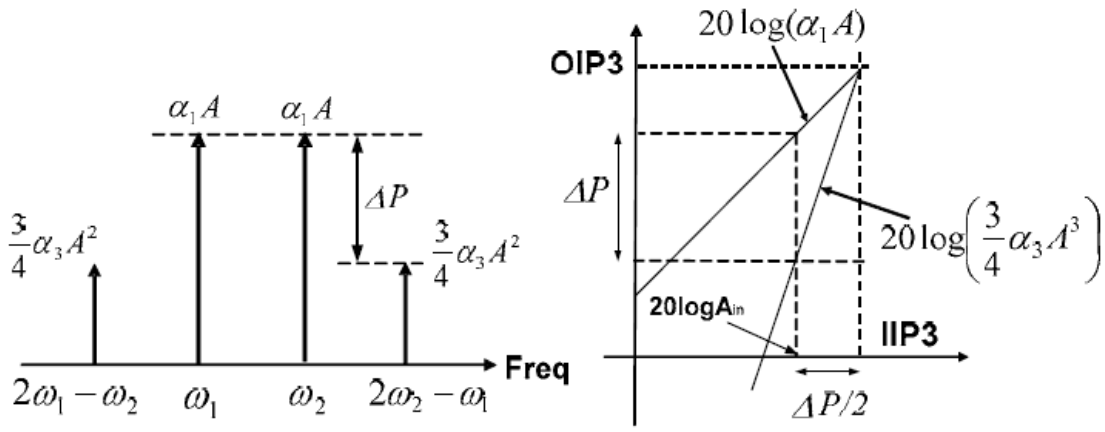

Figure 2-8: Illustration of Fundamental Tone Compared to IM3 Tone in IP3 Test [6]

The frequency response is obtained by forcing the TPG to generate a single tone and pass that tone through the DUT. The ORA will then measure the magnitude and phase of the resulting waveform at the frequency of the generated tone. However, this measurement includes the response of not only the DUT, but every component in the measurement path. To negate the effects of the additional components, the test tone is measured a second time. In the second measurement, the DUT is bypassed using an analog multiplexer and the waveform is re-measured. This represents the resulting magnitude and phase from the interaction with all other components within the measurement path. This includes the DAC, smoothing filter, analog multiplexer, and ADC. The resulting difference between these two measurements will represent the frequency response of only the DUT [7]. This procedure can be repeated for different

frequencies by sweeping the generated tone and measured tone together. This is illustrated in Figure 2-2. This allows for a measurement of the response over a range of possible frequency values.

Each noise measurement test begins by forcing the TPG to generate a single tone and passing that waveform through the DUT. The ORA will then calculate a measurement of the signal strength at the fundamental tone [4]. At this point, the test varies based on which of the four noise measurements is being performed. To obtain THD, the ORA will measure the magnitude of harmonic tones which occur at integer frequencies of the fundamental tone, while the TPG continues to generate the fixed fundamental tone. Each of these integer multiple frequencies of the fundamental tone represents a harmonic [20]. The THD can be obtained by taking the ratio of the fundamental tone signal strength to the average of these harmonic noise measurements. To obtain SNR, the ORA will measure the magnitude of frequencies that are not integer multiples of the fundamental tone, while the TPG continues to generate a fixed fundamental tone. This will collect the spectral content of the noise which is not a function of fundamental tone [4]. By averaging these collected noise measurements we obtain the average background noise of the system. The ratio between the signal strength of the fundamental tone and this background noise measurement will be the result of the SNR [20]. The measurement of SINAD, involves the combination of the tests for SNR and THD. Rather than selectively choosing measurement points which contain only a certain type of noise, SINAD selects all measurement points to include in the noise calculation with the exception of the fundamental tone. The SINAD can be obtained by taking the ratio of the fundamental tone signal strength to the average of the noise

measurement points [20]. An illustration of SNR, THD and SINAD can be seen in Figure 2-3.

The NF measurement represents a very important measurement to the device. Measurements of SNR, THD, and SINAD represent the ratio of signal strength to noise strength of the entire measurement path which includes the DAC and ADC. The NF measurement represents the degradation in SNR due to the DUT alone [20]. The NF test represents the longest of the noise tests, because this test requires the SNR to be measured twice. The first SNR measurement is of the DUTs output response. This represents the SNR of the complete path including the DUT. The second SNR measurement involves bypassing the DUT in the measurement path [4]. This can be performed using the analog multiplexor, MUX3, as illustrated in Figure 2-4 to bypass the DUT. By taking the ratio of these two measurements as described in Equation 2-2, the NF can be obtained [24].

2.3 Multipliers

Multipliers make up the core of the ORA. A multiplier is a binary circuit which utilizes a combination of logic elements and adder units to implement a multiplication operation on two binary numbers called the multiplier and multiplicand. The basic construction of a multiplier has two parts. The first stage is the generation of the partial products. The second stage is the summation of these partial products to produce the final result [29]. Multipliers are typically differentiated by their implementation of the first stage.

There are many common types of multiplier designs. Array multipliers represent the simplest class of multipliers consisting of a simple array of logical *AND* gates to

generate the partial products and full adders to implement the addition of each partial product row [29]. The fundamental shortcoming of this multiplier is that it is intended for unsigned multiplication. A simple modification is required to transform this multiplier for signed mathematics. By negating the upper most significant bits (MSB) of each partial product and the highest order partial product, the array multiplier is transformed into a Baugh-Wooley multiplier capable of signed multiplication [29]. These two multipliers represent the simplest designs, but both multipliers suffer from long calculation time and high area overhead. These two multipliers have been superseded by an improved multiplier known as the Modified-Booth multiplier.

The Modified-Booth multiplier seeks to reduce the overall size of the multiplier by exploiting patterns in the multiplier and multiplicand inputs to generate half the number of partial products as the Baugh-Wooley multiplier. The Modified-Booth algorithm is based on the fact that a radix-4 multiplier will require half the number of partial product stages as a radix-2. The Booth algorithm examines the contents of the multiplier register input and seeks to reduce the number of partial product rows required by logically combining partial products before they are generated [28]. Within the partial products of a standard signed multiplier, each partial product row can have only one of two possible values. These two values are either the multiplicand or all zeros. The Booth algorithm encodes possible radix-4 combinations in an entity called the Booth recoder [28]. This coded instruction is then applied to these reduced partial products as shown in Table 2-2.

Table 2-2: Booth Recoding Bits Decoded

Multiplier bits			Booth Code			Instruction
X_{2N+1}	X_{2N}	X_{2N-1}	S	M2	M1	Value of Partial Product
0	0	0	0	0	0	Multiplicand x 0
0	0	1	0	0	1	Multiplicand x 1
0	1	0	0	0	1	Multiplicand x 1
0	1	1	0	1	0	Multiplicand x 2
1	0	0	1	1	0	Multiplicand x -2
1	0	1	1	0	1	Multiplicand x -1
1	1	0	1	0	1	Multiplicand x -1
1	1	1	1	1	1	Multiplicand x -0

The Booth recoder then takes these coded bits, and uses a binary shift and invert operation to reconstruct the partial products. After the Booth algorithm is complete, there are only half the original partial products [28].

The Modified-Booth multiplier modifies the number of partial product bits to improve the performance of the multiplier, but this is not the only method to improve performance. The next technique is known as Wallace tree reduction. The Wallace tree reduction method relies on cleaning up the inefficiency when adding partial product bits to improve calculation speed [28]. In a serial arrangement of accumulation of the partial products, each partial product is serially added to the next partial product after it. This results in *N* partial products consuming *N-1* addition stages. Wallace tree reduction attempts to reduce the long computation time of accumulating these partial products by adding multiple partial product factors together simultaneously rather than sequentially. The Wallace tree reduction method is designed to reduce the number of partial products to two final rows. These two remaining partial products can then be summed together using any number of fast adder approaches [28]. Illustrated in Figure 2-9 is an example of how rearranging partial product addition stages can reduce the calculation time. Figure

2-9 (a) shows a serial summation of products resulting in a worst case path of three adders, but Figure 2-9 (b) rearranges these adders to result in a worst case path of two adders.

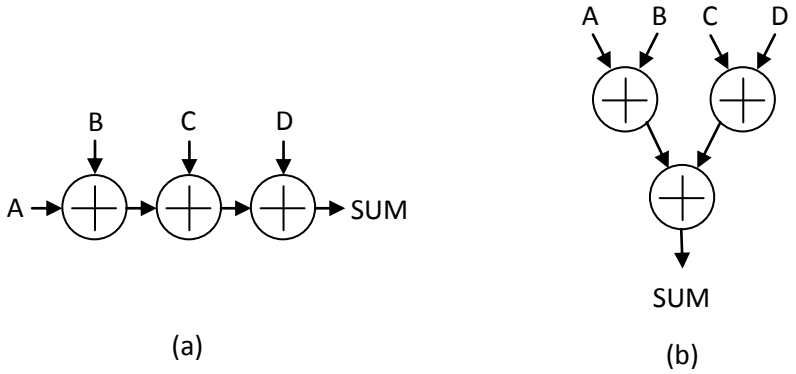

Figure 2-9: Illustration of Rearranging Partial Product to Decrease Worst Case Path

2.4 Coordinate Rotation Digital Computing Algorithm

The Coordinate Rotation Digital Computer (CORDIC) algorithm is a well known iterative approximation technique to perform various arithmetic operations**.** The algorithm is very simple to implement in hardware, because it uses only shift-and-add or shift-and-subtract operations to perform vector rotation in a two dimensional plane [29]. By using successive approximation the CORDIC algorithm can approximate the most basic mathematical functions. The most common use of the CORDIC algorithm is in approximating trigonometric functions [28]. The CORDIC algorithm can be broken into three distinct groups. The distinguishing factor of each group is how the CORDIC successive approximation function is implemented [29].

- Linear CORDIC – This CORDIC design is capable of expressing linear functions such as multiplication, division, addition, and subtraction. In this method the correction factor that is applied to each successive approximation remains constant. This allows it to be able to express linear operations [28].

- Circular CORDIC – This CORDIC design is capable of expressing trigonometric functions which can be expressed around the unit circle. Functions such as sine, cosine, tangent, arctangent, arcsine, arccosine, and square root can be expressed. In this method, the successive approximation is a factor of the arctangent function. This allows for circular functions to be expressed [28].
- Hyperbolic CORDIC – This CORDIC design is used for representing hyperbolic, logarithmic, and exponential functions. The hyperbolic CORDIC utilizes an inverse hyperbolic tangent function when performing a successive approximation. This allows for the expression of exponential and logarithmic based operations [28].

The basic operational theory of the CORDIC algorithm is that the CORDIC will approximate a two dimensional vector. To obtain this vector, the CORDIC will start at 0 degrees rotation and attempt to reach a goal of either phase or magnitude. Before each rotation, the CORDIC will be able to measure if it overshot or has not yet reached its goal [29]. Depending on this, it will either add or subtract the next rotation information from the current approximation. Each successive approximation in the circular CORDIC algorithm rotates by:

$$Rotation\ added(N) = \tan^{-1}(\ 2^{-N}\ ) \tag{2-23}$$

In this estimation, the variable N represents N^{th} rotation in the series. As evident by this formula, as the number of rotations is increased, the step size of the higher order rotations represents a decreasing amount of change. The illustration in Figure 2-10 represents the CORDIC algorithm attempting to reach a phase of 65 degrees.

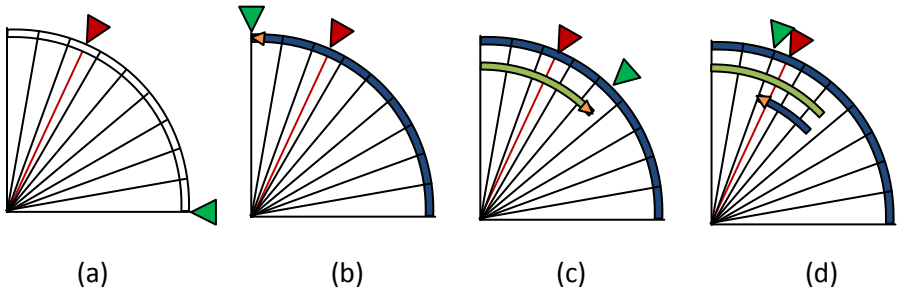

Figure 2-10: Illustration of CORDIC Rotational Procedure

In this example, the red arrow illustrates the fixed goal of 65 degrees, while the green represents the current CORDIC estimation. On the first attempt in Figure (a), the CORDIC begins at 0 degrees. The CORDIC is able to ascertain that it has not yet reached the goal of 65 degrees, so it adds the first rotational increment of 90 degrees shown in Figure (b). This estimate overshoots the goal of 65 degrees. Because the CORDIC has passed the goal, it must now take measures to change the direction of the next phase increment. This causes the CORDIC to subtract the next phase rotation of 45 degrees shown in Figure (c). This process is repeated in Figure (d) where it adds the third rotational phase giving it a current phase of 71 degrees. This process will continue to become more accurate as it successively approximates the measurement until a fixed number of rotations has been achieved by the CORDIC.

The CORDIC algorithm for linear, circular, and hyperbolic functions can all be combined into the unified CORDIC model [29]. In the unified CORDIC model, the approach is simplified into one vector *V* with three components *X*, *Y*, and *Z*. The vector components *X* and *Y* represent the projection of the vector *V* into the two dimensional xy-plane. The third component, *Z*, represents the angle of the vector with respect to the *X*

axis. For each successive approximation, the value of these components can be expressed as shown in Equation 2-24 [29].

$$V_{i+1} = \begin{cases} x_{i+1} = x_i - m\sigma_i 2^{-S(m,i)} y_i \\ y_{i+1} = y_i + \sigma_i 2^{-S(m,i)} x_i \\ z_{i+1} = z_i - \sigma_i \alpha_{m,i} \end{cases} \quad (2\text{-}24)$$

For each of the three CORDIC models, the CORDIC mode parameter *m*, rotational direction σ_i, shift sequence $S(m,i)$, and rotational angle $\alpha_{m,i}$ used in the definition of the unified CORDIC algorithm can be expressed as shown in Equations 2-25 through 2-28 [29].

$$m = \begin{cases} 0 & \text{For Linear CORDIC} \\ 1 & \text{For Circular CORDIC} \\ -1 & \text{For Hyperbolic CORDIC} \end{cases} \quad (2\text{-}25)$$

$$\sigma_i = \begin{cases} sign(Z_i) & \text{For Linear CORDIC} \\ sign(X_i + Y_i) & \text{For Circular CORDIC} \end{cases} \quad (2\text{-}26)$$

$$S(m,i) = \begin{cases} 0,1,2,3,4,5,\cdots & \text{m=1} \\ 1,2,3,4,5,6,\cdots & \text{m=0} \\ 1,2,3,4,4,5,\cdots & \text{m=-1} \quad \text{repeated at } \frac{3^{i+2}-1}{2} \end{cases} \quad (2\text{-}27)$$

$$\alpha_{m,i} = \frac{1}{\sqrt{m}} \arctan(\sqrt{m} \cdot 2^{-S(m,i)}) \quad (2\text{-}28)$$

Using these equations, the unified CORDIC model can be completely constructed for either the linear, circular, or hyperbolic model. The CORDIC algorithm however has three major disadvantages. The first disadvantage is the speed of calculation. The CORDIC relies on a method of successive approximation. After each pass through the CORDIC calculation, the results become more accurate [29]. Unfortunately, it requires

multiple iterations until the output of the CORDIC calculation begins to approximate the correct value within a reasonable margin of accuracy. The second disadvantage also occurs due to the successive approximation. This method will continuously approach the correct value, but never obtain the true value [29]. The third disadvantage in the CORDIC approach appears because of the successive shifting vector rotation. With each rotation, a successively decreasing amount of error is added to the result. This error is based solely on the number of rotations taken to achieve the result. This error can be expressed as shown in Equation 2-29, where *N* represents the number of rotations [28].

$$Error(N) = \prod_{n=0}^{N} \sqrt{(1 + m\sigma_i^2 2^{-2*S(m,i)})} \tag{2-29}$$

2.5 Thesis Restatement

This chapter has presented previous work on mixed-signal BIST using SSA [4]-[12]. This will become the foundation for the topic of this thesis, which includes the implementation of on-chip measurement techniques within the mixed-signal design and the automated generation of this BIST approach.

To achieve these ends, this thesis will focus on the design of the model required to produce these measurements. This thesis will present a new multiplier model which will reduce the size from conventional multipliers as well as increase the speed of calculation. Another goal of this thesis will be to introduce a technique to automatically generate a hardware description language for this mixed-signal BIST design. Finally, this thesis will present a method for calculating on-chip the results of the tests listed in this chapter utilizing a CORDIC based calculation unit.

Chapter Three

Mixed-Signal Built-In Self-Test Implementation

Implementing this mixed-signal Built-In Self-Test design requires the conceptual design to be capable of being expressed in hardware. This is usually achieved using a Hardware Description Language (HDL), which is a language that can be used to express hardware circuit elements for simulation and physical realization of the circuit. One of the most common forms of HDLs is known as Very High Speed Integrated Circuit Hardware Description Language (VHDL). Once this mixed-signal Built-In Self-Test design has been physically implemented from the VHDL, it can then be integrated with the intended analog components and the analog circuit to be tested.

This chapter will discuss the improvements to the implementation of the mixed-signal Built-In Self-Test for analog circuits. The topics covered include automatically generating the hardware descriptive model based on user-defined system performance parameters, producing sinusoidal waveforms in a digital core using a hardware description language which lacks trigonometric operators, and constructing an Output Response Analyzer using an improved multiplier design.

3.1 Program Generation

This mixed-signal BIST approach can be utilized in multiple applications; therefore there is a need to develop a method to generalize the model so that it can be implemented to fit multiple instances with differing requirements. In most HDLs, some form of parameterization is possible. This allows for HDL structures to be reconfigured based on the requirements of the application. However, some HDL structures prove too complex to be completely parameterized for all applications. Attempting to parameterize these complex models can result in a hardware description that is too difficult to verify proper operation due to the large number of conditional synthesis statements required to create a universalized model.

This thesis proposes a solution to the potential complexity of parameterizing this mixed-signal BIST approach by implementing a customized program designed to automatically generate the hardware description model given a set of user-defined system parameters. This program, *MSBISTGen.exe*, written in the C programming language is capable of generating the required VHDL models to implement the digital portion of mixed-signal BIST design; this includes the Test Pattern Generator, Output Response Analyzer, and test controller of the BIST design. Given user-defined requirements for the implementation of the design, this C program is also capable of generating the VHDL model to meet these design goals.

The mixed-signal BIST approach described in this thesis can be utilized in many applications, but the components which comprise the hardware model will remain relatively similar within each instance of this BIST model. The design will contain multiple NCOs for the TPG, multipliers and accumulator for the ORA, and multiplexers

to control the datapath as shown in Figure 2-4. However, based on the intended application, priority of the hardware implementation of this BIST design may focus on runtime of the tests, precision of measurements, or area resource consumption. Maximizing these design requirements can be achieved by altering key attributes such as the resolution of the DDS-based TPG or implementation of the Multiplier Accumulator pairs which comprise the ORA. However, by exploiting the fact that the majority of these components remain unchanged between implementations, we can use exact models for these unchanged elements in each design using this mixed-signal BIST. For those components to be altered based on specific system characteristics or other user parameters, the customized program can be used to insert, remove or modify portions of the hardware description to accommodate any changes.

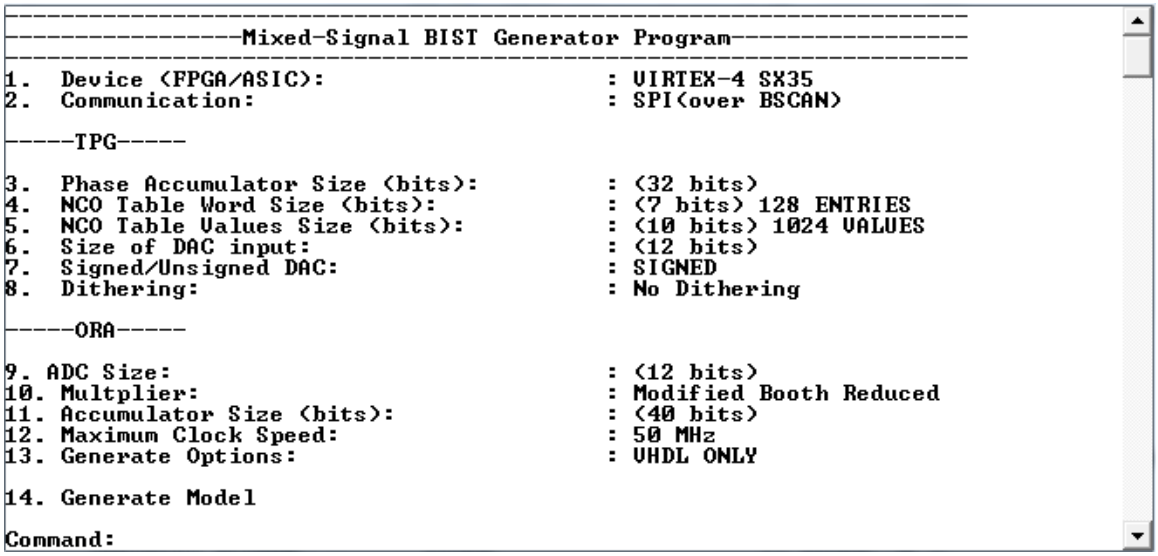

Figure 3-1: Screenshot from Automated Program to Generate Mixed-Signal BIST Model, *MSBISTGen.exe*

This program prompts the user to define the set of model parameters to be used to generate the VHDL model as shown in Figure 3-1. These options include key design parameters for the DDS-based TPG such as the size of the phase accumulator, and Look-Up Table used to generate the waveform. There are also additional options to format the output of the TPG to interface with varying sizes and formats of DACs. The ORA can also be customized to accommodate varying sizes of the ADCs. In addition, the type of multiplier and size of accumulator which compose the MAC can be selected.

The program takes each user input and interprets these choices into software flags which will correspond to hardware implementation decisions. After all decisions have been evaluated by the program, it will generate the hardware model in a VHDL file format and save the resulting model within its own project directory. To generate the VHDL files, the *createfile* function within the standard C library is invoked to generate each VHDL file. Then using the *fprintf* function, the VHDL formatted model is generated based on the user defined parameters and copied into each file. The structure of the generated model can be seen in Figure 3-2. Detailed information on the content of each generated file can be found in Table 3-1.

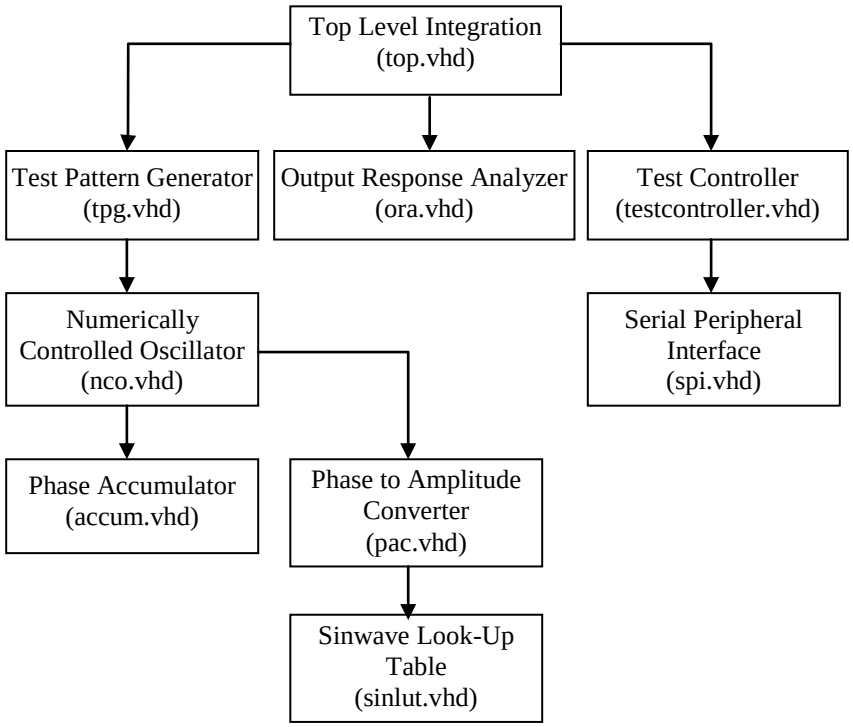

Figure 3-2: Hierarchical Tree of Files Generated for Mixed-Signal BIST Model

Table 3-1: Description of Generated Files

Filename	**Description**
top.vhd	This generated file contains the top level entity of the mixed-signal BIST model. The TPG, ORA, and Test Controller are interconnected on this level.
tpg.vhd	This generated file contains the DDS-based Test Pattern Generator model which instantiates each NCO.
ora.vhd	This generated file contains the SSA-based Output Response Analyzer model which instantiates both MACs.
testcontroller.vhd	This generated file contains the test controller entity used to send and retrieve results of the test.
nco.vhd	This generated file contains the definition for the NCO.
accum.vhd	This file contains the accumulator used to accumulator the phase information.
pac.vhd	This file contains the Phase to amplitude converter which uses the reduced sinwave Look-Up Table discussed in this chapter as well as any signal processing necessary to properly format the LUT output.
sinlut.vhd	This generated file contains the custom created Look-Up Table to implement the conversion between linear accumulation and sinusoidal output
spi.vhd	This file contains the serial peripheral used to serialize information to be passed in and out

3.1.1 Test Pattern Generator

At the heart of any Built-In Self-Test technology is a Test Pattern Generator. Unlike the traditional digital TPG design, a mixed-signal TPG requires a slightly different approach. The goal of the mixed-signal TPG is to create an analog waveform to stimulate the Device Under Test. As discussed in Chapter 2, this can be performed utilizing a DDS-based approach to generate either one or two tones. To achieve this we can simply utilize a linear accumulator in series with a Look-Up Table to convert these linear increments into a sinusoidal pattern [17]. When combined, these elements produce a Numerically Controlled Oscillator. Unfortunately the conversion from linear to sinusoidal notation cannot be easily performed in the hardware description languages of VHDL or Verilog, because these languages lack the standardize trigonometric functions necessary to perform a sinusoidal conversion. This leads to the problem of how to create a parametizable sinusoidal conversion LUT using HDL.

However, not all computer languages lack the trigonometric function. It is very common to find trigonometric and other complex mathematic operators built-in as standard libraries for more common computing languages. These programming languages typically also support functions to export strings to files. Therefore it is possible to write a software script which can solve mathematical equations and convert the results to strings in the form of HDL statements and export the HDL to files. When these files are synthesized for hardware development, it will appear as though the mathematical equation was implemented directly in hardware. This allows users to integrate tools, libraries, and features that common computer programming languages such as C, C++, JAVA, or PERL have been able to take advantage into less robust hardware languages.

The purpose of the NCO is to generate a sinusoidal waveform. By utilizing the *SIN* function from the standard *C* programming libraries we can generate exact values to be placed in a LUT in our NCO model to translate our linear accumulation to sinusoidal phase. By allowing the program to compute the answer to Equation 3-1, we can obtain the value of each entry in the table. In this equation the integer solution *TableEntry* represents the integer value to be placed in the LUT. The term *Amplitude* acts as a scalar multiplier to stretch the normally -1 to 1 value of the *SIN* function to a large integer range. The value of *Amplitude* will be used to determine the magnitude of the LUT output. Larger DACs will be able to accommodate a larger *Amplitude* value. Increasing the *Amplitude* term by a power of two, results in a linear increase in the size of the LUT. The value of β represents the maximum possible integer input value to this table. Unlike the *Amplitude* term, doubling the β term will result in doubling the size of the resulting LUT. The final variable, *Input*, represents the value that would be seen on the input to the LUT and ranges from 0 to β.

$$TableEntry = Amplitude * \sin\left(\frac{\pi \cdot Input}{2 \cdot \beta}\right) \quad (3\text{-}1)$$

$$Amplitude = 2^{Number\ \ of\ Bits-2} \quad (3\text{-}2)$$

After the C program has computed the answer to each possible input to this equation, it can place these inputs within a VHDL case-when statement to ensure that the synthesis tool which interprets the VHDL model will treat these elements as values for a LUT. To further reduce the area coverage from the LUT, only a portion of the SIN function needs to be stored within the LUT. In the Equation 3-1, we store only one

quarter of the complete waveform and exploit the repetition which occurs every 90º in the sinusoidal function. This now means that additional logic is required to format the input to and output from this table to be directly compatible with a DAC, but the overhead required to implement this additional logic is minimal when compared to the size of the LUT overhead.

The LUT entries stored in the Equation 3-1 can be visually represented as shown in Figure 3-3. In this illustration, the term *Amplitude* is expressed as the maximum magnitude output from the LUT and β is inversely proportional to the phase step resolution in the LUT. To create the full waveform of 360º, the quarter waveform table, which represents only one quadrant of the waveform, will need to be repeated four times. The logical repetition of this LUT is visually illustrated in Figure 3-4. This figure shows how an accumulation in the quarter wave LUT can be used to express the full sinusoidal waveform. In addition, the upper two Most Significant Bits (MSB) of the phase accumulator can be used to decode which quadrant of the full waveform the LUT is currently expressing. This relationship is shown in Table 3-2.

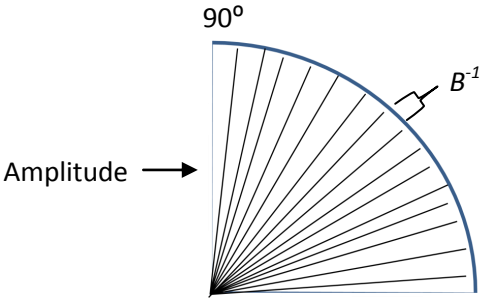

Figure 3-3: Illustration of Quarter Wave Table

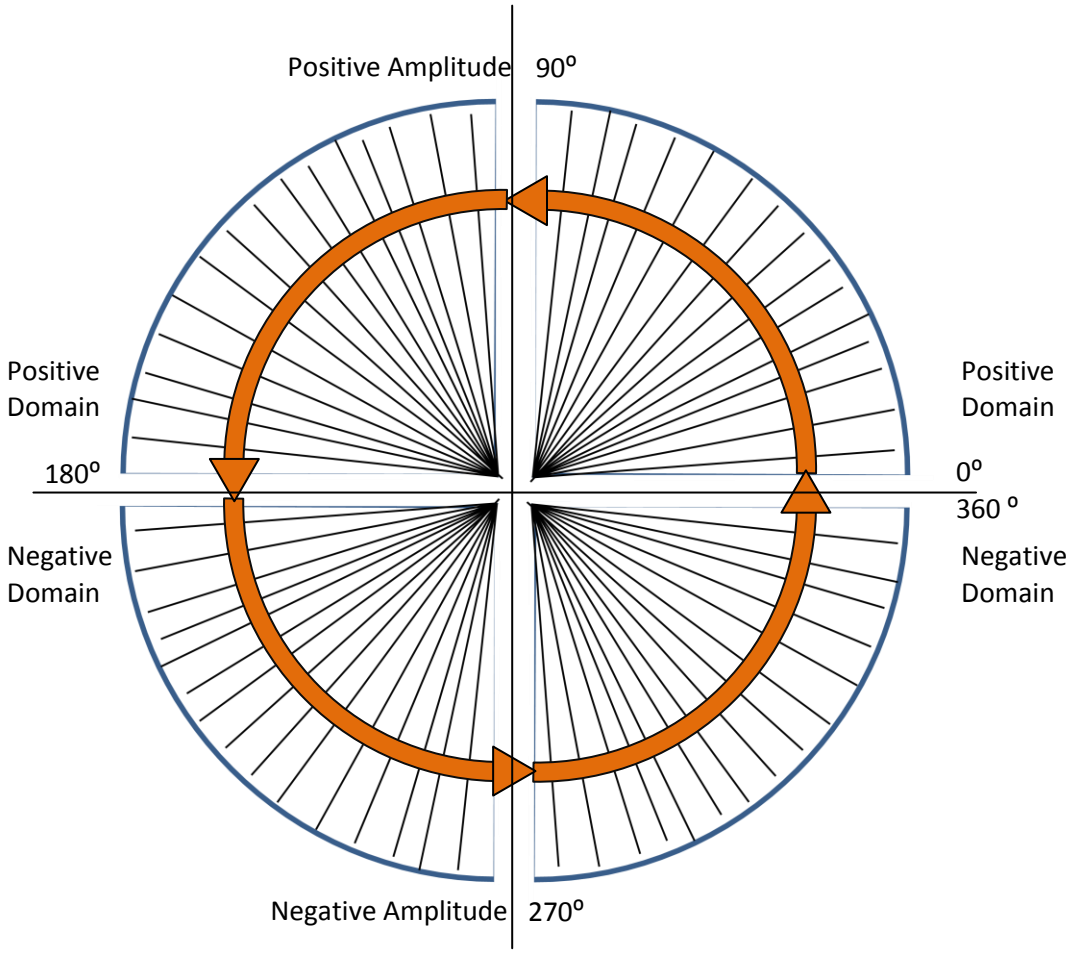

Figure 3-4: Illustration of Repeating Quarter Wave Table to Form Full Wave Table

Table 3-2: Decode Top Bits of Quarter Wave Phase Accumulator

Expressed Range	MSB	MSB-1	Direction	Output Sine Value
0º - 90º	0	0	Bottom to Top	Non-Inverted
90º - 180º	0	1	Top to Bottom	Non-Inverted
180º - 270º	1	0	Bottom to Top	Inverted
270º - 360º	1	1	Top to Bottom	Inverted

This single quarter wave encoded LUT is capable of expressing the entire range of output values in the sinusoidal waveform. However this output must be modified to do so. As shown in Table 3-2, by examining the top two MSBs of the phase accumulator, the quadrant position of the output waveform can be determined. Using this quadrant information, the direction the LUT is incremented, as accumulation continues, can be determined. This quadrant information also determines whether the output of the LUT needs to be expressed as a positive or negative result. This reduction results in a new version of the NCO model which relies on modifying the output of the LUT with generated bits which correspond to the current quadrant of the sine wave, rather than encoding these bits directly into the LUT. This is shown in Figure 3-5.

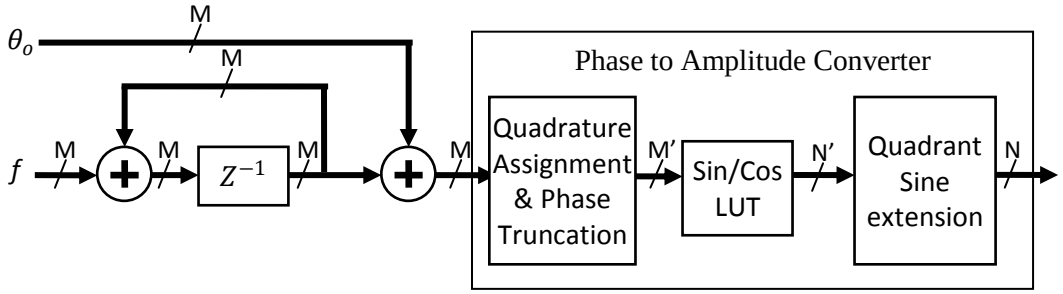

Figure 3-5: Illustration of Modified NCO model to include reduced LUT.

This reduction of the NCO provides for a smaller DDS-based TPG with no loss in resolution of the output waveform when compared to the originally purposed TPG. However there is an additional component which can be added to the NCO in the TPG to further improve resolution of measurement. As mentioned in Chapter 2, the DDS approach to generating tones has some drawbacks. The most notable is the generation of image tones and spurs resulting from the quantization of the signal, phase truncation, and harmonics of the generated tone. One method for reducing the presence of these spurious

tones is the Nicholas modified phase accumulator [17]. This component replaces the traditional phase accumulator used in the NCO. The Nicholas modified phase accumulator is not capable of removing the error which results from the DDS process, but instead redistributes the power spectrum of the spurious tones [17]. This results in many small spurious tones rather than one large tone in the DDS output.

The construction of the Nicholas modified phase accumulator utilizes one additional D Flip-flop set to toggle the input of the phase accumulator adder every other clock cycle and is illustrated in Figure 3-6. This does not affect the frequency word being added to the accumulator directly, but adds an additional component into this accumulator that acts to dither the value coming from it.

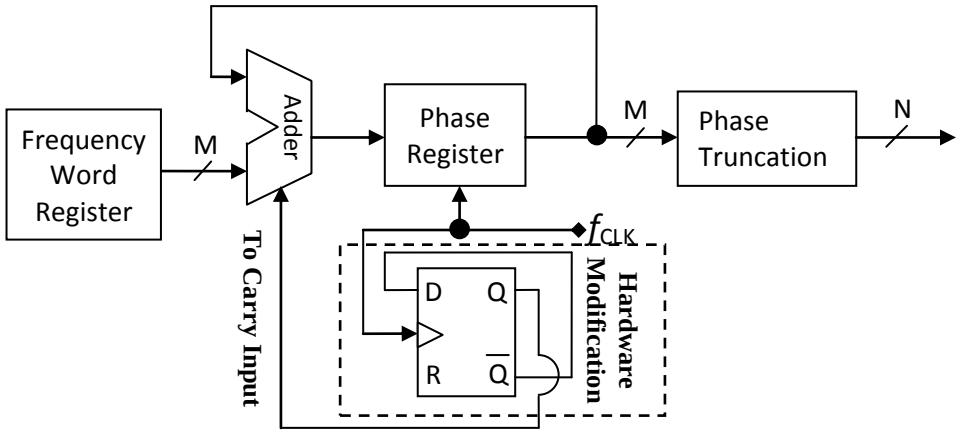

Figure 3-6: Block Diagram of Nicholas Modified Phase Accumulator [17]

The Nicholas modified phase accumulator reduces spurious tones that are generated in the DDS output, but this approach has two drawbacks. The first hindrance is that this technique introduces a fractional offset to the signal, represented as shown in Equation 3-3 [17]. In this equation, *M* represents the length of the phase accumulator. This frequency offset is smaller than the resolution of the DDS making it extremely difficult to detect.

$$f_{OFFSET} = \frac{f_{CLK}}{2^{M+1}} \quad (3\text{-}3)$$

The second disadvantage of the Nicholas modified phase accumulator is that it is possible the dithering process will introduce a small amount of error. If there is no phase truncation error in the samples, then the Nicholas modified phase accumulator will add an unnecessary dithering factor into the accumulation and introduce a small amount of unnecessary noise into the generated tone [17]. By utilizing an external program to generate this model, the user can determine if trade-offs of this dithering tool represent significant improvement to that particular instantiation of the BIST model and should be included.

Shown in Figure 3-7 is the resulting testing waveform generated by the DDS-based TPG of this mixed-signal BIST design and captured by a spectrum analyzer. In the center of this image is the generated single tone produced by the TPG at 14.4 kHz. Captured along with this main tone are four spurious tones generated due to the phase truncation stage within each NCO. These are the spurious tones which the Nicholas modified phase accumulator is designed to reduce. Shown in Figure 3-8 is the same tone generated by the mixed-signal BIST with the exception that Nicholas modified phase accumulators are used in the NCOs of the TPG rather than traditional accumulators. The four sharp spur tones which are present in Figure 3-7 have been broken into many less intense tones in Figure 3-8 due to the modification to the phase accumulator which minimizes the strength of the worst case spur. Shown in the top right corner of both figures is the measurement of the signal strength difference between the main tone and the strongest spurious tone. In the example implementation of the Nicholas modified phase accumulator, an improvement of 2.79 dB spurious free range was achieved.

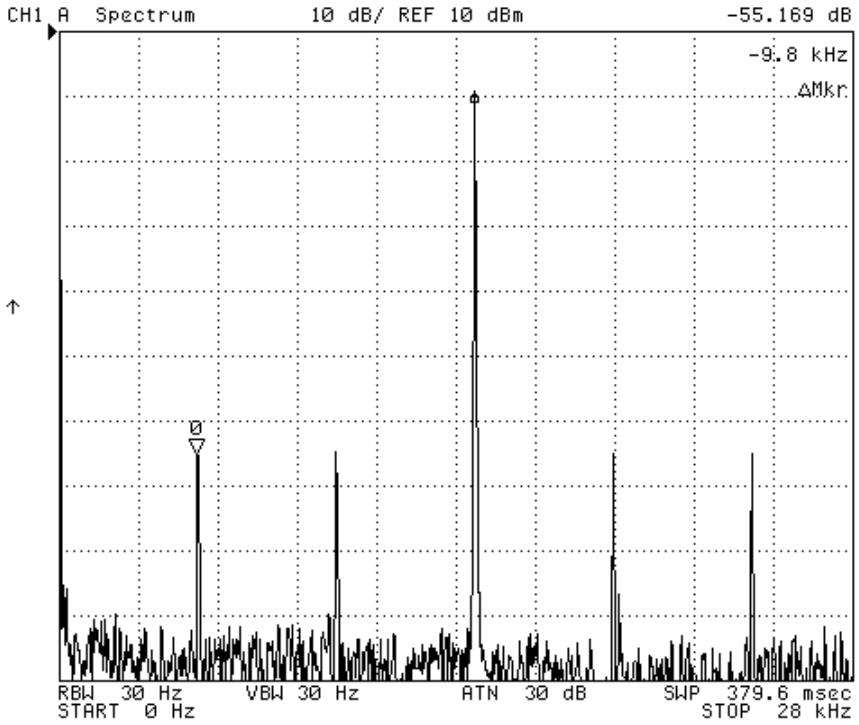

Figure 3-7: Spurious Free Dynamic Range of 55.169 dBc in a 14.4 kHz Tone Generated and by the Mixed-Signal BIST Without Nicholas Modified Phase Accumulator

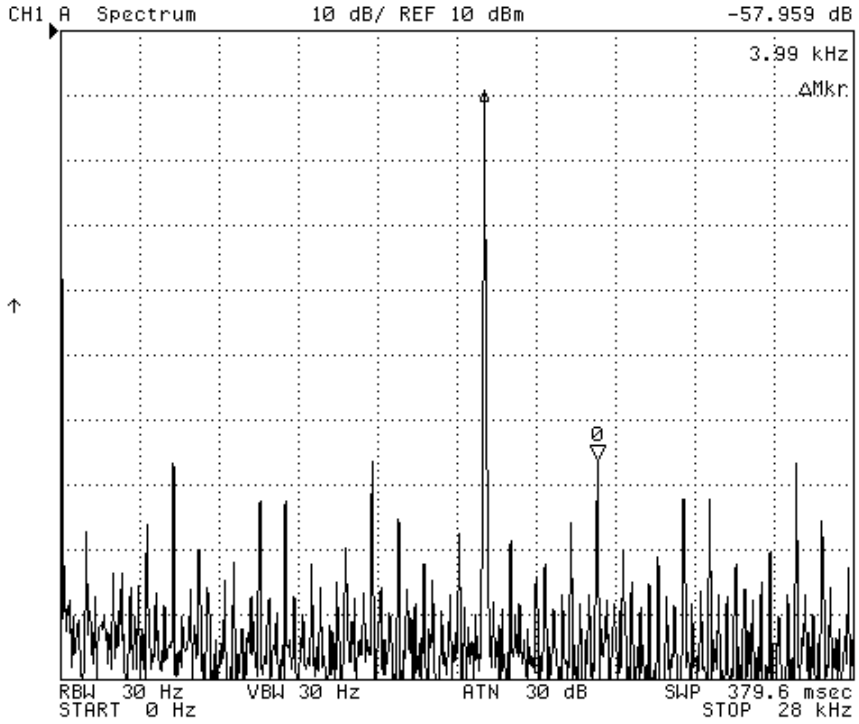

Figure 3-8: Spurious Free Dynamic Range of 57.959 dBc in a 24.4 kHz Tone Generated by the Mixed-Signal BIST With Nicholas Modified Phase Accumulator

3.1.2 Output Response Analyzer

The Output Response Analyzer is responsible for measuring the response from the DUT and calculating the results of the test. In the discussed model, the ORA achieves this calculation by using only two multiplier/accumulator (MAC) pairs to calculate the DC_1 and DC_2 terms, which represent the complex terms of magnitude and phase of the measured signal. To implement this we need a multiplier model fast enough to be able to handle high speed signals, yet small enough that it won't become a burden on the area overhead.

The multiplier suggested in this thesis is a modern variant of classic multiplier design. The most common multiplier models are the array multiplier, Baugh-Wooley multiplier and Booth multiplier. Background on these can be found in Chapter 2. When the Wallace tree reduction algorithm is combined with the Modified-Booth algorithm, the result is a fast and efficient signed multiplier which utilizes only half the typical number of partial product summations. By using enhancement and reduction techniques, it is possible to compress the size of this multiplier to an even greater extent. The first step in modifying this multiplier is to enhance its speed of calculation by pipelining each stage within the multiplier. The key points to pipeline would be between each stage of partial product addition and after the calculation of Booth encoded partial products. The next stage of modification involves a novel reduction technique.

This reduction technique is centered around the axiom that any two's complement number added to its negative equivalent will equate to zero. Therefore, adding a number and its two's complement, along with the partial products normally found in the multiplier will be equivalent to just adding the partial products. But these additional bits

from the additional negative two's complement number will allow a logical reduction of partial product bits. The negative number, -2^{N+1}, can be logically added to the generated partial products that come from the Modified-Booth algorithm, where *N* represents the size of the multiplicand in a two's complement format. This number will appear as all ones at and above the position *N+1* and all zeros below.

The first step in reduction is to take the 2's complement number and add it to the padding that exists in each partial product row. Any positive sign-extended number in the multiplier will normally be padded with zeros. Adding this padding of all zeros to a number of all ones will result in all the padding bits and the sign bit becoming all ones. Conversely any multiplier partial product which happens to be negative will be padded with ones. If this number of all ones is added to the negative padding of all ones, the result will be that all the padding bits will still be all one, but the lowest order sign bit will have inverted from one to zero. If we keep the sign bit and the first padded bit from each stage in this process, then this algorithm can be simplified. This summation of each reduction phase is illustrated in the Figures 3-9 through 3-14. Figure 3-9 illustrates the overall logical combination of the reduction factor with the partial products rows. Figure 3-10 shows the only two possible outcomes for each stage of reduction. Figure 3-11 through Figure 3-14 individually illustrate the reduction which takes place on each partial product's sign extension bits. The highlighted area in these illustrations represents the area that will be reduced in the following illustration.

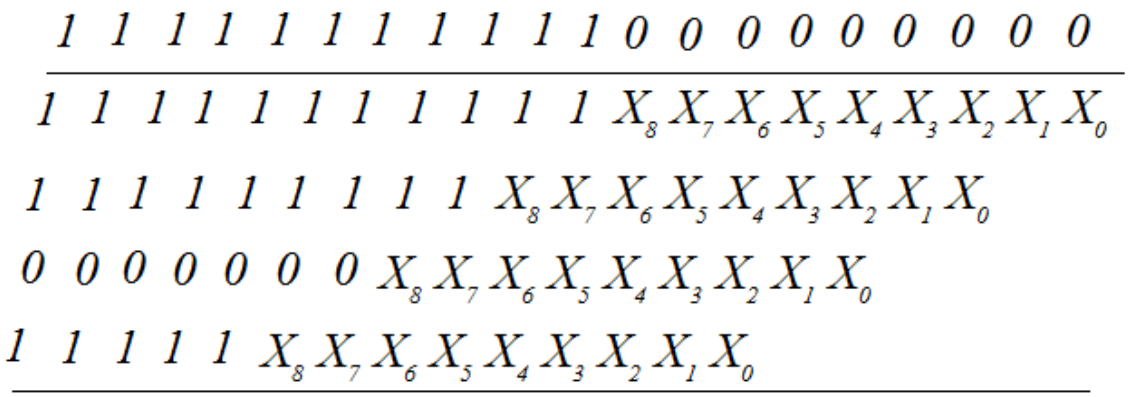

Figure 3-9: The Logical Addition of the Two's Complement Term with Partial Products. The First Row Represents the Correction Factor While Each Additional Row Represents the Partial Products Rows.

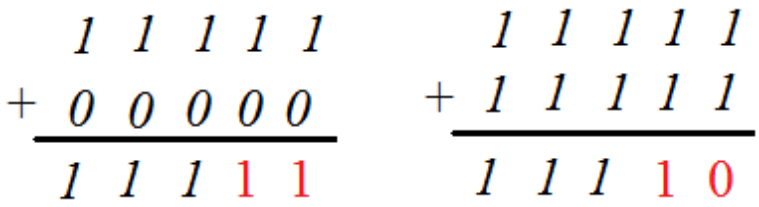

Figure 3-10: The Two Possible Outcomes for Each Stage of Reduction

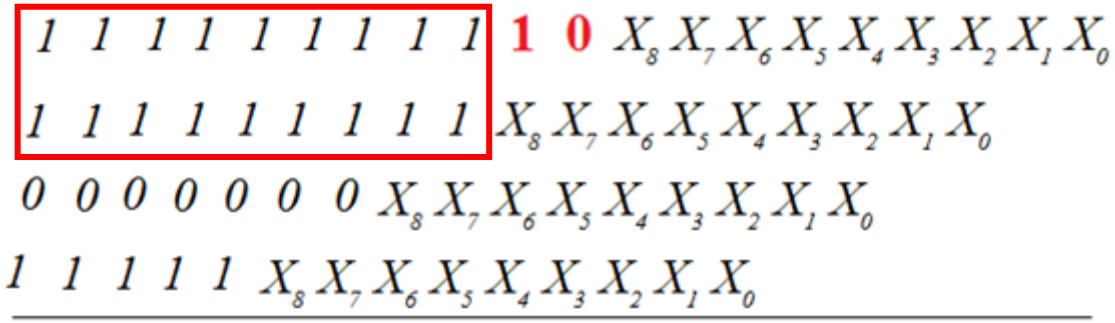

Figure 3-11: The Logical Addition of Offset Term with First Negative Partial Product. The Highlighted Area Represents the Next Area to Be Reduced.

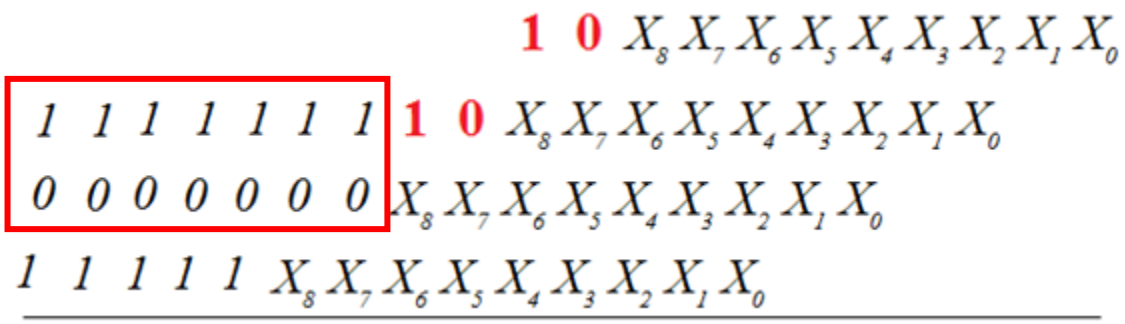

Figure 3-12: The Logical Addition of Offset Term with Second Negative Partial Product. The Highlighted Area Represents the Next Area to Be Reduced.

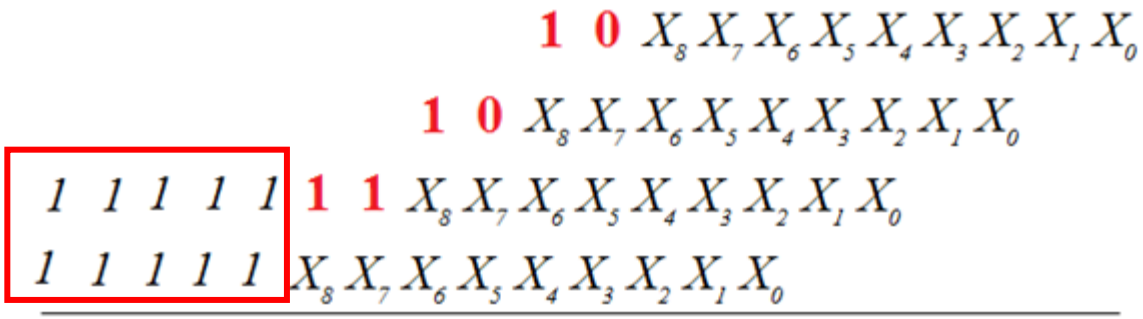

Figure 3-13: The Logical Addition of Offset Term with Third Positive Partial Product. The Highlighted Area Represents the Next Area to Be Reduced.

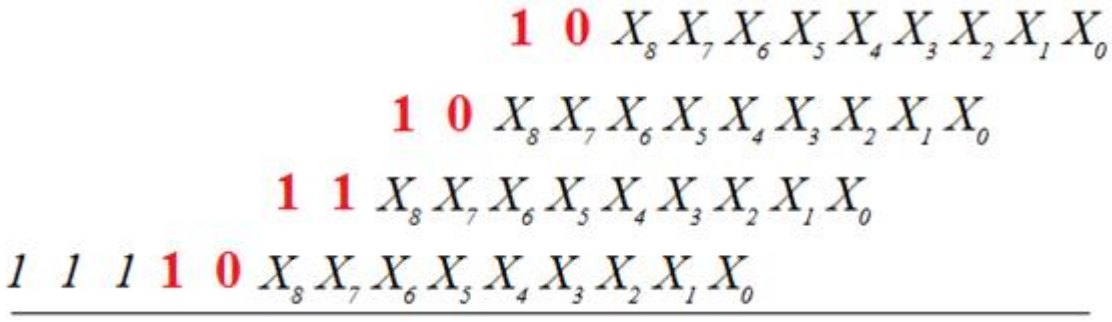

Figure 3-14: The Logical Addition of Offset Term with Final Negative Partial Product.

In this new model the MSB, the sign bit, of each partial product is inverted and a binary one is added to the MSB+1 position. All higher order bits that were used for sign extension can be modified to zero. As adding zeros will always yield zero, we can now logically remove this extra adder logic from the circuit. This takes what would have been a long partial product with sign extension bits up to N+M bits long and reduces each partial product down to a fixed N+2 bits. The two MSB can be expressed as shown in Equations 3-4 and 3-5.

$$X_{MSB} = not(X_{MSB}) \tag{3-4}$$

$$X_{MSB+1} = 1 \tag{3-5}$$

In an 8x8 bit modified Booth multiplier, this can remove twelve no longer needed adder units from the partial products adding phase. Larger multipliers will see an even larger reduction in sign extension bits. For example, a 25x25 bit modified Booth multiplier will be reduced by one hundred and forty-four bits. This represents approximately a 22% reduction in partial product bits from the original modified 25x25 Booth model. For symmetric multipliers, the number of reduced partial product bits is expressed Equation 3-6.

$$Reduced\ Bits = \sum_{n=1}^{\frac{Multiplier\ \ Size}{2}} \left[\frac{Multiplier\ \ Size}{2} - 2n\right] \tag{3-6}$$

Table 3-3. Reduction in Partial Product Bits by Size of Modified Booth Multiplier

Size of N in NxN	8	9	10	11	12	13	14	15	25	500
Original Number of Bits	76	95	115	138	162	189	217	248	663	250250
Reduced bits	12	16	20	25	30	36	42	49	144	62250
Percent Reduction	16%	17%	17%	18%	19%	19%	19%	20%	22%	25%

This reduction in size of the partial product terms comes at a price. To reduce these higher order sign extension bits, we must add one more number to the partial product to compensate for adding in this previous value. This new number is simply the two's complement of the previous value. This will always be a power of two with the value of 2^{N+1} in the Modified-Booth model. The solution to resolving this issue is to add this partial product within the Wallace tree structure as a carry in bit. This will result in simply converting one half-adder into a full-adder. This has the added benefit of requiring no additional partial product stages.

These improvements represent a significant size reduction for large MACs in the ORA. However, size reduction is not the only benefit of these MACs. This design can be further sped up by utilizing registers between each partial product addition stage to pipeline the multiplication. This will result in a faster multiplier, but multiple clock cycles will be required until the output is valid. This will cause an initial latency, but high overall bandwidth is achieved. Because this multiplier utilizes the Booth architecture to generate partial products, only half the partial products stages of a normal multiplier are needed. This will result in half the number of pipelining registers when compared to a traditional multiplier.

3.2 Built-In Self-Test Test Results

To verify that this program operates properly and to obtain a measurement from this mixed-signal BIST model, this design needed to be implemented in hardware. To achieve this, the digital portion of the BIST was generated using the C program,

MSBISTGen.exe, discussed in this chapter. This digital portion of the BIST was then synthesized within a Spartan-3 Field Programmable Gate Array (FPGA) manufactured by Xilinx. However, these FPGAs lack any analog components making it impossible to implement the complete BIST in one digital IC. Because the FPGA contains only the digital portion of the BIST design, a Printed Circuit Board (PCB) was developed containing the necessary analog components including the DAC, ADC, analog multiplexor, and sample DUT to interface to the FPGA. The completed mixed-signal BIST assembly is shown in Figure 3-15.

Figure 3-15: Image of Completed Analog Testing Printed Circuit Board Connected to the Spartan-3 FPGA.

This completed assembly was then used to test the performance of the sample DUT located on the PCB. This DUT consisted of two separate parts. The first element in the DUT was that of a tunable lowpass filter, where the cutoff point can be modified. The second element was a tunable common collector amplifier where supply voltage can be modified to clip the peaks of the waveform and introduce signal distortion. The original design goal of this mixed-signal BIST was to be able to measure the frequency response, linearity, and noise margins of an analog device. Using these two tunable parts of the DUT, it is possible to compare each of these measurements taken by the mixed-signal BIST to those expected by simulation and measurement with external equipment.

To communicate with the mixed-signal BIST, the MSBISTGen.exe program placed a Serial Peripheral Interface (SPI) bus within the model to serially stream results out of the BIST circuit and test setup information into the BIST circuit. Using this SPI interface and a computer which can communicate over this interface, a simple program was written to serially stream test information and collect test results for this BIST.

The first of these desired tests is that of the frequency response. Of the two parts of the DUT on the analog PCB, the tunable first order low pass filter has a known frequency response. Using the mixed-signal BIST to measure a range of distinct frequency points, the frequency response of the DUT was measured for different cutoff points of the tunable filter. The results of this measurement are shown in Figure 3-16. Ten frequency response plots obtained from the measurements by the mixed-signal BIST model are superimposed in this figure. These results show that as the cutoff point is moved to a higher frequency in each successive test, the mixed-signal BIST accurately

detects the change in response. In addition, these response curves match the expected response from simulation with a 20dB/decade slope after the cutoff point.

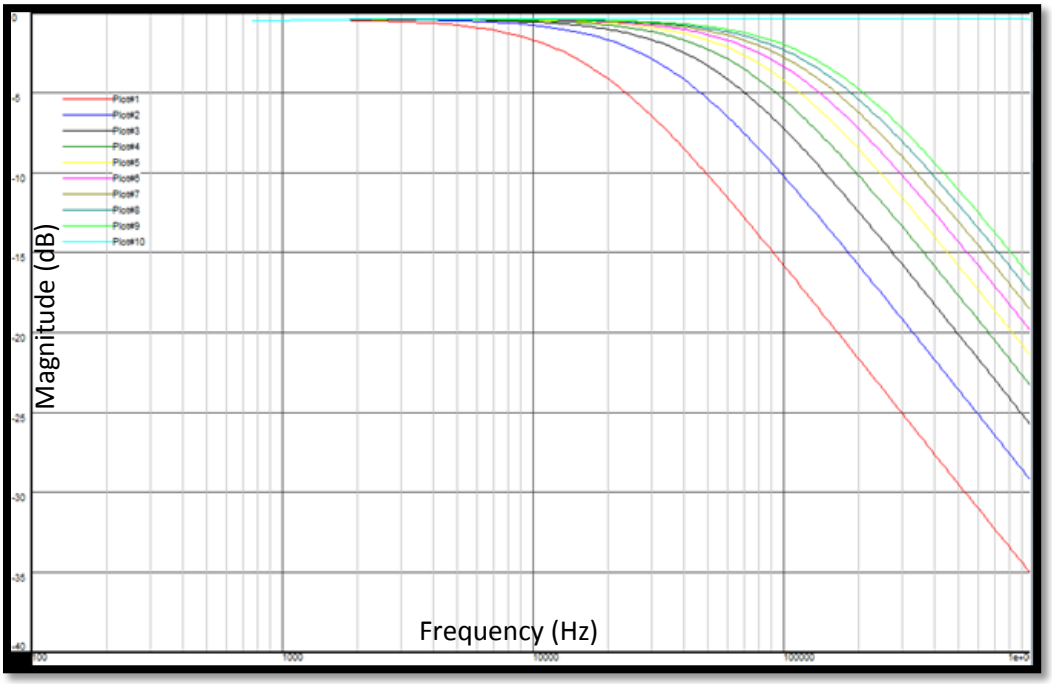

Figure 3-16: Frequency Response Measurement of the DUT from by the Mixed-Signal Built-In Self-Test

The second measurement of interest for the mixed-signal BIST is that of the linearity measurement. To obtain the linearity measurement, the BIST model will generate two tones through the DUT, specifically a common collector amplifier, which is designed to cause distortion. The BIST will then measure the signal strength of one of the two generated tones and the signal strength at the third order intermodulation frequency. The difference between these two tones represents the ΔP component in the linearity measurement. Using the tunable amplifier, the linearity of the device was swept between the extremes of very linear to nonlinear. These measurements can be observed in Figures 3-17 and 3-18, respectively.

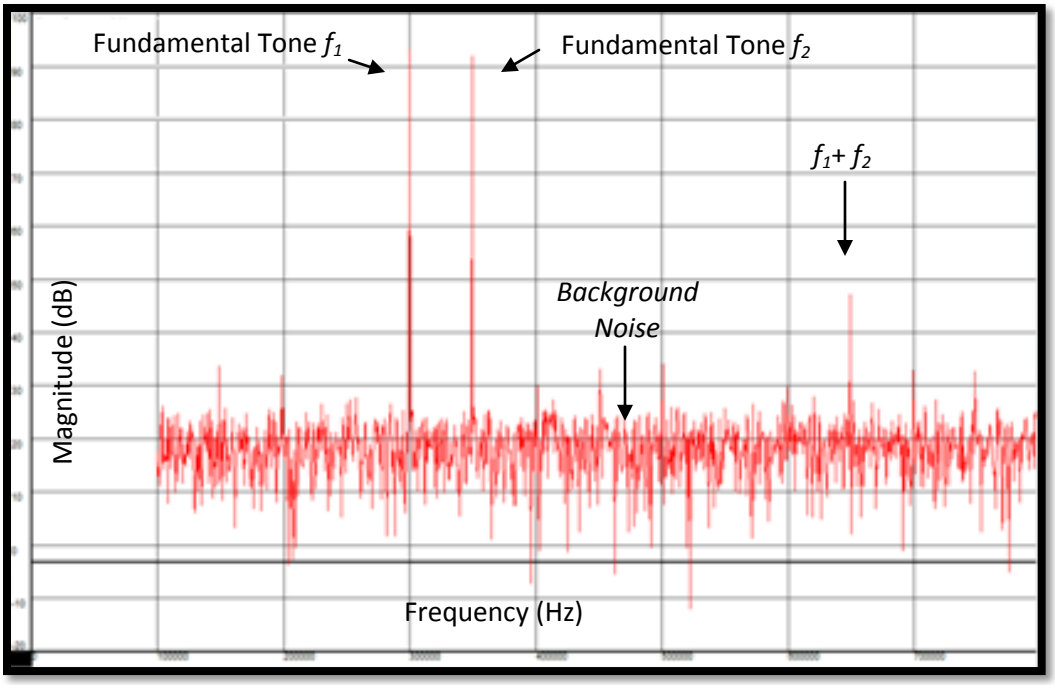

Figure 3-17: Linear Response from DUT Measured by the Mixed-Signal BIST.

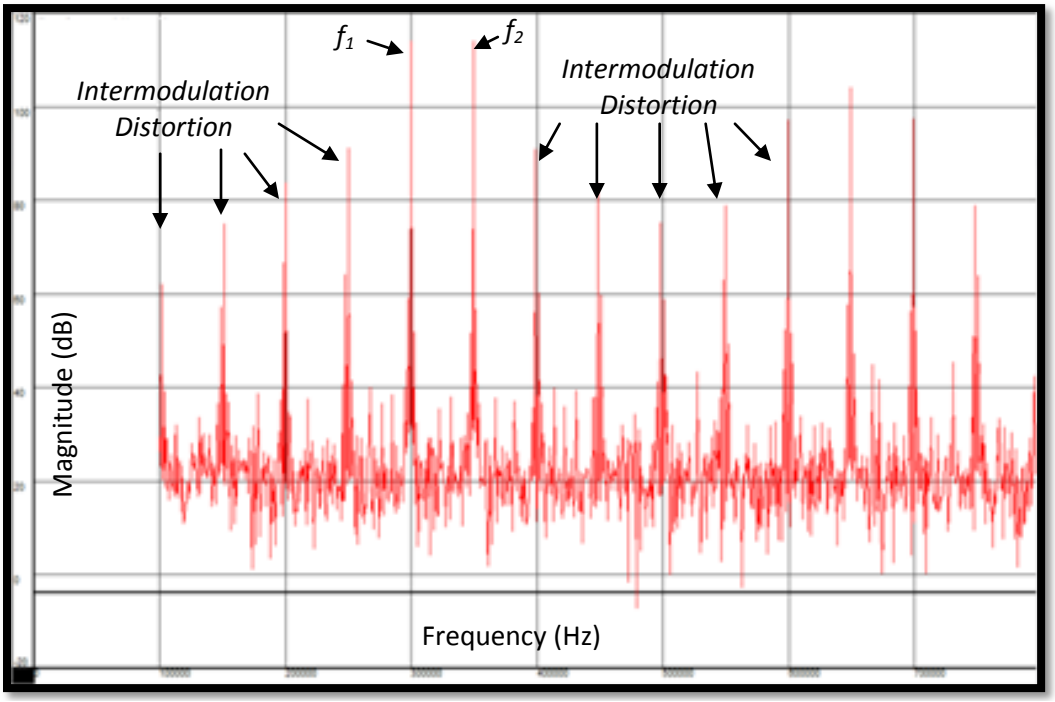

Figure 3-18: Non-Linear Response from DUT Measured by the Mixed-Signal BIST

In Figure 3-17, a linearity measurement is observed through the DUT. In this plot, the two generated tones, f_1 and f_2, can clearly be observed in the center of the image. The next strongest observable point is the frequency at the summation the summation of these two tones, f_1+f_2. No observable Intermodulation Distortion is seen in this plot. Figure 3-18 represents a measurement through a non-linear DUT. In this measurement, signals f_1 and f_2 can still be seen in the plot; however, the plot is saturated with distortion at the points of intermodulation between the fundamental tones. The mixed-signal BIST model was used to sample many varying degrees of non-linear by measuring the difference in signal strength between the fundamental and third order intermodulation term. Table 3-3 shows a sampling of these measurements. To verify that these measurements were accurate, these tests where simultaneously measured using an Agilent 4395A spectrum analyzer.

Table 3-3: Linearity Measurement Comparison of Mixed-Signal BIST Model and Spectrum Analyzer

Sample #	BIST Measurement	Spectrum Analyzer Measurement	Difference
1	13.033 dB	13.08 dB	-0.047 dB
2	13.110 dB	13.14 dB	-0.03 dB
3	13.365 dB	13.4 dB	-0.035 dB
4	13.845 dB	13.87 dB	-0.025 dB
5	14.590 dB	14.55 dB	0.04 dB
6	16.103 dB	16.12 dB	-0.017 dB
7	19.083 dB	19.07 dB	0.013 dB
8	24.219 dB	24.15 dB	0.069 dB
9	27.217 dB	27.01 dB	0.207 dB
10	30.321 dB	29.8 dB	0.521 dB
11	31.950 dB	31.25 dB	0.7 dB
12	36.024 dB	34.9 dB	1.124 dB

The final measurement of interest for the mixed-signal BIST is that of the Signal to Noise Ratio (SNR) and the Noise Figure (NF). To produce this test a single fixed tone was generated and the mixed-signal BIST measured the signal strength at many noise points and again at the generated tone. By taking the difference in the generated tone and average noise measurement, the SNR can be obtained. If this SNR measurement is also performed while bypassing the DUT, the NF can be obtained from the difference between the two SNR measurements. In Figure 3-19, an SNR measurement obtained from data collected from the ORA of the BIST shows a 44dB SNR measurement was obtained during the measurement of the DUT. Additionally, the mixed-signal BIST was used to perform multiple SNR measurements. Each measurement was recorded and compared to an SNR measurement obtained from an Agilent 4395A spectrum analyzer. The comparison between these measurements can be seen in Table 3-4.

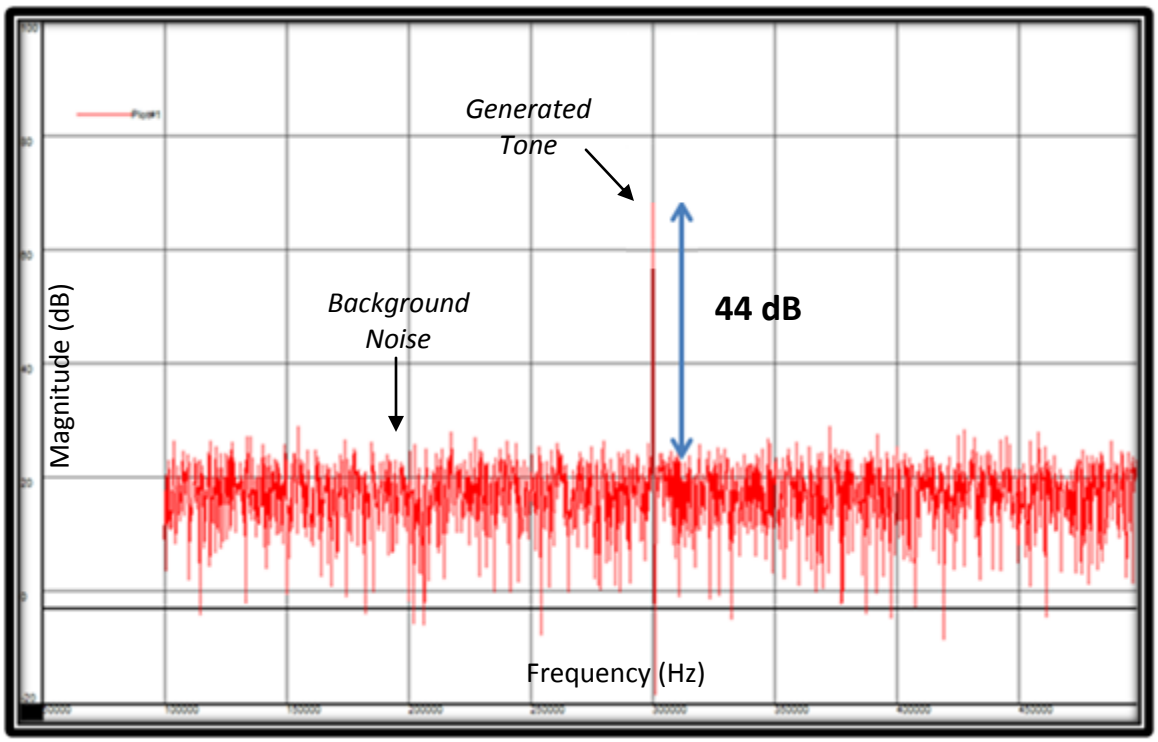

Figure 3-19: Signal to Noise Ratio Measurement from Data Collected from the Mixed-Signal BIST.

Table 3-4: SNR Measurement Comparison of Mixed-Signal BIST Model and Spectrum Analyzer

Sample #	BIST Measurement	Spectrum Analyzer Measurement	Difference
1	32.45 dB	32.78 dB	-0.33 dB
2	34.57 dB	34.97 dB	-0.4 dB
3	36.98 dB	36.15 dB	0.83 dB
4	38.91 dB	38.21 dB	0.7 dB
5	41.34 dB	40.38 dB	0.96 dB

Chapter Four

In-Circuit Calculations of Analog Characteristics

The primary purpose of a Built-In Self-Test system is to test a circuit component for faults to verify that the device is operating properly. This chapter focuses on additions to the originally proposed mixed-signal Built-In Self-Test model to further aid in performing fault testing on-chip. These additions include the development of digital components to interpret the accumulated results of the Output Response Analyzer into a magnitude and phase measurement as well as perform analog measurements on-chip. These new designs were implemented mainly to provide a rapid analysis of on-chip analog measurements within the mixed-signal Built-In Self-Test model. A secondary goal of these designs was to provide a flexible model that can easily be parameterized.

4.1 Built-In Self-Test On-Chip Calculation Circuit

In the current mixed-signal BIST model, the ORA captures results from each measurement by multiplying the measured waveform by the in-phase and out-of-phase waveform components of the frequency of interest and then accumulates the results. This obtains the accumulated values of the in-phase and out-of-phase components of a given frequency within the originally measured waveform. These values contain the relative phase and magnitude information necessary to evaluate tests such as linearity, noise

measurement, and frequency response. However, the form they are in at the output of the ORA does not allow these measurements to be directly evaluated for these tests. To obtain magnitude and phase information from these two extracted ORA components, DC_1 and DC_2, an additional component in the BIST architecture will be necessary.

Presented in Chapter 2 are three methods to calculate the magnitude from the ORA's results. Of these three methods, this thesis utilizes the approach for calculating the magnitude using the square root of the summation of the squares of the in-phase and out-of-phase components to obtain the magnitude of the measured signal strength. To perform the calculation, an additional circuit component which utilizes the Coordinate Rotation Digital Computing (CORDIC) algorithm was designed. This CORDIC technique can be used to calculate a variety of mathematical operations, including the square root of the summation of squares. To perform each computation the CORDIC approach utilizes a system of shift and add or shift and subtract operations to calculate the result of the operation in accordance with the unified CORDIC model. Additionally the CORDIC approach can be used to extract the relative phase measurement of the measured waveform in a method similar to how the magnitude is extracted from the in-phase and out-of-phase components.

Once the CORDIC Calculation Unit (CCU) has extracted the magnitude information from the DC_1 and DC_2 terms, the resulting magnitude can be used to calculate the results of a variety of analog tests. The primary tests of interest in this mixed-signal BIST design are linearity, noise, and frequency response.

To perform a linearity test, two tones are generated simultaneously, and the Intermodulation Distortion (IMD) between these tones is measured. To measure the IMD,

the signal strength between one of the fundamental tones, f_2, is compared to that at the Third Order Intermodulation (IM3) point at $2f_2$-f_1. The ratio of these two signal strengths, ΔP, represents our measurement of linearity. From this measurement, a variety of abstractions about the linearity of the system, as shown in Chapter 2, can be derived. To implement this mathematical comparison in hardware, it is necessary to divide the signal strength of the fundamental tone by that of the IM3 tone to obtain ΔP. To express ΔP in decibel, as it is usually shown, the result can be expressed as shown in Equation 4-1.

$$\Delta P(db) = 20 * \log_{10}\left(\frac{Signal\ Strength(f_2)}{Signal\ Strength\ (2f_2 - f_1)}\right)$$

To perform a noise measurement, a single tone is generated and passed through the DUT. The signal strength of the tones in the noise regions can then be extracted using the CCU and averaged. To average these signals an accumulator can be used to collect each measurement, and a divider can be used to divide by the number of samples collected from this accumulated value. The next step in obtaining a noise measurement is to measure the signal strength of the originally generated tone. This is achieved using the CCU to interpret the response from the ORA. By taking the ratio between the power in the fundamental tone and the average power in the noise measurements, the Signal to Noise Ratio (SNR) can be obtained. This is expressed in Equation 2-1.

To perform the frequency response test for either phase or magnitude, the frequency of interest must be swept over multiple tones, making multiple measurements. Unlike linearity and SNR measurements, there is no single answer to frequency response. Instead the result is a list of distinct measurements typically expressed as a plot. To that end, the CCU is capable of extracting the relative phase and magnitude from each ORA

measurement for the generated signal strength. Each measurement can then be collected from the CCU individually for either phase response or magnitude response of the DUT.

This entire approach can be summarized into one digital model. This can be constructed using a single divider, two accumulators, and two multiplexers in conjunction with the CCU. This model is illustrated in Figure 4-2. The divider provides two main functions. The first operation the divider can perform is to divide a current magnitude measurement from the CCU by a previous measurement. This is essential to perform a comparison between tones in the linearity test, SNR measurement, and magnitude response. The second operation the divider provides is the ability to average the noise measurements in the SNR test by dividing the accumulated noise by the number of noise samples collected. The first accumulator accumulates each noise measurement magnitude as well as any past magnitude measurements, while the second accumulator records the number of noise samples collected. However, unlike the magnitude value, obtaining a summation of the noise phase has little statistical information about the performance of the system. Therefore, it is not necessary to perform the heavy calculations to obtain average noise phase as the proposed circuit does for magnitude. The extracted phase information is, however, necessary for finding the phase of a specific signal of interest. The remaining multiplexors can be used for controlling the data path. An additional component, the Logarithmic Translation Unit, is used to express the final test results in decibels.

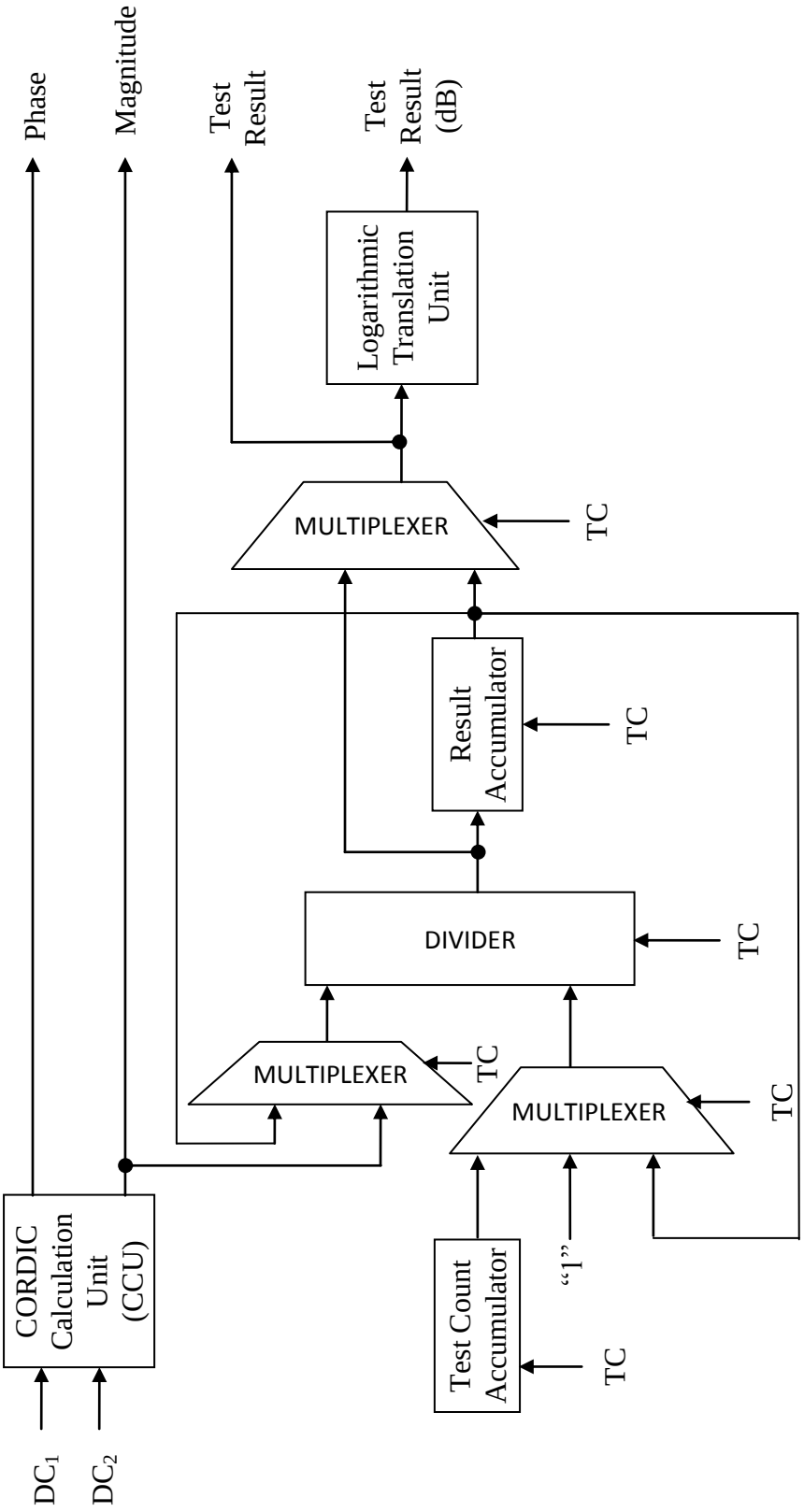

Figure 4-1: Complete In-Circuit Test Calculator Using CORDIC Calculation Unit

4.1.1 Coordinate Rotation Digital Computing Calculation Unit

The general relationship between the values in the ORA and desired magnitude and phase measurement can be summarized as shown in Equations 4-2 through 4-4.

$$ORA\ Result = DC_1 + iDC_2 = \frac{AN}{2}\cos(\theta) + \frac{AN}{2}\sin(\theta) \quad (4\text{-}2)$$

$$Magnitude = \sqrt{\left({DC_1}^2 + {DC_2}^2\right)} \quad (4\text{-}3)$$

$$Phase = \tan^{-1}\left(\frac{DC_2}{DC_1}\right) \quad (4\text{-}4)$$

Performing these operations in hardware can prove to be a difficult challenge, especially for models that can be parameterized or utilize large signals. Even implementing an inverse tangent Look-Up Table (LUT) for two 32-bit *DC* terms would consume a very large amount of area overhead. Therefore, an alternative to traditional implementation methods is needed to efficiently calculate the magnitude and phase.

The Coordinate Rotation Digital Computing (CORDIC) algorithm is used in many applications to perform basic mathematical functions and has been used in many pocket calculators since the Hewlett Packard HP 35 and within processors since the Intel 8087 [29]. In this thesis, the CORDIC algorithm will find a different application in signal processing. The CORDIC algorithm will be used to construct a physical circuit to transform the output of the Selective Spectrum Analysis (SSA) based ORA into the magnitude and phase measurements required to evaluate the results of the three desired analog tests: linearity, frequency response, and noise measurement.

This relationship between the ORA outputs, magnitude, and phase can be represented graphically as depicted in Figure 4-2. DC_1 and DC_2 can be depicted on the unit circle as two orthogonal vectors. The vector addition of these two *DC* terms will

result in a new composite vector. The length of this new vector represents the magnitude, while the phase is depicted as the angle from DC_1's vector to the composite vector. Because these values can be expressed in terms of the unit circle, the circular CORDIC is able to convert between these expressions for the *DC* terms, magnitude, and phase.

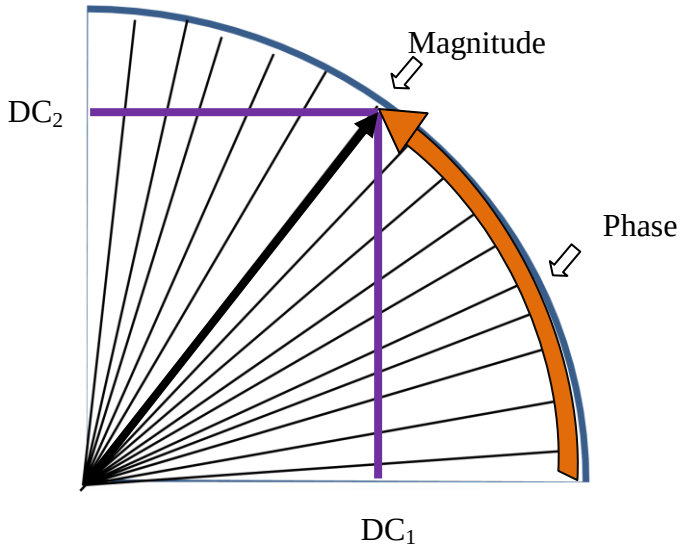

Figure 4-2: CORDIC Relationship on Unit Circle

To transform the in-phase and out-of-phase components, DC_1 and DC_2 respectively, of the ORA into the magnitude value shown in Equation 4-3, the square root of the summation of squares operation needs to be applied to the values of DC_1 and DC_2. To achieve this operation the circular CORDIC needs to be modified slightly from the example listed in Chapter 2. Rather than attempting to find the magnitude for a fixed angle, the square of the sum of squares circular CORDIC will allow variable values to be loaded into the two *X*-*Y* coordinates so the starting angle is set. Then the component

values will be rotated in the CORDIC model until the *Y*-axis value approximates zero. This leaves the *X*-axis value as the approximation of the magnitude.

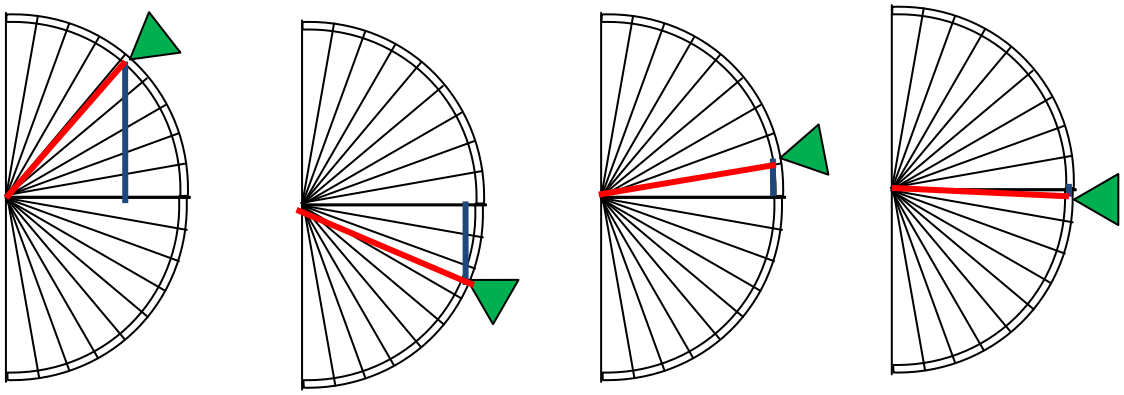

Figure 4-3: Illustration of CORDIC Rotational Procedure Minimizing Y-Axis Value to Maximize X-axis Value

As observed in Figure 4-3, the red line represents true magnitude of the X-axis value and Y-axis value together. As the Y-axis value, represented by the blue line, begins to shrink away, the X-axis value has to grow to keep a constant magnitude. When the Y-axis value become significantly small, the X-axis value can be said to approximate the actual magnitude.

The first step in the square of the sum of squares CORDIC is to load the starting values of DC_1 and DC_2 into two registers within the model. Based on the sine values of DC_1 and DC_2, the quadrant of the unit circle which the vector answer will be in can be established. However, this circular CORDIC algorithm requires that vector rotation begins at angle 0º. Therefore to rotate the vector to the correct starting place, the *Register I* and *Register Q* are loaded under the given conditions shown in Equations 4-5 and 4-6.

$$Register\ Q = \begin{cases} DC_1, & if\ DC_2\ is\ positive \\ -DC_1, & if\ DC_2\ is\ negative \end{cases} \quad (4\text{-}5)$$

$$Register\ I = \begin{cases} -DC_2, & if\ DC_2\ is\ positive \\ DC_2, & if\ DC_2\ is\ negative \end{cases} \quad (4\text{-}6)$$

Once the registers have been loaded, the CORDIC will begin the successive approximation process to obtain the correct magnitude. For the first iteration of this approximation *Register I* will have the absolute value of *Register Q* added to it, while *Register Q* will be updated to contain the absolute value of *Register Q* with the value of *Register I* subtracted away. Because of the conditional loading in the first step of this CORDIC calculator, the value in *Register I* now contains the magnitudes of |DC1| +|DC2| added together. This is however not the square root of the sum of squares, but the CORDIC's first approximation towards that goal. On the following iterations, the values of *Register Q* and *Register I* are combined again to their complement register, however the values to be added are shifted right by *N*-1 bits before the addition, where *N* represents the iteration number in the CORDIC model. These iterative steps will follow the form in Equations 4-7 and 4-8. For each successive iterative step, the value in *Register I* will always be positive and continue to increase as it begins to approximate the value of the square root of the sum of squares. The value *in Register Q* represents the relative overshoot in the CORDIC calculation. For each iteration, the magnitude of the value in *Register Q* will continue to decrease toward zero as *Register I* becomes more precise.

$$Register\ I = Register\ I + abs(Register\ Q \gg [N-1]) \quad (4\text{-}7)$$

$$Register\ Q = abs(Register\ Q) - (Register\ I \gg [N-1]) \quad (4\text{-}8)$$

The end result is an approximation of the square root of the sum of squares of DC_1 and DC_2 stored in *Register I*. This resolves the issue of generating the magnitude; however, another needed calculation is the phase of the signal. In reality, the calculation through the CORDIC process to obtain the magnitude coincidently generates the phase rotation information.

Unlike *Register I*, which is always positive and always increasing, *Register Q* will contain signed numbers and is always decreasing in magnitude. The sign value of *Register Q* at each iteration correlates to the direction of phase change that will be added to the current approximation to express the angle of the resulting waveform. The actual value of phase information added in each iteration is a function of the iteration number rather than the values in *Register I* and *Register Q*. If *Register Q* contains a positive value, this means that the resulting waveform has overshot its goal and needs to reduce the value of phase. The next phase increment will be subtracted from the current approximation in this case. A negative value for *Register Q* indicates the phase value is too small and needs the next phase value to be added to it. This change in phase for each iteration is expressed in Equation 4-9.

$$Phase(N) = Phase(N-1) - \text{sign}(\text{Register Q}) * \tan^{-1}(\frac{1}{2^{N-1}}) \qquad (4\text{-}9)$$

The only relevant information of importance for obtaining the direct calculation of phase in each is the sign value, or Most Significant Bit (MSB), of *Register Q* in each iterative step. This bit can be stored within its own separate phase register during the iterative process of calculating the magnitude. The final value of the phase register is capable of expressing any value from -180º to +180º. A LUT can even be implemented, if needed to translate the arctangent encoded value into a decimal format. Equation 4-9

can also substitute for the formula for translating between the phase register and decimal phase value in a LUT. Table 4-1 shows an example of a CORDIC unit running for ten iterations and obtaining ten phase bits. The complete model for the CORDIC Calculation Unit is illustrated in Figure 4-4.

Table 4-1: CORDIC Calculation Unit Phase Bits Weight for First Ten Phase Bits

Iteration	Generated Phase Bit	Bit Weight
1	MSB	90.0000º
2	MSB-1	45.0000º
3	MSB-2	26.5651º
4	MSB-3	14.0362º
5	MSB-4	7.1250º
6	MSB-5	3.5763º
7	MSB-6	1.7899º
8	MSB-7	0.8952º
9	MSB-8	0.4476º
10	MSB-9	0.2238º

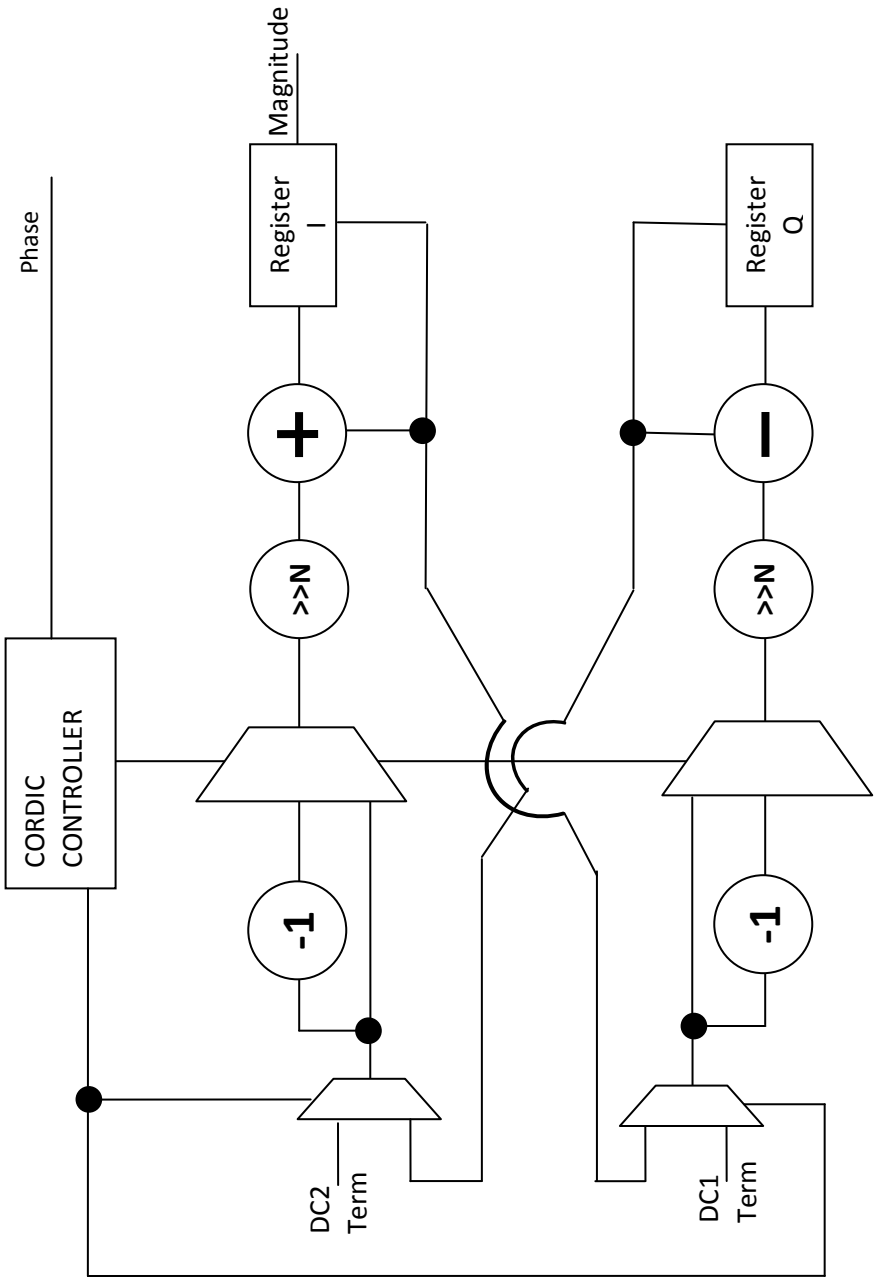

Figure 4-4: Illustration of Circular CORDIC Calculation Unit

This basic model suffers from two drawbacks. In this model of the CORDIC, each iterative step must shift, a variable amount, the values in *Register I* and *Register Q* then add them together. The first hindrance is that this model requires barrel shifters to be able to implement this selective shift length for the shift and add process. An *N*-bit barrel shifter would require *N* flip-flops and *N*, *N*-to-1 multiplexers to implement. This represents a large amount of area overhead to implement for mixed-signal BIST systems which are accumulating large values of DC_1 and DC_2, such as 32 bits and larger. To resolve this issue, the model can be pipelined to break apart the barrel shifting operation into *N*, 1- bit shift operations for each iteration. A simple counter can be used to keep track of the number of shifts.

The second problem with the implementation of a CORDIC is speed of calculation. The CORDIC model requires multiple clock cycles to complete the multiple iterations required to obtain the result. The standard model only requires one clock cycle per iteration, but by pipelining the shifting operation to remove the need for barrel shifters, now multiple clock cycles are needed for each iterative step. This increases the number of clock cycles required to obtain a valid result. However, once the CORDIC has begun calculating a measurement of the magnitude and phase, the data is latched into the circuit and no longer need to be held at the input. This can allow the ORA to capture the measurement for the next frequency while the CORDIC circuit simultaneously extracts magnitude and phase of the previous measurement. In addition, because the model has been pipelined, the maximum allowable clock frequency has been increased allowing the CORDIC to operate at a faster clock rate.

As with all CORDIC models, each iterative step introduces a moderate amount of error into the measurement. This error is fixed, based on the number of iterations the CORDIC takes to calculate the result and only skews the magnitude result. The error in each step is given in Equation 2-29. However, because each of the desired tests of linearity, frequency response, and noise measurement requires at least two measurements to be directly compared to each other, this error factor cancels in each test.

4.1.2 Translation of Magnitude Value to Decibels

The final result of the CORDIC calculation unit returns the magnitude and phase used in each test. However, these results are in an atypical format. Typically, measurements of signal strength difference, ΔP from the linearity test, magnitude response, and SNR are all given in the logarithmic units of decibels, *dB*. The results returned from the CORDIC calculation unit are in a non-logarithmic, linear format for amplitude. To simply evaluate the results of the test, the returned responses need to be in a decibel format.

The conversion between linear and logarithmic can be implemented within a single LUT. However, large values of magnitude will result in very large LUTs. There is a logarithmic CORDIC algorithm, discussed in Chapter 2, designed specifically to implement this conversion. However the area overhead required to construct a logarithmic CORDIC out-shadows its usefulness in smaller applications. To express these units in decibels a modified linear CORDIC can be used in place of the logarithmic CORDIC algorithm. The linear CORDIC consumes less resources than the logarithmic CORDIC allowing it to be efficiently implemented in transformation approximations.

This linear CORDIC block, referred to as the Logarithmic Translation Unit in Figure 4-1, in the test calculation circuit's design will take a linear magnitude value and re-express this value in the form of Equation 4-10.

$$OUTPUT = 20 \cdot \log_{10}(INPUT) \tag{4-10}$$

This CORDIC procedure operates on a simpler shift and subtract principle than the CORDIC Calculation Unit. This modified linear CORDIC operates on the approximation that each binary power of two can be represented in decibel by Equation 4-11.

$$20 \cdot \log_{10}(2^M) = M \cdot 6.0206dB \tag{4-11}$$

By finding the highest order power of two, the logarithmic value can be approximated to within 6.0206dB. However, a range of 6.0206dB represents a large amount of error in the measurement. To scale the error down, a correction factor can be added to the approximation. This is represented in Equation 4-12. This correction factor will utilize the remaining bits below the largest one bit to further improve accuracy. These lower bits can be represented by fractional addition to the current logarithmic approximation. A small Look-Up Table can be used to implement this addition by observing the values on the next four bits below the highest one bit. The values that will be entered into the Look-Up Table are expressed in Equation 4-13.

$$Approximation = 20 \cdot \log_{10}(2^M) + \ Correction\ Factor \tag{4-12}$$

$$LUT\ VALUE(N) = 20 \cdot \log_{10}(1 + \frac{N}{16}) \tag{4-13}$$

To obtain this logarithmic value, the CORDIC will begin by loading the *INPUT* value into one register as it did with the previous CORDIC, while the other register is preloaded with the maximum possible logarithmic value expressible by the *N*-bit *INPUT* value. The largest possible number expressed by a power of two as the *INPUT* register is shown in Equation 4-14, where *N* represents the size in bits of the *INPUT* register. This CORDIC's approach is to shift through each bit of the *INPUT* register from the MSB down to the LSB, until it locates the first bit with the value of one. For each shift operation performed the CORDIC will subtract the value of approximately 6 dB from the second register, which was preloaded with the maximum possible value. Once a bit with the value of one is located, the next four bits in the *INPUT* register are passed through a small 4-bit Look-Up Table. This LUT value is then added to the second register containing the current approximation for the decibel value. The result is a very small logarithmic approximation unit consisting of one subtractor unit, one adder unit, and a 4-bit LUT. The model for the Logarithmic Translation Unit is illustrated in Figure 4-5.

$$Highest\ Possible\ Value = ceil(20 \cdot \log_{10}(2^{N-1}) - 1) \tag{4-14}$$

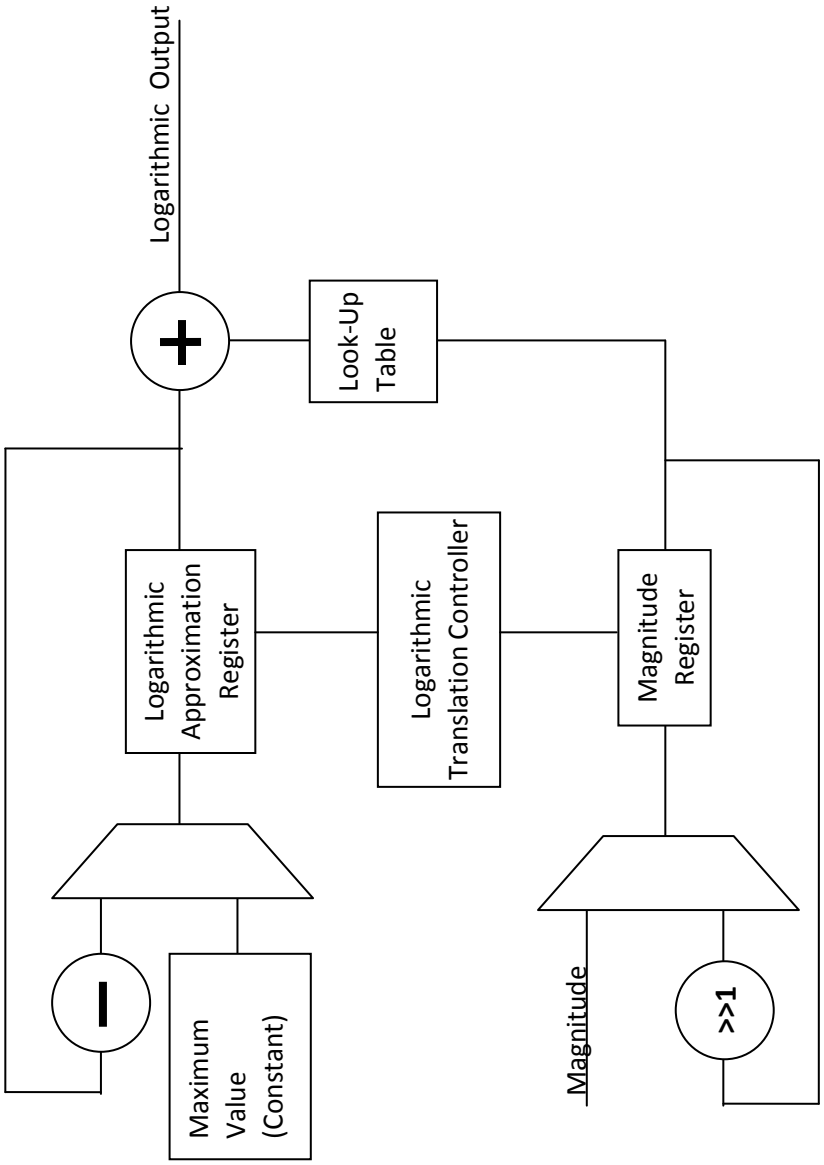

Figure 4-5: Illustration of Logarithmic Translation Unit

4.2 Built-In Self-Test Implementation Test Results

To experimentally verify the operation of this BIST approach using the on-chip calculator, the proposed model was synthesized and programmed onto a Xilinx Spartan 3 S-1000 Field Programmable Gate Array (FPGA). Then each of the three required tests was performed and their results measured. The following section highlights the findings of these tests.

The first test of interest was the frequency response measurement. The results of this measurement are depicted in Figure 4-6. The blue plot represents the measurement of the off-chip calculations, while the red plot represents the on-chip calculation, and black represents the measurements obtained by a spectrum analyzer. This test requires many data points to be measured. By extracting values directly from the output of the CCU, rather than the output of the logarithmic translation unit, a more accurate answer can be obtained for this test to avoid any rounding that occurs in the division process and conversion to logarithmic format.

The results obtained show that the difference between the calculation of the on-chip and off-chip measurements differ by less than one tenth of a decibel in this example. Both of these measurements differ by less than two tenths of a decibel from the actual frequency response recorded by the spectrum analyzer. This shows that the CCU which is generating the magnitude is properly producing magnitude measurements on-chip. A similar comparison between the calculated phase performed off-chip, using the method elaborated upon in Chapter 2, and the phase reported by the CCU can be performed.

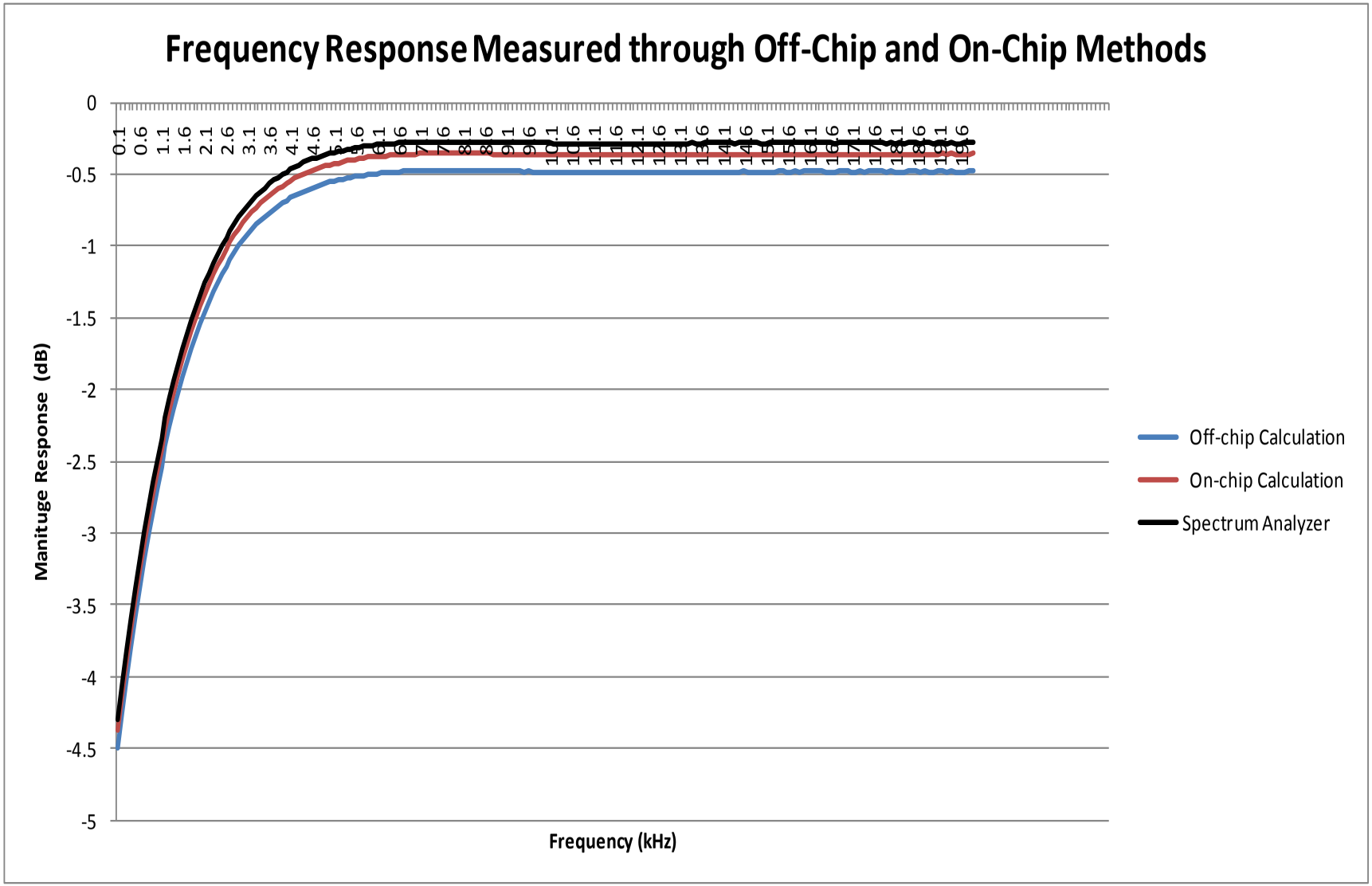

Figure 4-6: Plot of Frequency Response Using On-Chip and Off-Chip Calculations for Magnitude

This second test result illustrates a comparison of the linearity ΔP measurement obtained by the proposed circuit in this chapter plotted against the same test results calculated off-chip using the ORA results. These results are shown in Figure 4-7. This test utilizes the same color palette to express the plots taken as the frequency response measurement. Represented in red is the result of the on-chip calculation circuit, represented in blue is the calculation performed off-chip, and represented in black is the correct measurement as recorded by an external spectrum analyzer. Both plots of on-chip and off-chip calculation follow the same trend line and approximate the correct measurement. At higher linearity measurements, the measured IM3 tones become too weak to properly measure with small ADCs in the mixed-signal BIST model. This results in deviation from the measurement obtained by the spectrum analyzer at high linearity measurements.

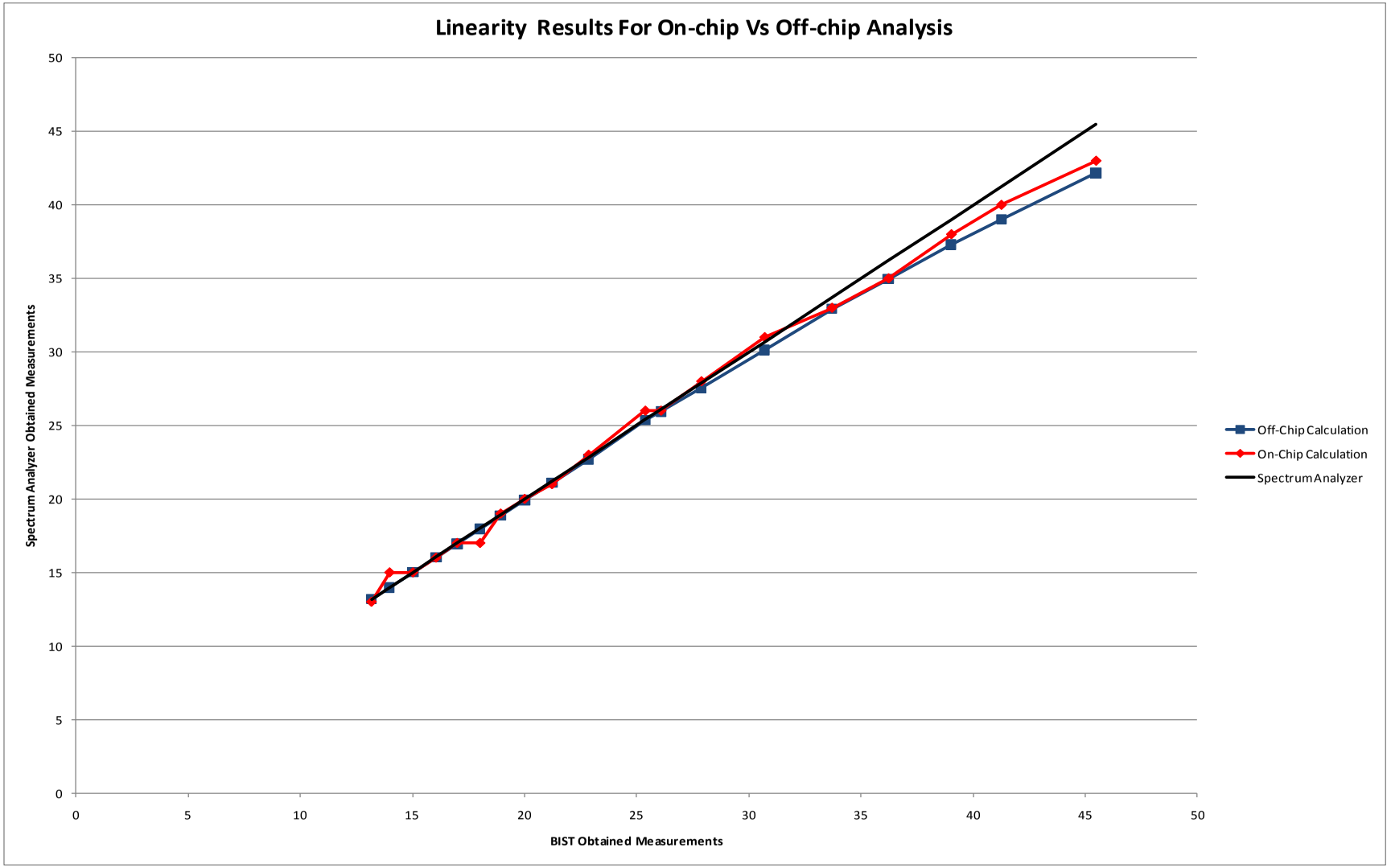

Figure 4-7: Plot of Linearity ΔP Results for On-Chip and Off-Chip Test Calculations

There are two minor differences between these two plots. The first difference occurs in the lower range of measurements from 20dB and below. The results in this range represent a measurement and division by two numbers which are very close in magnitude to each other. This hardware model operates on integer values and cannot perform floating point calculations as the off-chip calculation does; because of this, some information about the signal difference is lost during the division.

Another potential source for error in this range is the translation process into the logarithmic decibels. Because the logarithmic translation circuit's input is an integer number, there are decibel values which cannot be achieved due to rounding. The first non-zero response from the logarithmic translation circuit occurs when the input has a value of two, this equates to an output of 6dB, making values 1-5dB not expressible due to the rounding. However, for larger inputs, the logarithmic translation circuit can display the correct decibel approximation. The seven smallest input values to the Logarithmic Translation Unit are shown in Table 4-2. This table illustrates how smaller inputs have increasingly larger inexpressible range between steps when compared to larger inputs.

Table 4-2: Small Value Measurements on the Input of the Logarithmic Translation Unit

Input Value	Output Value (dB)
1	0 dB
2	6 dB
3	10 dB
4	12 dB
5	14 dB
6	16 dB
7	17 dB

The second difference becomes more noticeable in the higher range, but occurs throughout the plot. Because the output of the logarithmic translation circuit can only be

expressed as integer decibels, the maximum accuracy of the output can only be guaranteed to be within 1dB of the intended measurement. As observed in the on-chip result, each of the outputs is within 1 dB of the off-chip calculation.

The final test of interest represents a measurement of the Signal to Noise Ratio (SNR) of the system. Using this CCU to calculate the magnitude of multiple noise measurements and a simple accumulator and divider to average these noise measurements, the SNR can be obtained. When the measurements obtained by an off-chip calculation using only the ORA results are compared to the measurements obtained by the on-chip calculator, the results appear close. Small deviations are present in the measurements due to rounding and averaging.

Chapter Five

Summary and Conclusions

This thesis discussed a method of testing the performance of analog circuits in mixed-signal integrated circuit using a Built-In Self-Test approach presented in [4]-[12]. This approach utilizes selective spectrum analysis to isolate and capture particular frequency components of a measured analog waveform to be analyzed. To evaluate the performance of an analog device the Built-In Self-Test approach implemented three tests: linearity to evaluate the intermodulation distortion, frequency response to evaluate the relative magnitude and phase shift within the operational range, and noise figure to determine the signal degradation due to the analog circuit.

5.1 Mixed-Signal Built-In Self-Test Improvements

A number of improvements to the mixed-signal Built-In Self-Test design were presented in this thesis. The first improvement to this design was the introduction of a method to automatically generate the hardware description language model, given a set of user definable parameters. These parameters can be used to match system requirements to generate a specific hardware model for a specific implementation. This program allows a developer the ability to selectively trade-off performance for area overhead.

To generate a waveform, the test pattern generator utilizes a direct digital synthesis approach. This requires a numerically controlled oscillator to generate the

waveform from a phase accumulator. This is typically performed with waveforms stored in look-up tables. The improvement to the mixed-signal BIST encodes only one quarter of a complete waveform, as opposed to the entire waveform, to generate the output. This compresses the required area of the look-up table by seventy-five percent as opposed to NCOs which encode the entire table, while adding only minor decoding hardware to reconstruct the full waveform.

Most hardware description languages lack the trigonometric functions necessary for expressing a sinusoidal waveform. This is a necessary component in the generation of the NCO in the TPG. An additional feature of the automatic generation of the mixed-signal BIST is the ability to automatically generate the look-up tables that comprise the NCOs to the size of the model. This program takes advantage of the trigonometric functions built into the standard C programming language libraries and re-expresses the output within a VHDL file format. This implements a customized sinusoidal look-up table in VHDL without needing VHDL operations for trigonometric functions.

A further improvement to the Direct Digital Synthesis based Test Pattern Generator to improve spurious noise was presented. This improvement involved the replacing the traditional accumulators used in the DDS design with a modified Nicholas accumulator. These new accumulators act to dither the accumulated phase used in generating the output waveform. This can reduce the effect of quantized noise, spurious tones, and image tones generated unintentionally by the DDS-based TPG in the mixed-signal BIST. This is performed by spreading the noise of the few strong spurious tones into many small signal noise tones. The program treats this new accumulator as an

optional replacement to the model allowing the developer to decide if it should be generated within the design.

Additional improvements to the mixed-signal BIST approach were made in the selective spectrum analysis based output response analyzer. Because the ORA consists only two pairs of multipliers and accumulators, there is little to be improved upon. However, for this implementation of the mixed-signal BIST, a new multiplier was introduced which improves in the speed of calculation. This multiplier was based on the modified Booth multiplier architecture, which reduces the number of partial products in half. This was further optimized for speed and area by structuring the partial product addition stages using the Wallace-tree reduction technique. Then a new approach for reducing the sign extension bits of the partial products was implemented into the multiplier design. Finally, this multiplier was pipelined to greatly improve the speed of the calculation.

5.2 Mixed-Signal Built-In Self-Test Additions

The originally proposed mixed-signal BIST model is capable only of extracting the in-phase and out-of-phase components of a measured signal. These components are necessary for evaluating on-chip performance; however the format these components are in is not conducive for calculating the on-chip measurements. To convert these two components into the magnitude and phase of the measurement, a new addition to the mixed-signal BIST model was employed.

Utilizing the Coordinate Rotation Digital Computing algorithm, a new component to the model was constructed which allowed the mixed-signal BIST to calculate the

components of magnitude and phase from the ORA output. This model utilized an iterative successive approximation approach to obtain the accumulated magnitude of the signal of interest. In the processes of obtaining the magnitude, the iterative rotational aspect of this CORDIC method also obtains the phase. When combined with additional circuitry, the output of this unit allows for the analog measurements and tests to be evaluated on-chip. Utilizing additional features of the CORDIC algorithm, a logarithmic translation circuit was constructed to convert the linear amplitude measurement obtained by the calculation CORDIC into a logarithmic decibel format. This simplifies the final result and facilitates on-chip measurement.

5.3 Areas for Future Research and Development

It is recommended that improvements be made to reduce the overall area of the test calculation circuit. When compared to the area of the ORA, the current test calculation circuit represents a significant increase in area. Further improvements should be made to CORDIC Calculation Unit to increase the calculation speed. At present, the CCU represents the longest register to register delay in the entire mixed-signal BIST model. Improving this component in the design will increase the maximum operating speed of the entire mixed-signal model.

After a reduced test calculation circuit has been finalized, this new entity should be integrated into the automatic mixed-signal BIST generation program along with any user-configurable options associated with the entity. Additional improvements can be made to the automated BIST generation program. Currently, the automated mixed-signal BIST generation program generates a communication interface length specific to the

implementation of that particular model. This forces the designer using the generator to develop a custom communication interface independently. An improvement to the BIST generator would be to export information about the contents of the communication interface to aid in the development of a communication interface.

Bibliography

[1] R. C. Jaeger, *Introduction to Microelectronic Fabrication.* Upper Saddle River: Prentice Hall, 1998.

[2] C. H. Stapper, "LSI Yield Modeling and Process Monitoring," IBM J. Res. Develop. vol. 20, no. 3, pp. 228-234, 1976.

[3] C. E. Stroud, *A Designer's Guide to Built-In Self-Test*. Boston: Springer, 2002.

[4] J. Qin, C. E. Stroud and F. F. Dai, "FPGA-Based Analog Functional Measurements for Adaptive Control in Mixed-Signal Systems," IEEE Trans. on Industrial Electronics, vol. 54, no. 4, pp. 1885-1897, 2007.

[5] J. Qin, C. E. Stroud and F. F. Dai, "Phase Delay Measurement and Calibration in Built-In Analog Functional Testing," Proc. IEEE Southeastern Symp. on System Theory, pp. 145-149, 2007.

[6] F. Dai, C. Stroud, C. Yang and S. Qi, "Automatic Linearity (IP3) Test with Built-In Pattern Generator and Analyzer," Proc. IEEE International Test Conf., pp. 271-280, 2004.

[7] F. F. Dai, C. E. Stroud, and D. Yang, "Automatic Linearity and Frequency Response Tests With Built-In Pattern Generator and Analyzer," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 14, no. 6, pp. 561-572, 2006.

[8] D. Yang, C. E. Stroud and F. F. Dai, "Built-In Self-Test for Automatic Analog Frequency Response Measurement". Proc. IEEE International Symp. on Circuits and Systems, pp. 2208-2211, 2005.

[9] J. Qin, C. E. Stroud, and F. F. Dai, "Noise Figure Measurement Using Mixed-Signal BIST," Proc. IEEE International Symp. on Circuits and Systems, pp. 2180-2183, 2007.

[10] C. E. Stroud, D. Yang and F. F. Dai, "Analog Frequency Response Measurement in Mixed-Signal Systems," Proc. IEEE International Symp. on Circuits and Systems, pp. 5736-5739, 2006.

[11] J. Qin, C. E. Stroud and F. F. Dai, "Test Time of Multiplier/Accumulator Based Output Response Analyzer in Built-In Analog Functional Testing," Proc. IEEE Southeastern Symp. on System Theory, pp. 363-367, 2009.

[12] J. Qin, C. E. Stroud, and F. F. Dai, "Phase Delay in MAC-Based Analog Functional Test in Mixed-Signal Systems," Proc. IEEE North Atlantic Test Workshop, pp. 44-50, 2006.

[13] S. Qi, Foster Dai, C. E. Stroud, and D. Yang, "Analog Circuit Testing Using Built-In Direct-Digital Synthesis", Proc. IEEE North Atlantic Test Workshop, pp. 190-195, 2004.

[14] G. J. Starr, J. Qin, B. F. Dutton, C. E. Stroud, F. F. Dai and V. P. Nelson, "Automated Generation of Built-In Self-Test and Measurement Circuitry for Mixed-Signal Circuits and Systems," Proc. IEEE International Symp. on Defect and Fault Tolerance in VLSI Systems, pp. 11-19, 2009.

[15] C. E. Stroud, J. R. Morton and A. Islam, "An Automated BIST Approach for Mixed-Signal Systems," Proc. IEEE North Atlantic Test Workshop, pp. 202-209, 2004.

[16] D. K. Pradhan and I. G. Harris, *Practical Design Verification*. New York: Cambridge, 2009.

[17] J. Vankka and K. Halonen, *Direct Digital Synthesizers: Theory, Design and Applications*. Boston: Kluwer, 2001.

[18] E. C. Chu, *Inside the FFT Black Box: Serial and Parallel Fast Fourier Transform Algorithms*. Boca Raton: CRC Press, 2000.

[19] W. L. Briggs, *DFT: an Owner's Manual for the Discrete Fourier Transform*. Philadelphia: SIAM, 1995.

[20] B. Razavi, *RF Microelectronics*, Upper Saddle River: Prentice Hall, 1998.

[21] J. C. Pedro and N. B. Carvalho, "On the Use of Multitone Techniques for Assessing RF Components' Intermodulation Distortion," IEEE Trans. On Microwave Theory & Techniques, vol. 47, no. 12, pp. 2393-2402, 1999.

[22] J. Mielke, "Frequency-Domain Testing of ADCs," IEEE Design & Test of Computers, vol. 13, no. 1, pp. 64-69, 1996.

[23] M. Soma, S. Huynh, J. Zhang, S. Kim, and G. Devarayanadurg, "Hierarchical ATPG for Analog Circuits and Systems," IEEE Design & Test of Computers, vol. 18, no. 1, pp. 72-81, 2001.

[24] S. D. Huynh, K. Seongwon, M. Soma, and Z. Jinyan, "Testability Analysis and Multi-Frequency ATPG for Analog Circuits and Systems," Proc. International Conf. on Computer-Aided Design, pp. 376-383, 1998.

[25] P. Variyam and A. Chatterjee, "Test Generation for Comprehensive Testing of Linear Analog Circuits Using Transient Response Sampling," Proc. IEEE VLSI Test Symp., pp. 60-65, 1994.

[26] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.

[27] R. C. Jaeger, Microelectronic Circuit Design. Boston: McGraw-Hill, 2008.

[28] E. E. Swartzlander, *Application Specific Processors*. Boston: Kluwer, 1997.

[29] J. M. Muller, *Elementary Functions: Algorithms and Implementation*. Boston: Birkhäuser, 1997.

[30] J. Qin, *Selective Spectrum Analysis (SSA) and Numerically Controlled Oscillator (NCO) in Mixed-Signal Built-In Self-Test*, Doctoral Dissertation, Auburn University, 2010.