2002

Error Compensation in Pipeline and Converters

Kannan Sockalingam

ERROR COMPENSATION IN PIPELINE A/D CONVERTERS

By

Kannan Sockalingam

B.S. University of Maine, 2000

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

(in Electrical Engineering)

The Graduate School

The University of Maine

August, 2002

Advisory Committee:

Donald M. Hummels, Professor of Electrical and Computer Engineering, Advisor

Fred H. Irons, Castle Professor of Electrical and Computer Engineering

Ioannis Papantonopoulos, Wireless Infrastructure, Texas Instruments Inc.

ERROR COMPENSATION IN PIPELINE A/D CONVERTERS

By Kannan Sockalingam

Thesis Advisor: Dr. Donald M. Hummels

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Science
(in Electrical Engineering)
August, 2002

This thesis provides an improved calibration and compensation scheme for pipeline Analog-to-Digital Converters (ADCs). This new scheme utilizes the intermediate stage outputs in a pipeline to characterize error mechanisms in the architecture. The goal of this compensation scheme is to increase the dynamic range of the ADC.

The pipeline architecture is described in general, and tailored to the 1.5 bits/stage topology. Dominant error mechanisms are defined and characterized for an arbitrary stage in the pipeline. These error mechanisms are modeled with basis functions. The traditional calibration scheme is modified and used to iteratively calculate the error characteristics. The information from calibration is used to compensate the ADC.

The calibration and compensation scheme is demonstrated both in simulation and using a custom hardware pipeline ADC. A 10-bit 5 MHz ADC was designed and fabricated in 0.5 μm CMOS to serve as the demonstration platform. The scheme was successful in showing improvements in dynamic range while using intermediate stage outputs to efficiently model errors in a pipeline stage. An application of the technique on the real converter showed an average of 8.6 dB improvement in SFDR in the full Nyquist band of the ADC. The average improvement in SINAD and ENOB are 3.2 dB and 0.53 bits respectively.

ACKNOWLEDGMENTS

The work described in this thesis has been supported in part through a gift from the Texas Instruments Inc. Analog University Relations program. The department gratefully acknowledges the support of its graduate and undergraduate programs provided by Texas Instruments Inc. through this program.

I would like to first thank my parents, Dr. and Mrs. Sockalingam for providing the opportunity to pursue my education here at the University Of Maine. And also to my brothers for their support throughout this educational endeavor. I would especially like to thank Rathna for being patient during the development of this thesis.

Thank you, Don M. Hummels for being patient and providing the support and guidance throughout my bachelors and masters degrees. Thank you, Fred H. Irons for introducing me to data converters, Texas Instruments and fishing. Thank you, Ioannis Papantonopoulos for providing the hands-on experience at TI and also for being patient and forgiving throughout the development of my research. Thank you, Rick Thibodeau for hanging in there (with me) in the VLSI challenge and the Kenduskeag. Thank you, Scott G. Saucier for being a great tutor, colleague and friend. Thank you, Ron Bryant for the valuable pointers and advice on pipeline ADCs. And last but not least, thank you Tim Monk for helping with the hardware testing and data manipulation.

In memory of my uncle, Chidambaram Sethu...

TABLE OF CONTENTS

ACKNOWLEDGMENTS ii

LIST OF TABLES v

LIST OF FIGURES vi

Chapter

1 Introduction 1

1.1 Background 1

1.2 Purpose of the Research 2

1.3 Thesis Organization 3

2 Pipelined ADC 5

2.1 General Architecture 5

2.2 Two-Stage Pipeline Development 6

2.3 Multiple Stage Pipeline Construction 10

2.4 1.5 Bits/stage Architecture 12

2.5 Error Contributions in Pipeline ADCs 14

2.5.1 Sample and Hold Error 15

2.5.2 Sub-ADC Error 16

2.5.3 Sub-DAC Error 17

2.5.4 Gain Error 19

2.6 Measuring Distortion 20

2.7 Simulated Pipeline ADC 21

3 Calibration and Compensation of Pipeline ADCs 31

3.1 Proposed Compensation Architecture 31

3.2 Calibration Scheme 34

3.3 Basis Functions 38

3.3.1 S/H Error Basis Functions 38
3.3.2 Intermediate Bits Basis Functions 39
3.4 Calibration Algorithm 41
3.5 Implementation on Simulated ADC 42
4 Hardware Implementation 48
4.1 A 10-bit 5 MHz Pipeline ADC 48
4.2 Design Overview 51
4.2.1 Clock Description 51
4.2.2 Pipeline Stages 52
4.3 Performance 55
4.4 Calibration and Compensation 56
5 Conclusion 65
REFERENCES 67
BIOGRAPHY OF THE AUTHOR 69

LIST OF TABLES

2.1 Parameters for simulated 10-bit Pipeline ADC (Units in Volts). 27

LIST OF FIGURES

2.1 Generic Pipeline ADC block diagram. 7

2.2 Generic stage block diagram. 7

2.3 An M-bit ADC. 8

2.4 An N-bit ADC connected to an M-bit ADC. 8

2.5 Digital correction for an $(N + M - 1)$-bit ADC. 10

2.6 Different pipeline combinations. 11

2.7 Block diagram of a single stage (1.5 bits/stage). 13

2.8 Sub-DAC outputs in the 1.5 bits/stage architecture. 13

2.9 Example of digital correction in a 1.5 bits/stage pipeline converter. 14

2.10 Residual characteristics for 1.5 bits/stage. 17

2.11 Error characteristics for a single stage with nominal thresholds. 18

2.12 Error characteristics for a single stage with actual thresholds. 18

2.13 Spectral plot of THS1240 ADC with $F_t = 15.5$ MHz. 22

2.14 Simulated 10-bit Pipeline (Ideal) with $F_t = 1$ MHz. 23

2.15 Time varying S/H input signal. 24

2.16 Model of S/H circuit. 25

2.17 Special case in S/H circuit. 25

2.18 Locations of parameters for simulated ADC. 27

2.19 Simulated 10-bit Pipeline (Non-ideal) with $F_t = 1$ MHz. 28

2.20 Simulated ADC output spectra showing error dominance of early stages in a pipeline. 30

3.1 First and second stage block diagram with error models. 33

3.2 Calibration trajectories in the code/slope space. 42

3.3 SFDR performance for Simulated ADC using proposed scheme. 46

3.4 Simulated ADC output spectra illustrating different compensation schemes. 47

4.1 Digital image of the final packaged die. 49

4.2 Floorplan of the final packaged die. 50

4.3 2-Phase Non-Overlapping Clock Timing. 52

4.4 Single stage block diagram in 10-bit ADC. 53

4.5 Differential Comparator implementation. 53

4.6 S/H and gain portions of the stage. 54

4.7 Sub-DAC circuit implementation. 55

4.8 INL and DNL plots for real ADC. 57

4.9 Output spectral plot of ADC for $F_t = 1.2$ MHz. 58

4.10 Sampling capacitor status for different clock phases showing previous sample dependence. 59

4.11 Uncompensated and compensated output spectrum for a single test signal using a 10-bit ADC. 61

4.12 Compensated output spectrum for 10-bit ADC using different basis function sets. 62

4.13 SFDR performance across the Nyquist band for a 10-bit ADC. 63

4.14 SINAD performance across the Nyquist band for a 10-bit ADC. 64

CHAPTER 1

Introduction

1.1 Background

Analog to Digital Converters (ADCs) are widely used electronic components which often limit the performance of modern signal processing systems. There has been a significant effort to improve device performance, particularly for components used in wireless applications. Performance gains can be realized by at least three different methods. Large gains are possible by improving device characteristics. This can be realized by using advanced processes that define today's evolving electronics manufacturing technology. A second method is by improving circuit design technologies or ADC architectures. Advances and changes in circuit design may be able to provide the lacking dynamic range of current ADCs. A final method is by calibrating and compensating existing devices. Using current devices with compensation provides additional gains for existing technology. ADC compensation is especially beneficial for wide-bandwidth receivers that require a large dynamic range of operation [1].

A calibration and compensation scheme is known as dynamic compensation if the dynamic nature of the input signal is accounted for in the compensation procedure. These types of compensation schemes have been successful in providing increased spurious free dynamic range (SFDR) in current ADCs [2, 3, 4, 5]. Dynamic compensation requires the ADC to be calibrated in order to obtain representations of the errors involved. Information about the error functions is then used to compensate the ADC by removing the error from the ADC output samples. A variety of calibration methods have been investigated [3, 6, 7, 8].

A pioneering concept to dynamic ADC compensation was the phase-plane approach [2]. This approach indicated that many amplitudes and test frequencies can be used

to excite all possible output codes in an ADC in order to calibrate it. This concept was implemented in hardware using an improved calibration scheme with sinusoidal test signals [3]. Sinusoidal calibration schemes sometimes require 50 - 100 calibration signals to provide a good representation of the device error functions. Large sets of data allow a histogram method to be used in effectively modeling all the error characteristics [6]. This scheme was not time efficient due to the large number of calibration signals needed. An alternative to many calibration signals was investigated using theoretical non-iterative mathematical models to efficiently model the error mechanisms [5]. The non-iterative approach relied on a reduced number of sinusoidal test signals. However a totally different approach to calibration without sinusoidal test signals, is using pseudo-random signals generated specifically for calibration [9]. Finally, time being an important factor in calibration, multi-tone calibration signals can be used to reduce calibration time and eliminating the need for expensive test equipment [7, 8].

Acunto and Arpaia extended the histogram [6] method to calibrate pipeline architecture ADCs [10]. The method was successful but not very effective on improving the performance of a pipelined ADC. This thesis will apply the non-iterative method to the pipeline ADC but with unique basis functions to model architecture specific error characteristics.

1.2 Purpose of the Research

This thesis will concentrate on pipeline architecture ADCs, which have become the architecture of choice for high speed and moderate to high resolution devices. These devices are particularly important in wireless receiver applications because pipeline ADCs have moderate resolution, high speed and consume low power. However the dynamic range of operation is often limited in current pipeline designs. A compensation scheme unique to pipeline converters will facilitate smaller and more efficient receiver designs because lower resolution ADCs with large dynamic ranges can be used.

Previous efforts in pipeline calibration and compensation have succeeded in providing promising results. Acunto [11] investigated histogram-based compensation schemes based on ADC output code and slope, and demonstrated his results on pipeline ADCs. Although the calibration technique was successful, significant gains in the dynamic range were only realized in the 2^{nd} and 3^{rd} Nyquist bands. In these bands, speed limitations dominate the distortion characteristics of the ADC, and distortion which is related to the pipeline architecture may be masked. This thesis attempts to provide a compensation scheme that will work on pipeline ADCs, and fully define the dominant error mechanisms that exists in these types of architectures.

It is well known that the design specifications are tightest for the first few stages in a pipeline ADC. The error introduced in pipeline ADCs is often dominated by the first few stages within the device [12]. However since the DAC and interstage gain nonlinearities cannot be characterized using the final ADC resolution, the intermediate stage outputs are needed to effectively predict the error mechanisms present in each stage. The proposal of this thesis is to use the intermediate stage output bits in the compensation scheme. This thesis focuses on defining these errors and demonstrate that these errors can be compensated using intermediate stage outputs (if available). Apart from the DAC and gain nonlinearities, slope dependent error is also included in the compensation scheme [2]. This work also defines a calibration scheme unique to pipeline ADCs.

1.3 Thesis Organization

This thesis is structured to provide some background followed by theory, simulation and results obtained from a hardware implementation.

Chapter 2 gives an introduction to the concept of pipelining in general. The general case will be tailored to the 1.5 bits/stage architecture used in the hardware implementation. This chapter also includes a fully simulated pipeline ADC based on

this implementation. The simulation allows for evaluation of the proposed calibration and compensation procedure for an ADC with known error characteristics.

Chapter 3 introduces the calibration and compensation schemes and the theory behind these concepts. This chapter describes in detail a method of calibration and compensation for pipeline converters using the intermediate stage outputs. Sections in this chapter also outline the mathematical justification behind the proposed compensation scheme. A simulation of the ADC using the compensation scheme is also included in this chapter.

Chapter 4 describes the design of a 10-bit 5 MHz pipeline ADC used to test the compensation scheme in this thesis. Details of the design and layout of the converter are also discussed. The ADC is calibrated and compensated using the proposed architecture and the results from compensation are discussed in detail.

Chapter 5 summarizes this thesis by discussing the outcome of the real ADC test compared to the simulated results. A brief section on possible future work is also included.

CHAPTER 2

Pipelined ADC

This chapter provides an overview of the pipeline ADC architecture. The pipelined structure is developed and discussed. Dominant error distortion mechanisms for pipeline ADCs are identified and a detailed simulation of a 1.5 bits/stage pipeline ADC is developed to illustrate the effect of various distortion mechanisms.

2.1 General Architecture

ADCs are designed using many architectures. The chosen architecture may depend on the output resolution (number of bits), speed (sampling rate) or power requirements. The pipeline architecture is optimized for high speed conversion and offers lower power consumption then comparable speed devices. Typically the pipeline concept is used for fairly low resolution but high speed conversion. This thesis focuses on ADCs designed using pipeline architectures.

A generic pipeline converter is shown in Figure 2.1. Each 'stage' in the ADC includes a low resolution quantizer. The i^{th} stage provides 2 outputs. The first output, q_i, is a coarse resolution digital representation of its input voltage. q_i is an integer value that can range from 0 to $2^{B_i} - 1$, where B_i is the bit resolution of the i^{th} stage. The second output is a residual voltage, r_i, obtained by measuring the difference between the input and the voltage predicted by q_i. This residual voltage is passed onto the next stage in the pipeline and q_i is sent to the digital correction logic. Subsequent stages attempt to improve the final representation by quantizing the residual voltage. All the q_i's are collected in 'digital correction' which is used to combine these coarse estimated values into a final higher resolution representation of X.

All stages are clocked by the same clock. Once Stage 1 produces r_1 and q_1, Stage 2 begins quantizing r_1 while Stage 1 is processing the next input sample. This

continuous processing of samples is the concept of 'pipelining'. The efficiency of a pipeline architecture makes it a good candidate for high speed conversions. Only low resolution stages are needed to obtain a higher resolution output. An example of an industry pipeline ADC is the Texas Instruments THS1240 (12-bit 40MHz) [13].

Figure 2.2 shows a block diagram of a typical stage. A Sample and Hold (S/H) is on the input of each stage block since conversion must take place while the previous stage is processing the next sample. Each stage has a B_i-bit ADC (or sub-ADC) to provide the digital output for that particular stage. A DAC (sub-DAC) with similar resolution is used to convert this digital output to an analog voltage. This voltage is subtracted from the initial input sample giving the error voltage, e_i. The resulting residual error, e_i, is scaled by a gain factor and sent to the following stage as r_i. The gain typically has a value of 2^{B_i-1} which depends on the stage resolution. The scale factor is used to scale the residual to the full operating range of the next stage. Selecting the gain as a power of two greatly simplifies the digital correction logic.

Breaking down the high resolution conversion into stages with low resolution has advantages and disadvantages. An important advantage is a high conversion rate. Using stages also saves real-estate on silicon (this will become more evident in the later parts of this chapter). If space is not an issue, more stages can be added to obtain higher resolution. The following section describes in detail the theory of a pipeline converter using M-bit and N-bit converters.

2.2 Two-Stage Pipeline Development

A pipeline ADC can be constructed using low resolution ADCs to represent stages in a high resolution converter. Consider an M-bit ADC that only includes the S/H, sub-ADC, sub-DAC and subtracter as shown in Figure 2.2. Figure 2.3 shows the M-bit converter with an input X and outputs q and e. The input range is assumed to be limited to $|X| \leq V_R$, so that the quantization interval for this M-bit ADC is $Q_M =$

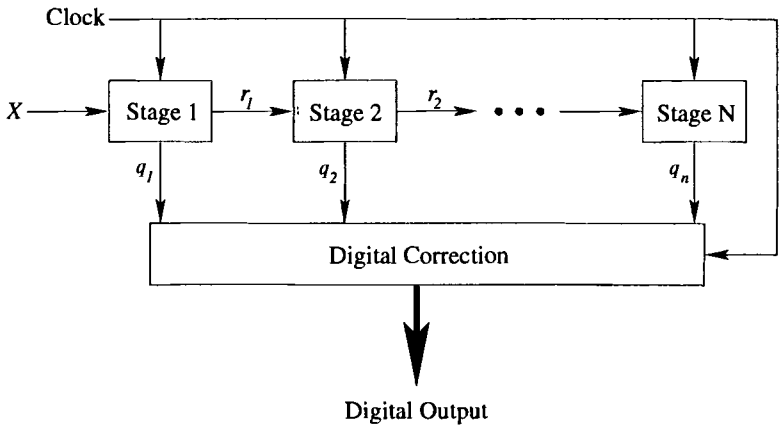


Figure 2.1: Generic Pipeline ADC block diagram.

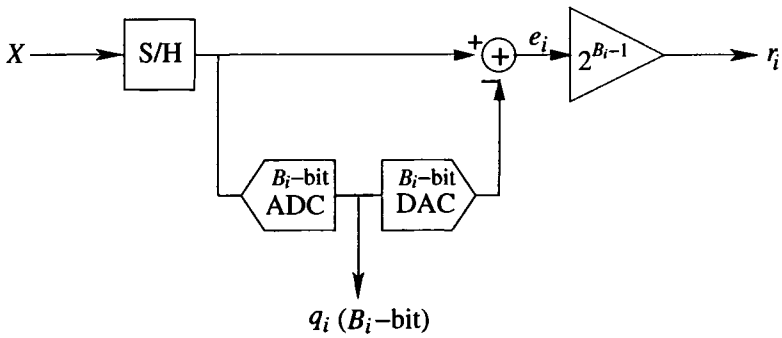


Figure 2.2: Generic stage block diagram.

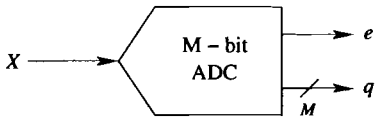


Figure 2.3: An M-bit ADC.

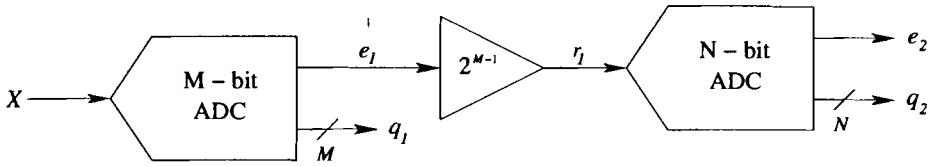


Figure 2.4: An N-bit ADC connected to an M-bit ADC.

$2V_R/2^M$. The converter output q is an integer ranging from 0 to $2^M - 1$. This integer value represents an M-bit digital representation of the input. For a mid-tread ADC, the average input voltage which could produce output q is given by $Q_M(q - (2^{M-1} - 1))$. Let e be the residual error between this estimate and the input voltage X.

$$X = \left(q - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right) + e, \text{ where } |X| < V_R. \tag{2.1}$$

Since the M-bit converter is of a mid-tread design the residual error e can be kept small. For an ideal ADC, the error is bounded by $|e| \leq Q/2$ (except the saturation points of the input range). Due to design and process limitations, in practice the error will exceed these bounds, and a more realistic assumption is that $|e| \leq Q$. If e is available, a second ADC can be used to improve the output by quantizing this error. An N-bit converter is connected at the output of the M-bit converter as in Figure 2.4. Notice that a gain factor is selected to make $|r_1| < V_R$ (assuming $|e| \leq Q$). Equation 2.1 can be written for the N-bit converter,

$$r_1 = \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^N}\right) + e_2, \text{ where } |e_2| < Q_N = \frac{2V_R}{2^N}. \tag{2.2}$$

However since $r_1 = 2^{M-1}e_1$ then

$$e_1 = \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^{N+M-1}}\right) + \frac{e_2}{2^{M-1}}. \tag{2.3}$$

The M-bit converter output (Equation 2.1) may be rewritten as

$$\begin{aligned} X &= \left(q_1 - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right) + e_1 \\ &= \left(q_1 2^{N-1} - \left(2^{N+M-2} - 2^{N-1}\right)\right)\left(\frac{2V_R}{2^{N+M-1}}\right) + e_1. \end{aligned} \tag{2.4}$$

By combining Equations 2.3 and 2.4, an expression for the input X and the quantized outputs can be obtained as

$$X = \left(q_1 2^{N-1} + q_2 - \left(2^{N+M-2} - 1\right)\right)\left(\frac{2V_R}{2^{M+N-1}}\right) + \frac{e_2}{2^{M-1}}. \tag{2.5}$$

This new expression for X is essentially Equation 2.1 written for a single $(N+M-1)$-bit ADC. By combining the two ADCs, a higher resolution estimate of the input is obtained. The desired digital output from Expression 2.5 is given by $q = q_1 2^{N-1} + q_2$. The output q calculation is carried out in the digital correction portion of the architecture. The quantization interval for the combination converter is, $Q_{N+M-1} = 2V_R/2^{N+M-1}$. Since $|e_2| \leq 2V_R/2^M$, the maximum error for the combined result is bounded by $|\frac{e_2}{2^{M-1}}| \leq \frac{2V_R}{2^N 2^{M-1}} = Q_{N+M-1}$.

The purpose of the digital correction is to calculate the final q for each sample by adding up all the intermediate stage outputs, q_i. From the M and N-bit example, the final q is given by $q_1 2^{N-1} + q_2$. The 2^{N-1} is a binary shift to align the bits before carrying out a binary addition. The outputs are shifted according to the resolution each stage contributes to a final estimate. Assuming q_1 and q_2 are broken down into binary bits, Figure 2.5 shows the alignment of these bits before the addition operation. Note that

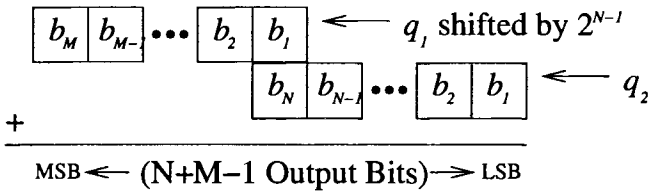


Figure 2.5: Digital correction for an $(N + M - 1)$-bit ADC.

in this figure, b_1 is the Least Significant Bit (LSB) and b_M or b_N is the Most Significant Bit (MSB) of the corresponding outputs.

Operation of the converter for input voltages at or above V_R requires special consideration. Close examination of the error correction scheme shows that incorrect results are generated if both $q_1 = 2^M - 1$ and $q_2 = 2^N - 1$. This converter state generates a numerical overflow and will not produce a valid ADC output code. To solve this problem, the range of valid outputs for the M-bit converter is usually limited. Valid outputs range from 0 to $2^M - 2$, so that this device never produces the problematic output code $2^M - 1$. Since a mid-tread design is being used, discarding this output code will not degrade performance. The residual r_i will still satisfy the bound $|r_i| < V_R$ provided the input is limited by $|X| < V_R$. This will ensure that the sample is estimated correctly and no information is lost. Only the last stage of the pipeline is allowed to output its saturation state $q_2 = 2^N - 1$. This configuration corrects the saturation behavior of the converter, and actually reduces the hardware required to implement all intermediate pipeline stages.

2.3 Multiple Stage Pipeline Construction

The number of stages in a pipeline varies. In the previous example, the M and N-bit converters can be considered as 2 stages in an $(N+M-1)$-bit ADC. As mentioned in Section 2.1, cascading smaller resolution stages can save real-estate in silicon. As the resolution of a stage decreases, the number of comparators needed for the entire system

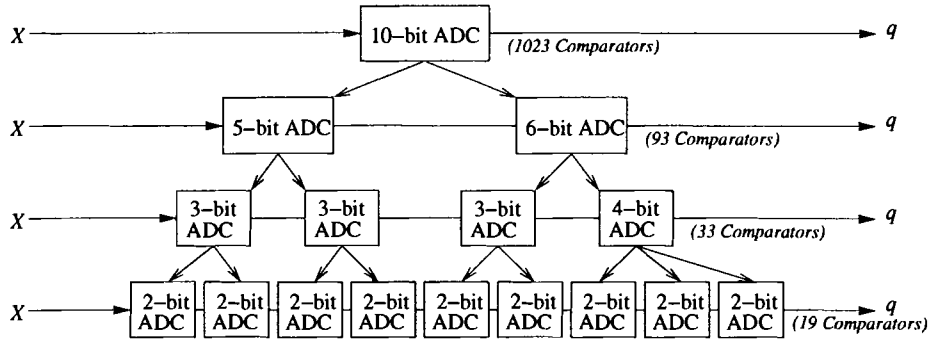


Figure 2.6: Different pipeline combinations.

also decreases. An M-bit stage can have up to $2^M - 1$ comparators to obtain M-bits of resolution.

For example, Figure 2.6 shows four different possible implementations of a 10-bit converter. As discussed in Section 2.2, a 10-bit converter could be split into two stages using pipelined 5 and 6-bit stages. A flash 10-bit ADC requires $2^{10} - 1 = 1023$ comparators. The 5-bit stage would require $2^5 - 2 = 30$ comparators and the 6-bit converter would have $2^6 - 1 = 63$ comparators for a total of 93 comparators for this two stage implementation. The process can be continued by using pipeline architectures for the 5 and 6-bit stages. The 5-bit stage can be reduced to a combination of two 3-bit stages and similarly the 6-bit can be reduced to a 3-bit and 4-bit stage. The total number of comparators for this 4 stage implementation is $(2^3 - 2) + (2^3 - 2) + (2^3 - 2) + (2^4 - 1) = 33$ comparators. Notice the number of comparators is drastically reduced by incorporating many lower resolution stages. These 4 stages can be further broken down into combinations of 2-bit stages. This architecture cascades nine 2-bit stages. The implementation requires only $8(2^2 - 2) + (2^2 - 1) = 19$ comparators. Each 2-bit stage (except the last) only implements 3 quantization intervals; this topology is known as the '1.5 bits/stage' architecture in literature. The following section describes the 1.5 bits/stage architecture.

2.4 1.5 Bits/stage Architecture

Using 1.5 bits/stage relaxes the requirements on the sub-ADC, so efficient high speed designs can be developed. The 1.5 bits/stage can only have three valid quantization intervals, '00', '01'and '10'. In order to prevent overflow in the digital correction, '11' is not a valid output unless from the last stage. To get M-bits of resolution, $M-1$ stages are needed. Using this unique topology, the gain factor can be set to a constant of 2 between stages.

Figure 2.7 shows a typical structure for a single stage of the 1.5 bits/stage topology. Assuming the input has a range $|X| \leq V_R$, the quantization interval size is $V_R/2$. The sub-ADC should have thresholds of $V_R/4$ and $-V_R/4$. In practice, the actual ADC thresholds will be different from these 'nominal' values. The ADC thresholds are illustrated in Figure 2.8. Figure 2.8 also includes the corresponding digital outputs of the sub-ADC and the analog voltages which are produced by the sub-DAC. The digital word is sent to the digital correction portion of the ADC to be combined with all the other stage outputs. The digital output word provides the input to the sub-DAC. This sub-DAC is designed to produce one of three possible output voltages. For the 1.5 bits/stage architecture these voltages are set to $V_R/2$, 0, and $-V_R/2$ for sub-ADC outputs codes 00, 01 and 10 respectively. The sub-DAC output is subtracted from the input and the residual is multiplied by a gain of 2. If the sub-DAC outputs are accurate, this gain will scale the residual error to $\pm V_R$ (provided the sub-ADC quantization thresholds are within $V_R/4$ of their nominal values). The following stage will then quantize the residual error from this stage.

Digital correction for a 1.5 bits/stage pipeline is fairly simple. Conceptually, the first stage in a pipeline provides the MSB for the final output and as the sample moves through the pipeline, each stage output become less significant and the last stage provides the LSB. Figure 2.9 shows how the bits from each stage are added together [14]. All stages shown in this figure are assumed to have a 1.5 bits/stage architecture.

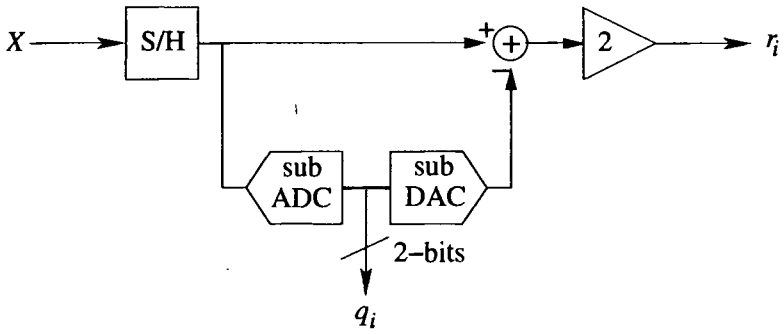


Figure 2.7: Block diagram of a single stage (1.5 bits/stage).

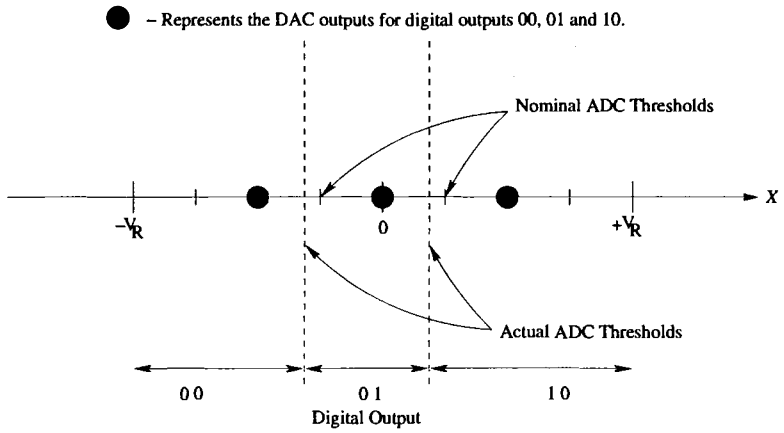


Figure 2.8: Sub-DAC outputs in the 1.5 bits/stage architecture.

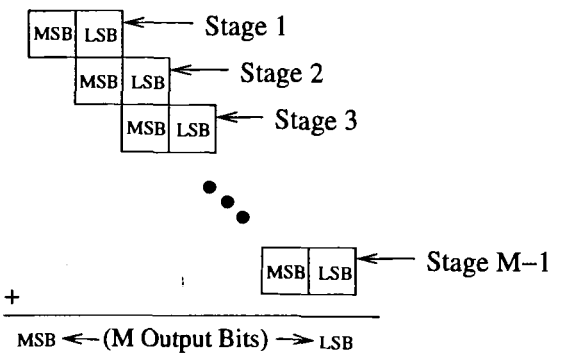


Figure 2.9: Example of digital correction in a 1.5 bits/stage pipeline converter.

Note that the data from preceding stages need to be maintained in memory until the last set of outputs is available for the final conversion to be calculated.

2.5 Error Contributions in Pipeline ADCs

Previous sections discussed the operation of a pipeline ADC. This section addresses possible error mechanisms that can be present in a pipeline converter. Extensive work has been done on characterizing error mechanisms and an overview can be found in [12]. The goal here is to provide a functional description of how errors in the various blocks shown in Figure 2.2 affect the output sample. The following paragraphs discuss error in the S/H, sub-ADC, sub-DAC and gain stage. Previous sections looked at the pipeline structure using M and N-bit ADCs; this same concept can be used to analyze any section in a pipeline. For example if there are three 4-bit stages in a 10-bit ADC, the analysis between the second and third (last) stage can be represented using the same concept as in Section 2.2 with $M = 7$ and $N = 4$. Hence, the pipeline can be broken into two stages in any section of the pipeline to be analyzed.

2.5.1 Sample and Hold Error

The S/H is the first block in each stage, and determines the actual input signal bandwidth for the converter. S/H error can be modeled as a small change in the ideal input sample X. Let the held sample be represented as $X + \Delta_X$. The M and N-bit ADC can be used to show the effects of the sample error component Δ_X at various points in the pipeline architecture. Refer to Figure 2.4. First look at the S/H contributions in the M-bit ADC.

$$\begin{aligned} X + \Delta_X &= \left(q_1 - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right) + e_1 \text{ and} \\ r_1 &= 2^{M-1}e_1. \end{aligned} \tag{2.6}$$

Notice that Δ_X offsets the residual error e_i. The residual r_1 is the input to the next stage. Let Δ_{r_1} be defined as the error from the second stage S/H. The input to the N-bit converter is $r_1 + \Delta_{r_1}$ and can be written

$$r_1 + \Delta_{r_1} = \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^N}\right) + e_2. \tag{2.7}$$

Using the same development as in Section 2.2, substituting Equation 2.6 into 2.7 gives

$$e_1 = \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^{N+M-1}}\right) + \frac{e_2}{2^{M-1}} - \frac{\Delta_{r_1}}{2^{M-1}}. \tag{2.8}$$

The expression for X through a $(N + M - 1)$-bit 2-stage pipeline converter can be shown as

$$\begin{aligned} X + \Delta_X &= \left(q_1 2^{N-1} + q_2 - \left(2^{N+M-2} - 1\right)\right)\left(\frac{2V_R}{2^{M+N-1}}\right) + \frac{e_2}{2^{M-1}} - \frac{\Delta_{r_1}}{2^{M-1}}. \\ X &= \left(q_1 2^{N-1} + q_2 - \left(2^{N+M-2} - 1\right)\right)\left(\frac{2V_R}{2^{M+N-1}}\right) + \frac{e_2}{2^{M-1}} - \Delta_X - \frac{\Delta_{r_1}}{2^{M-1}}. \end{aligned} \tag{2.9}$$

The error from the first stage is unscaled and has the most effect on the output sample. The second stage error is scaled down by 2^{M-1}, so it's effect on the output is reduced. Since the S/H errors become more insignificant as the sample moves through the pipeline, it can be seem that S/H errors are dominated by the first few stages in the pipeline architecture. The above result for the M and N-bit example can be generalized for an arbitrary pipeline architecture. The error at the input to the i^{th} stage in a pipeline will be divided by the total gain in stages preceding that stage. For example, for a 1.5 bits/stage architecture, an error $\Delta_{r_{i-1}}$ at the input of the i^{th} stage (Equation 2.9) will add a component $\frac{\Delta_{r_{i-1}}}{2^{i-2}}$ to the final output error expression. The denominator factor, 2^{i-2}, is the total gain for the $i-1$ previous stages in the pipeline.

2.5.2 Sub-ADC Error

Errors in the sub-ADC of a stage can be characterized as changes in the thresholds of the sub-ADC. Changes in sub-ADC threshold will change the residual error r_i which is passed to the following stages. However, provided the residual error remains within the bounds $|r_i| \leq V_R$, none of the development of Section 2.2 changes. In this case the change in the sub-ADC threshold does not affect the final pipeline converter resolution. Figure 2.10 shows a plot of the residual error produced for a single stage in the 1.5 bits/stage architecture. Notice in the plot that the sub-ADC thresholds can vary by as much as $\pm Q/4$ without allowing the residual error to exceed the bound $|r_i| < Q/4$. This can be seen in Figures 2.11 and 2.12. Figure 2.11 shows the residual error characteristic using the ideal ADC thresholds. The ADC outputs are given in brackets, and determine which of the three error curves is used to produce the residual. Figure 2.12 shows a similar plot but with non-ideal sub-ADC thresholds. Notice that the transition between the error curves are given by the values of the sub-ADC thresholds. Although the thresholds have changed, the residual error will still be in the $\pm V_R$ operating range of the following stage in the pipeline. Changes up to $\pm Q/4$ in the sub-ADC thresholds

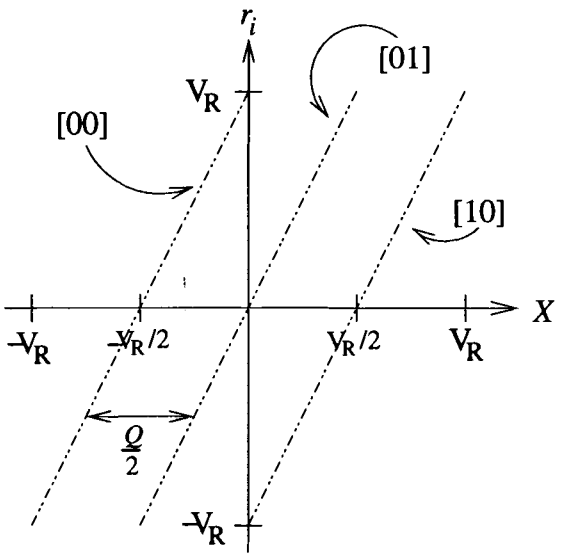


Figure 2.10: Residual characteristics for 1.5 bits/stage.

is tolerable and will not change the final digital output from the ADC. Changes in these thresholds will change the residual signal but this will be quantized in the following stage and the digital correction should compensate for these changes.

2.5.3 Sub-DAC Error

The role of the sub-DAC is to provide a voltage to be subtracted from the input sample to the stage. The sub-DAC in a 1.5 bits/stage structure can have 3 possible output voltages (This corresponds to 3 sub-ADC outputs). Figure 2.8 displays the desired outputs from the sub-DAC. Errors in the sub-DAC are variations in the voltage that is being output. For an M-bit stage, the desired sub-DAC output is $\left(q_1 - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right)$. Let Δ_{D1} represent the error in the sub-DAC output of the first stage. Then (2.1) can be rewritten as

$$\begin{aligned} X &= \left(q_1 - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right) + \Delta_{D1} + e_1 \text{ and} \\ r_1 &= 2^{M-1}e_1. \end{aligned} \tag{2.10}$$

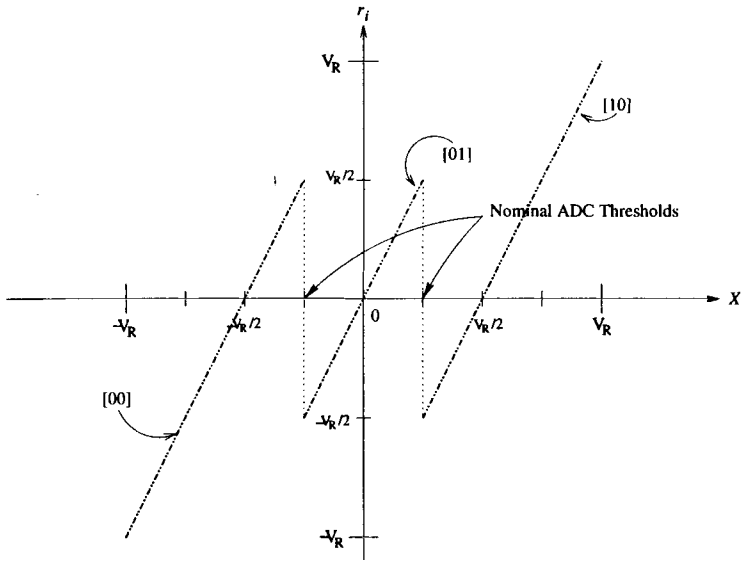


Figure 2.11: Error characteristics for a single stage with nominal thresholds.

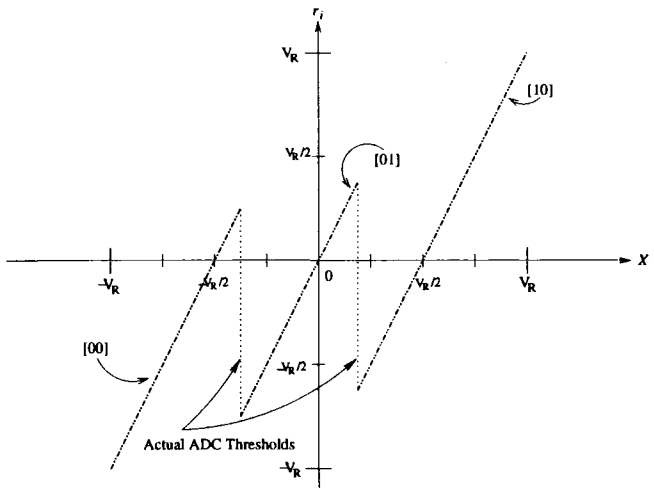


Figure 2.12: Error characteristics for a single stage with actual thresholds.

Defining Δ_{D2} as the second sub-DAC error,

$$\begin{aligned} r_1 &= \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^N}\right) + \Delta_{D2} + e_2 \text{ and} \\ e_1 &= \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^{N+M-1}}\right) + \frac{\Delta_{D2}}{2^{M-1}} + \frac{e_2}{2^{M-1}}. \end{aligned} \tag{2.11}$$

This effect is similar to that of Equation 2.9. The equation for an $(N + M - 1)$-bit converter is

$$X = \left(q_1 2^{N-1} + q_2 - \left(2^{N+M-2} - 1\right)\right)\left(\frac{2V_R}{2^{M+N-1}}\right) + \Delta_{D1} + \frac{\Delta_{D2}}{2^{M-1}} + \frac{e_2}{2^{M-1}}. \tag{2.12}$$

This shows that the sub-DAC errors directly effect the output resolution. The contributions from each stage DAC errors are similar to S/H errors, except that this distortion is expected to be a nonlinear function of the binary DAC input (as opposed to the actual input for the S/H). Sub-DAC errors in each stage are scaled down by the total gain factor from all previous stages. Usually, only the first few stages will show dominance in error contributions. As in the case of S/H errors, errors in the sub-DAC output can be generalized from the M and N-bit architecture described to arbitrary pipeline architectures. At the input of the i^{th} stage, the contribution of Δ_{Di} will be scaled down by the total gain of stages preceding that stage. For the 1.5 bits/stage case, an error Δ_{Di} in the i^{th} stage will be reflected by a term $\frac{\Delta_{Di}}{2^{i-2}}$ in the final error expression. By using this gain factor, any two stages in a pipeline can be analyzed using the method described in this section.

2.5.4 Gain Error

The gain portion of a stage is responsible for scaling the residual error to $\pm V_R$ for the next stage. Changes in the gain will alter the input range for the next stage. Let the gain in the first stage be denoted by $(1+\Delta_{G1})2^{M-1}$, where Δ_{G1} reflects the deviation

from the desired gain. The M-bit ADC equations are

$$\begin{aligned} X &= \left(q_1 - \left(2^{M-1} - 1\right)\right)\left(\frac{2V_R}{2^M}\right) + e_1 \text{ and} \\ r_1 &= (1 + \Delta_{G1})2^{M-1}e_1. \end{aligned} \tag{2.13}$$

The gain error scales the residual value. In the N-bit ADC,

$$\begin{aligned} r_1 &= \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^N}\right) + e_2 \text{ and} \\ e_1 &= \left(q_2 - \left(2^{N-1} - 1\right)\right)\left(\frac{2V_R}{2^{N+M-1}}\right) + \frac{e_2}{2^{M-1}} + e_1\Delta_{G1}. \end{aligned} \tag{2.14}$$

Equations 2.13 and 2.14 can be combined to get the following,

$$X = \left(q_1 2^{N-1} + q_2 - \left(2^{N+M-2} - 1\right)\right)\left(\frac{2V_R}{2^{M+N-1}}\right) + e_1\Delta_{G1} + \frac{e_2}{2^{M-1}}. \tag{2.15}$$

The effect of a gain error in the first stage is the addition of a term which is proportional to the first stage residual. In this case, the distortion contribution is a function of the difference between the stage input and the sub-DAC output. Typically in switched-capacitor designs, the gain is obtained using capacitor ratios and changes in the capacitance will affect the gain values. Gain error typically reduces the effective number of bits of the converter. Generalizing this result for the i^{th} stage in a pipeline, the gain error term $e_i\Delta_{Gi}$ will be scaled down by the total gain of stages preceding that stage. For the 1.5 bits/stage architecture, the error term $e_i\Delta_{Gi}$ will be reflected by a term $\frac{e_i\Delta_{Gi}}{2^{i-2}}$ in the final error expression.

2.6 Measuring Distortion

The driving force behind this thesis is to obtain better dynamic range or linearity in a pipeline converter and to test the validity of using intermediate stage outputs for

compensation purposes. A common measure of linearity used in the industry is called SFDR - Spurious Free Dynamic Range. SFDR is the difference in dB between the power of the desired signal and the highest spurious signal in the output spectrum of the converter. For example, Figure 2.13 shows a spectral plot of output samples collected from a Texas Instruments THS1240 (12-bit 40MHz) ADC [13]. The converter was driven using a pure sinusoidal signal at 15.5 MHz with an amplitude of -2 dBFS. 'dBFS' is the power level of the signal in relative to a full-scale sinusoid. Distortion introduced by the converter is reflected by a large number of harmonics in the calculated output spectral plot. The numbered terms in the spectral plot represent harmonics of the signal of interest. For example, '1' is the fundamental of the signal, '2' is the second harmonic, etc. Higher order harmonics outside the first Nyquist band are aliased into the baseband. The largest spurious signal in the output spectrum is identified as the 3^{rd} harmonic of the input signal, at an amplitude of -79.1 dBFS. For this example the SFDR value is approximately 77.1 dB.

Other terms used to characterize a converter include SINAD - Signal to Noise Ratio and Distortion. This measure is the ratio between the signal power and all remaining power in the output sample sequence [15]. ENOB - Effective Number of Bits is another measure and is given by

$$\text{ENOB} = \frac{1}{6.02}(\text{SINAD} - 1.76 - L), \tag{2.16}$$

where L is the loading of the signal in dBFS. For an ideal N-bit ADC, the ENOB would be N bits, SINAD would be $6.02N + L + 1.76$, and SFDR would be arbitrarily large.

2.7 Simulated Pipeline ADC

A pipeline converter is simulated using the design and architecture described in Section 2.4. A 10-bit ADC using 1.5 bits/stage topology is simulated using MATLAB.

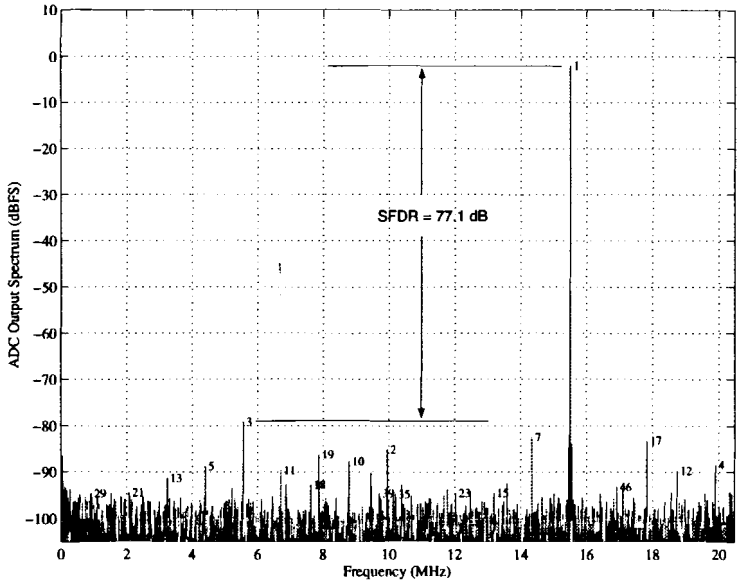


Figure 2.13: Spectral plot of THS1240 ADC with $F_t = 15.5$ MHz.

Nine stages and digital correction as described in Section 2.4 are used, and all the thresholds and gains in each stage may be controlled independently. This is important in order to include the error mechanisms in the pipeline and to measure the effects on the converter performance. The sample rate was set to 4.096 MHz to replicate the hardware implementation described in Chapter 4. Figure 2.14 shows a spectral plot of samples from an ideal simulated converter. In this plot, all the values for thresholds and reference voltages are kept ideal. The input range for this 10-bit ADC was set to ± 1V, giving an ideal quantization interval of $Q = 2/1024 = 0.002$ V. The plot was obtained using an input signal of 1 MHz and amplitude -2 dBFS with additive White Gaussian Noise with variance $1.14(Q^2/12)$.

Typical S/H errors are nonlinearly related to the input voltage and time rate of change of the S/H signal [2]. Figure 2.15 shows a plot of the input signal $X(t)$ as a function of time, t. Let t_o denote the ideal sample time of interest. Then $X(t)$ may be approximated by a Taylor series $X(t) = X(t_o) + \dot{X}(t_o)(t - t_o)$ for values of t close to t_o. If the actual sample time for the S/H deviates from the desired value of t_o, an error

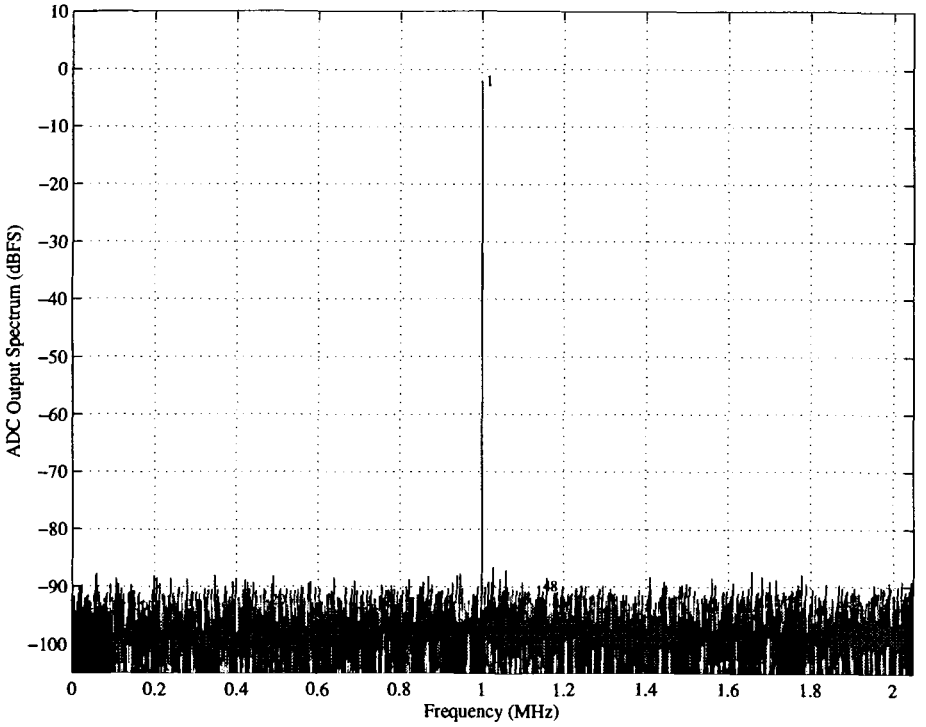


Figure 2.14: Simulated 10-bit Pipeline (Ideal) with F_t = 1 MHz.

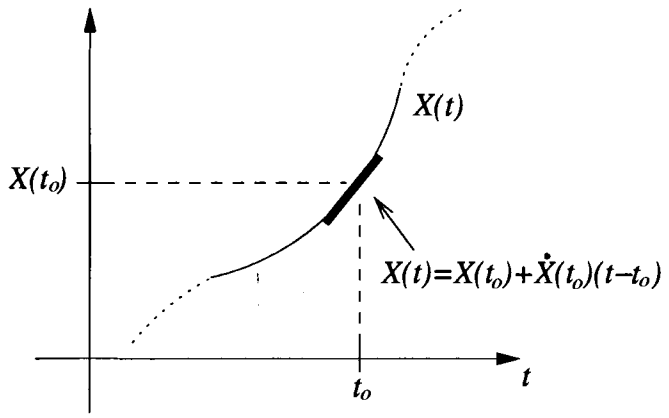


Figure 2.15: Time varying S/H input signal.

in the S/H output is generated. The error is given by

$$\Delta_X = \dot{X}(t_o)(t_{sample} - t_o). \tag{2.17}$$

If the sample time offset is random (sample time jitter), then the effect of Δ_X (2.17) is the addition of a noise component in the output of the converter (with amplitude modulated by $\dot{X}(t)$) [16]. This effect does not generally introduce harmonic distortion in the output signal. However, most S/H circuits introduce sample time errors which are nonlinearly related to the voltage being sampled. Letting $g(X(t_o))$ be the sample time offset as a function of the input voltage gives

$$\Delta_X = \dot{X}(t_o)g(X(t_o)). \tag{2.18}$$

This term does introduce harmonic distortion in the output spectrum.

Figure 2.16 shows a basic S/H circuit which consists of a NMOS gate connected to a sampling capacitor. The NMOS switch is clocked at the sampling frequency of the ADC. An example of the clock and input waveform is also displayed in the figure. V_{GS} is the gate-to-source voltage of the NMOS switch given by $V_{GS} = V_{Clk} - X(t)$. V_{Clk} is V_1 when the clock is high and transitions to V_0 at the sample instant. Figure 2.17 shows

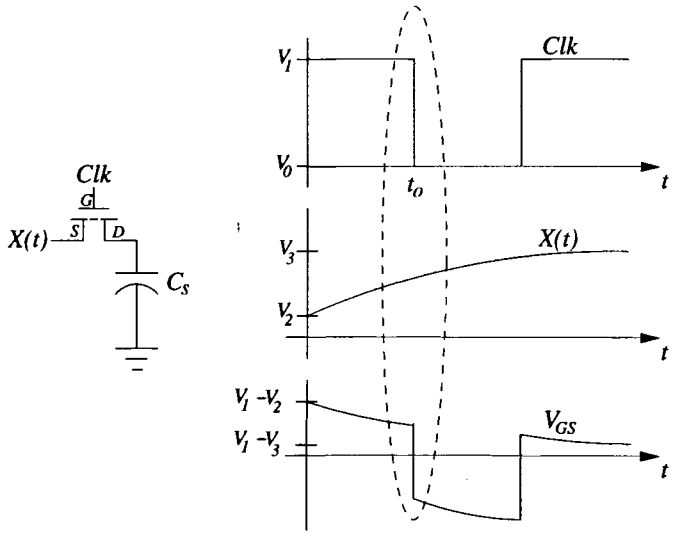


Figure 2.16: Model of S/H circuit.

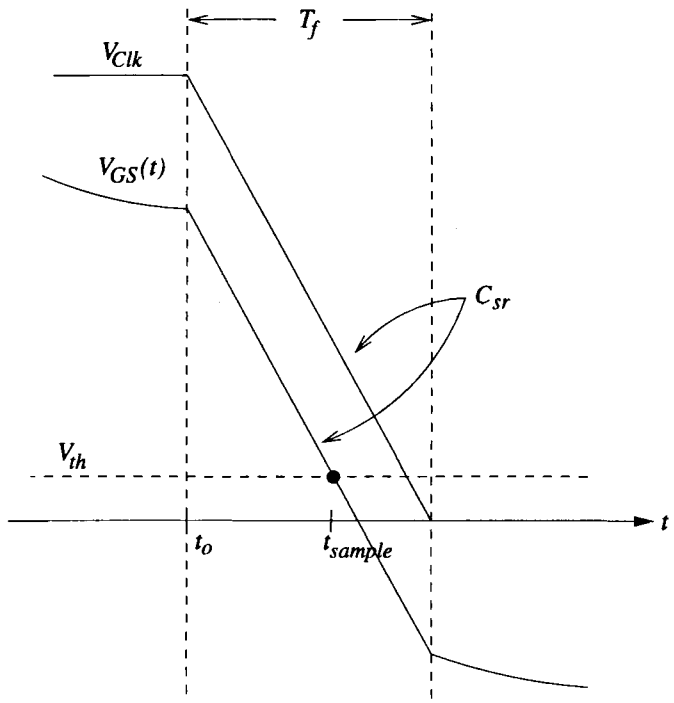


Figure 2.17: Special case in S/H circuit.

an enlarged portion of the dashed circle in the previous figure. T_f is the fall time of the clock transition, V_{th} is the threshold voltage for the NMOS device and C_{sr} is the clock slew rate. Assuming a constant clock slew rate over the clock transition,

$$C_{sr} = \frac{V_1 - V_0}{T_f}. \tag{2.19}$$

This clock slew rate determines the sample time by

$$t_{sample} - t_o = \frac{V_{GS}(t_o) - V_{th}}{C_{sr}}. \tag{2.20}$$

Combining $V_{GS}(t_o) = V_1 - X(t_o)$ and (2.20) gives the functional form for $g(X(t_o))$ for this NMOS switch sample

$$g(X(t_o)) = t_{sample} - t_o = \frac{V_1 - X(t_o) - V_{th}}{C_{sr}}. \tag{2.21}$$

Using this term to represent the switching characteristic, Δ_X can be rewritten

$$\begin{aligned} \Delta_X &= \dot{X}(t_o)\frac{V_1 - X(t_o) - V_{th}}{C_{sr}} \\ &= \dot{X}(t_o)\frac{T_f(V_1 - X(t_o) - V_{th})}{V_1} \end{aligned} \tag{2.22}$$

This model was implemented in the simulation model of the ADC to model S/H error characteristics using $V_1 = 5$V, $V_{th} = 0.9$V, and $T_f = 1$ns.

Apart from the S/H error, variations in each threshold for every stage was also included. The values are tabulated in Table 2.1. The locations of these parameters in each stage are shown in Figure 2.18. Sub-DAC error is controlled by the DAC thresholds. Sub-ADC error is controlled by the ADC threshold locations. Gain error is included by variations in the gain value which has an ideal value of 2 for the 1.5 bits/stage architecture.

	2(DAC1)	2(DAC2)	2(DAC3)	ADC1	ADC2	Gain
Ideal	-1.000	0.000	1.000	-0.25	0.25	2.000
Stage 1	-1.002	0.001	0.998	-0.23	0.35	1.995
Stage 2	-1.003	0.002	0.998	-0.20	0.30	2.003
Stage 3	-0.997	-0.001	0.999	-0.20	0.35	1.990
Stage 4	-1.006	-0.001	1.002	-0.25	0.25	1.990
Stage 5	-1.002	0.001	1.002	-0.22	0.28	2.001
Stage 6	-0.998	0.000	0.998	-0.27	0.22	1.997
Stage 7	-0.998	0.002	0.998	-0.30	0.35	2.002
Stage 8	-1.005	0.000	1.005	-0.28	0.23	1.999
Stage 9	-1.005	0.001	1.005	-0.20	0.20	2.000

Table 2.1: Parameters for simulated 10-bit Pipeline ADC (Units in Volts).

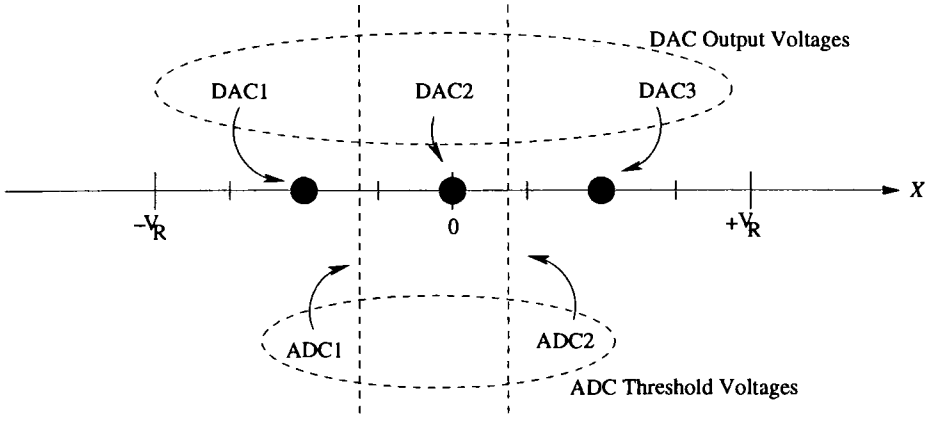


Figure 2.18: Locations of parameters for simulated ADC.

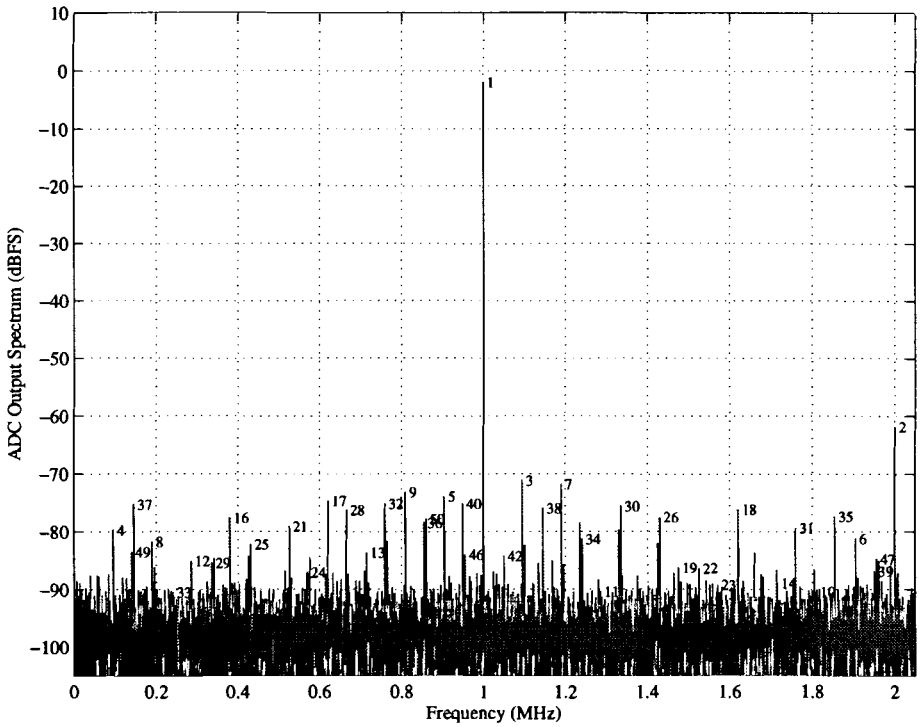
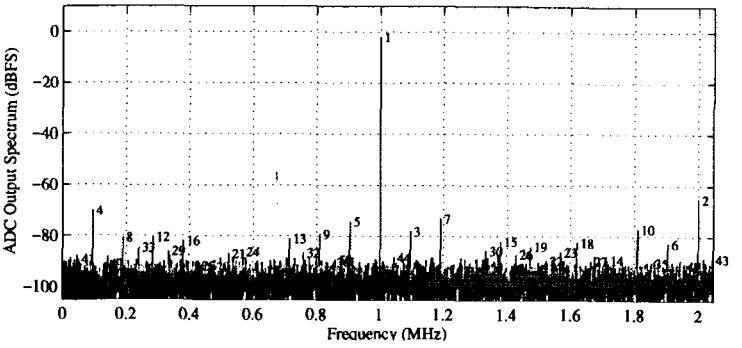


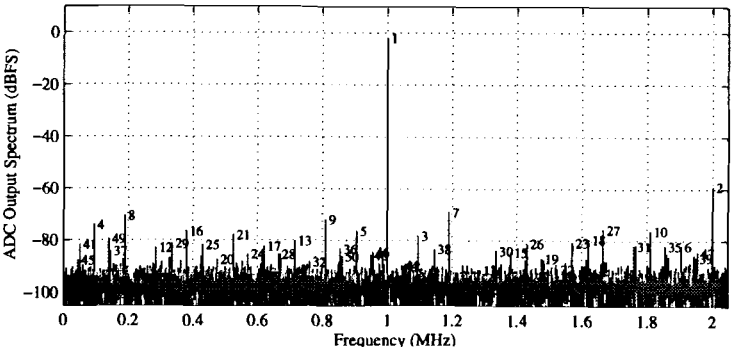
Figure 2.19: Simulated 10-bit Pipeline (Non-ideal) with F_t = 1 MHz.

Figure 2.19 shows a spectral plot of output samples collected using the simulated converter with all the errors incorporated into the model. The test signal was at a frequency of 1 MHz at an amplitude of -2 dBFS. The input can have a $\pm$1V range (V_R= 1V). Simulation parameters were selected to give realistic distortion levels reflecting measurements of actual converters.

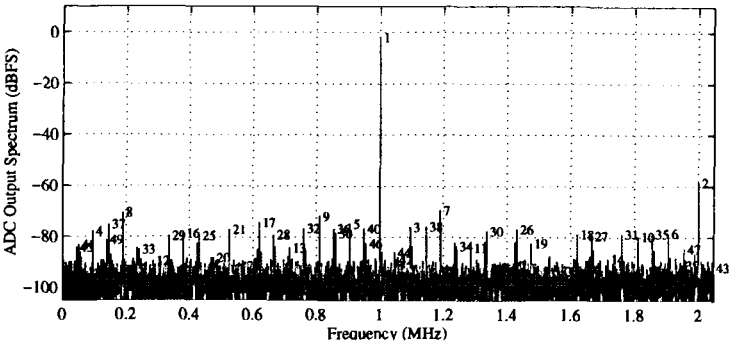
Simulations confirm that errors in the first stage of the converter are dominant in a pipeline. Any error produced in the front end of the pipeline is carried through the entire conversion. Figure 2.20 shows spectral plots with errors present in different locations in the pipeline. The plots provide a visual representation of how much distortion from the first three stages is present in a pipeline converter. In conclusion, the simulated converter behaves as predicted using error mechanisms presented in previous sections.



(a) Output spectrum with error in first stage only.



(b) Output spectrum with error in first and second stages only.



(c) Output spectrum with error in first, second and third stages only.

Figure 2.20: Simulated ADC output spectra showing error dominance of early stages in a pipeline.

CHAPTER 3

Calibration and Compensation of Pipeline ADCs

The previous chapter discussed the workings of a pipeline ADC and some error mechanisms that affect ADC performance. This chapter describes the concept of dynamic error compensation and methods of implementing the scheme (calibration). This chapter concludes with a discussion of testing the proposed scheme on a simulated pipeline described in Chapter 2.

3.1 Proposed Compensation Architecture

In the past, significant work has been done in compensating dynamic error in ADCs [3, 6, 7]. Although there are many approaches to this problem, the ultimate goal is to provide a solution that eliminates any signal distortion introduced by the ADC. In the industry, ADC manufacturers typically set SFDR specifications in the range of the quantization error levels of the converter. This implies the dynamic range of an ADC is expected to be around $6N$ dB, N being the number of bits. This means that the difference in power between the fundamental and the next highest spurious signal will be in the order of $6N$ dB [5].

Schemes implementing error compensation on flash converters have been successful [1]. However efforts in compensating pipeline ADCs have not been very successful. Previous techniques have modeled the error introduced by the ADC as a function of the ADC output code and a combination of the input signal derivative and/or previous ADC output code [3, 17]. These schemes are not as effective on a pipeline when compared to a flash ADC [11]. This failure in compensation technique is due to the different architecture that exists in a pipeline converter. This thesis proposes a scheme using the intermediate stage outputs from early stages of the pipeline to characterize the non-linearity introduced by the converter.

A close look at the pipeline stage and its error mechanisms show that the sub-DAC non-linearity can only be predicted using inputs to the sub-DAC. Figure 3.1 shows the first two stages in a pipeline with errors in each path shown as a function of input, slope, stage outputs and residual error. $\dot{X}$ is the derivative of the input signal captured by the S/H. f_s, f_d and f_g represent errors from the S/H, sub-DAC, the gain and summing components respectively. In the figure, the errors introduced by these components are represented as functions added into the signal path during conversion.

S/H errors are included for each stage of the pipeline. However, only the first stage includes the slope-dependencies of the S/H error. This is because the first stage input is the dynamic input signal, while all other stages are sampling the held residuals being produced by the previous stage.

Previous efforts in compensation did not utilize the sub-ADC outputs, which made compensation of the distortion caused by f_d and f_g difficult. As shown in Figure 3.1, the error introduced by the sub-DAC is a function of the sub-ADC outputs. This is also true for error introduced after the residual error is calculated, which is a function of the stage outputs and the input signal. This model suggests that intermediate outputs from each stage are important in predicting the error introduced by the converter. The proposal of this thesis is to use the intermediate stage output bits in the compensation scheme.

ADC calibration involves the estimation of the unknown distortion characteristic for a particular converter. The resulting characteristic could be stored in a table and used to compensate for the distortion as the ADC operates in real time. To calibrate the converter, the converter is tested with many input signals and the data is used to solve for coefficients associated with generic error expressions. These coefficients may then be used to build a look-up table for the error.

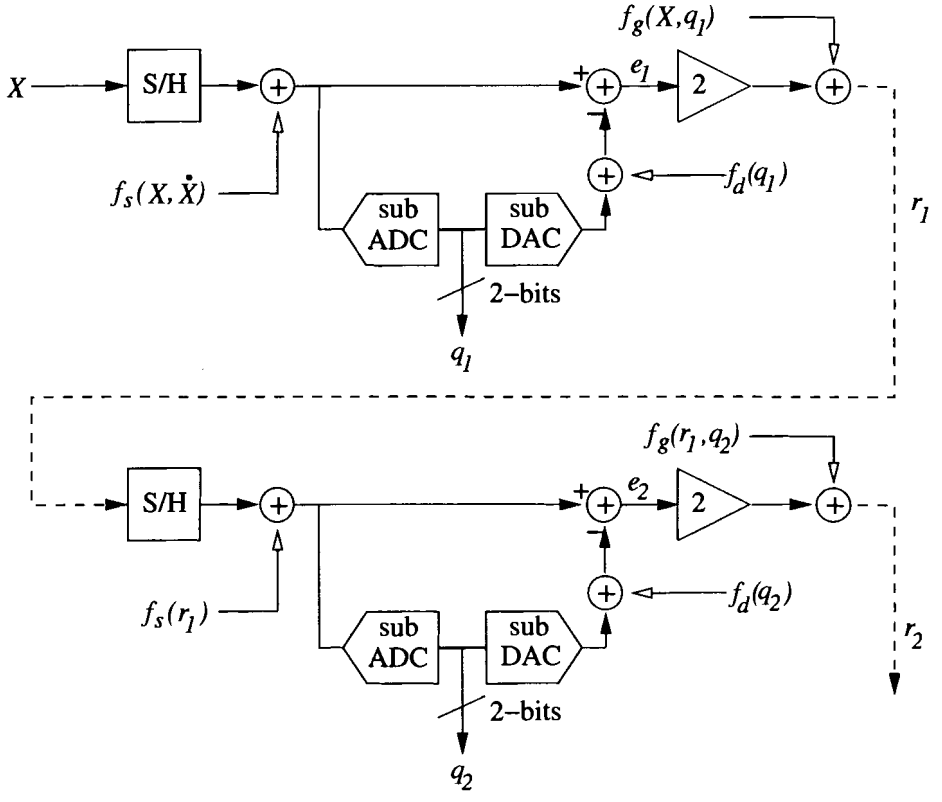


Figure 3.1: First and second stage block diagram with error models.

3.2 Calibration Scheme

In order to compensate an ADC, a look up table with the error values needs to be created. For example, for a slope dependent error, the error is a function of the input and its derivative. Since the ADC output code is (almost) proportional to the input voltage, an attempt is made to predict the error as a function of the output code and the numerically calculated rate of change of this code.

The estimated error function is denoted as $e(x_i, d_i)$, where x_i is the ADC output code for the i_{th} sample and d_i denotes other information on which the error depends (input signal derivative, q_1, q_2, etc.). Then the compensated value, y_i for the i_{th} sample can be represented by

$$y_i = x_i - e(x_i, d_i). \tag{3.1}$$

This means for every possible input sample there is a corresponding estimated error value that will be subtracted from it. A calibration scheme is used to create an error expression $e(x_i, d_i)$ that characterizes the error for all possible ADC states. In this thesis, calibration is accomplished by driving the converter with sinusoidal test signals and mathematically solving for a function $e(x_i, d_i)$ which minimizes the measured distortion power.

The proposed calibration scheme is based on the non-iterative method [5] and does not require knowledge of the input signal amplitude or phase. This scheme also does not rely on test signals that cover all the possible output codes, but uses multiple amplitudes and an array of test frequencies to cover most of the ADC output codes. This reduces calibration time compared to [3]. Since harmonic content is important for error calculations, all test signals are filtered sinusoidal signals to remove distortion and to approximate pure sine-waves. To avoid repetition of samples and spectral leakage, test frequencies are chosen to be odd multiples of F_s/n. F_s is the sampling frequency of the ADC and n is the number of samples obtained for each of the calibration signals.

Assume F_t is a test frequency used for calibration. A vector notation is convenient to represent the compensation of an entire calibration signal. Let $\vec{x} = [x_1, x_2 ... x_n]^T$ represent a vector of ADC output samples collected for a single calibration signal. Let $\vec{d} = [d_1, d_2 ... d_n]$ be a vector of values corresponding to the error function being characterized. Rewriting Equation 3.1 in a vector form, the vector of compensated values for n samples is

$$\vec{y} = \vec{x} - \vec{e}(\vec{x}, \vec{d}) \tag{3.2}$$

where

$$\vec{e}(\vec{x}, \vec{d}) = \begin{bmatrix} e(x_1, d_1) \\ e(x_2, d_2) \\ \vdots \\ e(x_n, d_n) \end{bmatrix}. \tag{3.3}$$

Since the goal is to minimize distortion, the distortion in each calibration test signal may be measured by evaluating the FFT of $\vec{y}$ at the harmonics of the input calibration frequency, F_t. Let $\tilde{T}$ be the transformation matrix which gives the 2^{nd} through M^{th} harmonic of the input signal, then

$$\tilde{T} = \begin{bmatrix} e^0 & e^{-j2\pi\frac{f_2}{F_s}} & e^{2(-j2\pi\frac{f_2}{F_s})} & e^{3(-j2\pi\frac{f_2}{F_s})} & \dots & e^{(n-1)(-j2\pi\frac{f_2}{F_s})} \\ e^0 & e^{-j2\pi\frac{f_3}{F_s}} & e^{2(-j2\pi\frac{f_3}{F_s})} & e^{3(-j2\pi\frac{f_3}{F_s})} & \dots & e^{(n-1)(-j2\pi\frac{f_3}{F_s})} \\ \vdots & & & & & \\ e^0 & e^{-j2\pi\frac{f_M}{F_s}} & e^{2(-j2\pi\frac{f_M}{F_s})} & e^{3(-j2\pi\frac{f_M}{F_s})} & \dots & e^{(n-1)(-j2\pi\frac{f_M}{F_s})} \end{bmatrix}. \tag{3.4}$$

$\tilde{T}$ is an $(M-1) \times n$ matrix with $f_i = iF_t$ for $i = 2, ..., M$. This transformation is used because most of the distortion in a sinusoidal signal is assumed to be contained in the first M harmonics of the signal. Hence $\vec{Y} = \tilde{T}\vec{y}$ is the vector containing the 2^{nd} through M^{th} harmonics of the compensated signal. The objective of this compensation scheme is to be able to minimize the total error contained by these harmonics. Using Equation

3.2, $\vec{Y}$ may be written in terms of the sample vector $\vec{x}$ and the error characteristic

$$\vec{Y} = \tilde{T}\vec{y} = \tilde{T}\vec{x} - \tilde{T}\vec{e}. \tag{3.5}$$

Since the error functions are non-linear and hard to predict, basis function representations are used to represent $e(x, d)$. For a set of L known basis functions,

$$e(x, d) = \sum_{k=1}^{L} \alpha_k b_k(x, d). \tag{3.6}$$

Applying the transformation matrix to the error functions,

$$\tilde{T}\vec{e} = \tilde{T}\tilde{B}\vec{\alpha} \tag{3.7}$$

where

$$\tilde{B} = \begin{bmatrix} b_1(x_1, d_1) & b_2(x_1, d_1) & \dots & b_L(x_1, d_1) \\ b_1(x_2, d_2) & b_2(x_2, d_2) & \dots & b_L(x_2, d_2) \\ \vdots & & & \\ b_1(x_n, d_n) & b_2(x_n, d_n) & \dots & b_L(x_n, d_n) \end{bmatrix} \text{ and } \vec{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_L \end{bmatrix}. \tag{3.8}$$

$\vec{\alpha}$ is a vector of coefficients used to weight the basis functions in $\tilde{B}$. The vector $\tilde{T}\tilde{B}$ gives the 2^{nd} through M^{th} harmonic of each of the basis function evaluated on the calibration data. The compensated harmonic components $\vec{Y}$, can be written as a linear function of unknown basis function coefficients $\vec{\alpha}$,

$$\vec{Y} = \tilde{T}\vec{x} - \tilde{T}\tilde{B}\vec{\alpha}. \tag{3.9}$$

The goal is to select coefficients $\vec{\alpha}$ which minimize the length of the vector $\vec{Y}$ for a collection of input signals at various frequencies and amplitudes. This presents a

least-squares problem that can be evaluated with data from K calibration test signals. Rewriting Equation 3.9 for the entire set of K test signals gives,

$$\begin{bmatrix} \vec{Y}_{(1)} \\ \vec{Y}_{(2)} \\ \vdots \\ \vec{Y}_{(K)} \end{bmatrix} = \begin{bmatrix} \tilde{T}_{(1)}\vec{x}_{(1)} \\ \tilde{T}_{(2)}\vec{x}_{(2)} \\ \vdots \\ \tilde{T}_{(K)}\vec{x}_{(K)} \end{bmatrix} - \begin{bmatrix} \tilde{T}_{(1)}\tilde{B}_{(1)} \\ \tilde{T}_{(2)}\tilde{B}_{(2)} \\ \vdots \\ \tilde{T}_{(K)}\tilde{B}_{(K)} \end{bmatrix} \vec{\alpha} \tag{3.10}$$

For the calibration set, $\vec{Y}$ represents the harmonics of compensated test signals and should be minimized to calculate the error characteristic. The minimization in the least-squares sense can be obtained using the pseudo-inverse of $\tilde{T}\tilde{B}$ [18]. In implementation, the values in $\tilde{T}\tilde{B}$ are obtained directly using FFTs of each of the vectors $\vec{b_k}(\vec{x}, \vec{d})$ to obtain the $(M-1) \times L$ matrix. This method saves time and memory instead of evaluating the large matrices of $\tilde{T}$ and $\tilde{B}$ separately. Hence the solution for $\vec{\alpha}$ can be shown to be

$$\vec{\alpha} = \begin{bmatrix} \mathcal{R}\{\tilde{T}_{(1)}\tilde{B}_{(1)}\} \\ \mathcal{R}\{\tilde{T}_{(2)}\tilde{B}_{(2)}\} \\ \vdots \\ \mathcal{R}\{\tilde{T}_{(K)}\tilde{B}_{(K)}\} \\ \mathcal{I}\{\tilde{T}_{(1)}\tilde{B}_{(1)}\} \\ \mathcal{I}\{\tilde{T}_{(2)}\tilde{B}_{(2)}\} \\ \vdots \\ \mathcal{I}\{\tilde{T}_{(K)}\tilde{B}_{(K)}\} \end{bmatrix}^{+} \begin{bmatrix} \mathcal{R}\{\tilde{T}_{(1)}\vec{x}_{(1)}\} \\ \mathcal{R}\{\tilde{T}_{(2)}\vec{x}_{(2)}\} \\ \vdots \\ \mathcal{R}\{\tilde{T}_{(K)}\vec{x}_{(K)}\} \\ \mathcal{I}\{\tilde{T}_{(1)}\vec{x}_{(1)}\} \\ \mathcal{I}\{\tilde{T}_{(2)}\vec{x}_{(2)}\} \\ \vdots \\ \mathcal{I}\{\tilde{T}_{(K)}\vec{x}_{(K)}\} \end{bmatrix} \tag{3.11}$$

where $[\ ]^{+}$ denotes a pseudo-inverse, $\mathcal{R}\{\ \}$ denotes the real term, $\mathcal{I}\{\ \}$ denotes the imaginary term, and K is the number of calibration signals. Once values for $\vec{\alpha}$ are obtained, they are used to weigh the basis functions evaluated for the input samples to be compensated. Hence if $\vec{\alpha}$ is known then for L number of basis functions and an input,

$\vec{x}$, the compensated signal, $\vec{y}$, can be evaluated using

$$\vec{y} = \vec{x} - \sum_{k=1}^{L} \alpha_k b_k(\vec{x}, \vec{d}). \tag{3.12}$$

3.3 Basis Functions

In the previous section, the basis functions were described as fairly generic functions for the compensation scheme. The ultimate goal of the basis functions are to characterize a particular error mechanism and can be selected to match a given ADC architecture. These functions should be selected to match the theoretical error models. In the following sections, S/H error and sub-DAC error are fitted with basis functions.

3.3.1 S/H Error Basis Functions

Section 2.5.1 described the possible mechanisms behind S/H errors. The S/H error can be modeled as $f_s(X, \dot{X}) = g(X)\dot{X}$, as described in Section 2.5.1. $g(X)$ is an unknown function of the S/H switching characteristic. For compensation, this function must be written in terms of the ADC output code x. Up to the quantization error of the converter, the ADC input voltage is linearly related to the output code

$$X \approx \frac{x - x_{off}}{S}, \tag{3.13}$$

where x_{off} is an offset and S is a scale factor. Using this approximation, the unknown characteristic $g(X)$ can be modeled using Gaussian basis functions, giving

$$g(X) \approx g\left(\frac{x - x_{off}}{S}\right) = \sum_{i=1}^{L} \alpha_i e^{-\frac{1}{2}\left(\frac{x-c_i}{\sigma_x}\right)^2}, \tag{3.14}$$

Here $g(X)$ is modeled as a summation of L basis functions that have coefficients $\alpha_1, \alpha_2, ..., \alpha_L$. The c_i values are centers for the Gaussian functions and are chosen to cover most of

the possible output codes of the ADC. σ_x is the standard deviation of these functions and is chosen to provide significant overlap between functions to ensure a good fit to the switching error characteristic. Estimates of the slope $\dot{X}$ may be calculated using a variety of techniques including discrete-time filtering of the output sequence, using a low-resolution sample of an analog slope waveform, or by using several ADCs with sample clocks slightly offset from each other [2, 4]. Let $\hat{\dot{X}}$ represent the estimate derivative. The generic S/H error function can now be written in a basis function representation using basis functions of the form

$$b_i(x, \dot{X}) = \hat{\dot{X}} e^{-\frac{1}{2}\left(\frac{x-c_i}{\sigma_x}\right)^2} \text{ for } i = 1, 2, ..., L. \tag{3.15}$$

These basis functions are only used for f_s in the first stage because the S/H in the second stage is not a function of the input signal derivative as shown in Figure 3.1. The second stage S/H error is $f_s(r_1)$, which is a function of the residual from the first stage. Additional basis functions for the second and later stage S/H's were not added, since arbitrary functions of the residual will be modeled by the basis functions used for $f_g(\,)$.

3.3.2 Intermediate Bits Basis Functions

A new addition to the typical dynamic compensation scheme is the addition of basis functions to characterize the sub-DAC non-linearity in each stage. The error in the sub-DAC is a function of the intermediate stage outputs. The sub-DAC errors are represented in Figure 3.1 as f_d and the gain error is shown as f_g.

For the first stage in a pipeline, assume X is the input sample, N is the stage resolution in bits and q_1 is the corresponding sub-ADC output (q_1=0,1,...,$2^N - 2$). $f_d(q_1)$ is a function of the sub-DAC error and depends on the stage output,

$$f_d(q_1) = \alpha_1\delta(q_1) + \alpha_2\delta(q_1 - 1) + ... + \alpha_{(2^N-2)}\delta(q_1 - (2^N - 2))$$

$$= \sum_{i=0}^{2^N-2} \alpha_i \delta(q_1 - i). \tag{3.16}$$

Here $\delta(k)$ is the Kronecker delta function, for

$$\delta(k) = \begin{cases} 0 & k \neq 0 \\ 1 & k = 0 \end{cases} \tag{3.17}$$

Assuming the s^{th} stage in a pipeline has N_s-bits of resolution, this error characteristic can be represented with the following basis functions,

$$b_i(q_s) = \delta(q_s - i), \quad \text{for } i = 0, 1, ..., 2^{N_s} - 2. \tag{3.18}$$

Therefore $(2^{N_s} - 1)$ basis functions are needed to characterize $f_d(q_s)$ in this stage.

The second error function using the intermediate stage outputs is $f_g(X, q_1)$ for the first stage. The gain error characteristic of the first stage is a function of its input, X, and stage output bits, q_1. Depending on the stage output, different nonlinear characteristics may be realized and can be shown

$$\begin{aligned} f_g(X, q_1) &= \begin{cases} g_0(X) & q_1 = 0 \\ g_1(X) & q_1 = 1 \\ \vdots & \\ g_{2^N-2}(X) & q_1 = 2^N - 2 \end{cases} \\ &= g_0(X)\delta(q_1) + g_1(X)\delta(q_1 - 1) + ... + g_{2^N-2}(X)\delta(q_1 - 2^N - 2). \end{aligned} \tag{3.19}$$

The unknown error characteristic $g_i(X)$ was represented by polynomials of X and can be written as

$$g_i(X) = \sum_{l=0}^{\theta} \alpha_l^{(i)} X^l. \tag{3.20}$$

These unknown functions can be used to estimate f_g, and (for N-bits of resolution)

$$f_g(X, q_1) = \sum_{i=0}^{2^N-2} \sum_{l=0}^{\theta} \alpha_l^{(i)} X^l \delta(q_1 - i). \tag{3.21}$$

The total number of basis functions for this stage will be $(\theta+1)(2^N-1)$ to characterize $f_g(X, q_1)$. Generic basis functions for the s^{th} stage with an input sample $X \approx \frac{x-x_{off}}{S}$, stage resolution of N_s-bits and stage outputs q_s can be shown as

$$b_k(x, q_s) = \left(\frac{x - x_{off}}{S}\right)^l \delta(q_s - i), \quad \text{for } i = 0, 1, ..., 2^N - 2. \quad l = 0, 1, ..., \theta. \tag{3.22}$$

The total number of basis functions for both, f_d and f_g, are $\theta(2^N-1)+(2^N-1)$. However, in the case of $l = 0$, $b_k(x, q_s)$ are the same basis functions given by $b_i(q_s)$. These functions need not be repeated, hence only using $b_k(x, q_s)$ is sufficient to model the error characteristics using the intermediate stage outputs. This means there will be a total number of $(\theta + 1)(2^{N_s} - 1)$ basis functions using the intermediate stage outputs.

3.4 Calibration Algorithm

The ADC is tested with a set of sinusoidal calibration signals to calculate the $\vec{\alpha}$ values. The code vs. slope plane is used to determine the coverage in terms of frequency and amplitude of the calibration frequencies. Figure 3.2 shows the code/slope plot of 20 calibration frequencies used to calibrate a 10-bit ADC. Each of the 10 calibration frequencies has two amplitudes to cover more of the code/slope area. The amplitudes for calibration were chosen to be -0.5 dBFS and -6 dBFS. The calibration frequencies were stepped in 182 kHz steps starting from 409 kHz to 2.047 MHz. The lowest test frequency was determined by the filters available in the test setup and the highest calibration frequency was set to the nearest odd multiple of F_s/n to the Nyquist

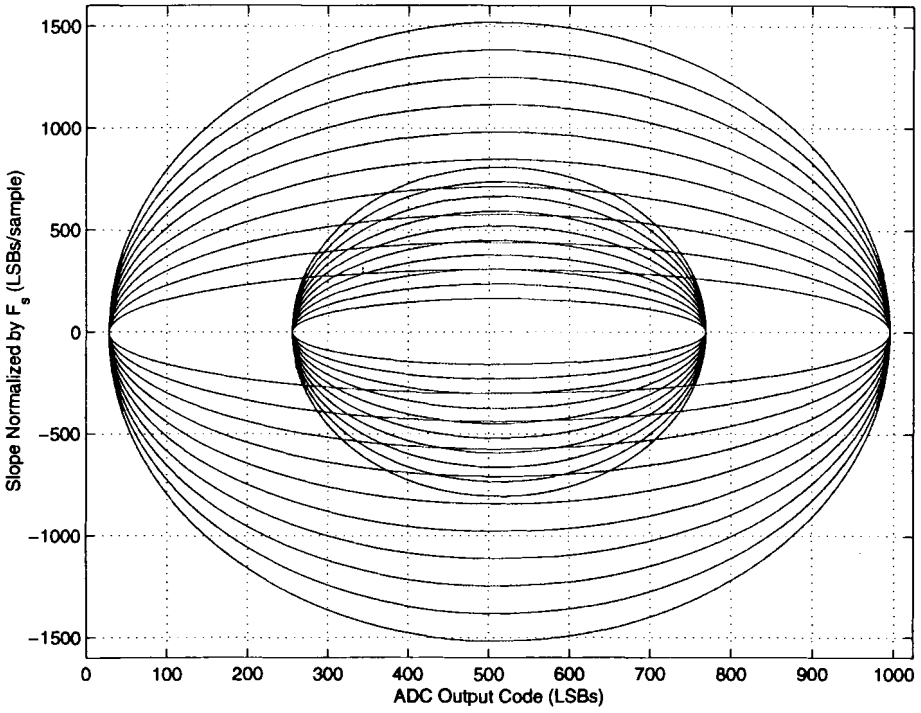


Figure 3.2: Calibration trajectories in the code/slope space.

frequency $F_s/2$, where n is the number of samples and F_s is the sampling frequency of the ADC.

Once the data is collected for the calibration signals, $\tilde{T}\vec{x}$ and $\tilde{T}\tilde{B}$ are calculated using FFTs. The coefficient vector is then obtained using Equation 3.11. These coefficients are then used to compensate any test signal using Equation 3.12.

3.5 Implementation on Simulated ADC

The compensation scheme described in previous sections was implemented using the simulated ADC described in Section 2.7. The simulated ADC was clocked at $F_s = 4.096$ MHz. Data for each calibration frequency were 2^{14} (16k) samples long. Calibration was done using frequencies and amplitudes listed in the previous section. The 2^{nd} through 20^{th} harmonic was used for calibration.

The calibration data collected was used to calculate the corresponding $\vec{\alpha}$ values as outlined in Section 3.2. However, when calculating the pseudo-inverse, the tolerance in MATLAB's 'pinv' was set to 100. This value was based on the singular value decomposition (SVD) of $\tilde{T}\tilde{B}$. A sharp decrease in singular values indicates any value below that transition is unstable and the range of values before the decrease is used as the tolerance.

The basis functions were chosen based on Section 3.3. For this 10-bit ADC, values of $x = S = 512$ were used in Equation 3.13. Nine basis functions of the form given in Equation 3.15 were included using $c_i = [0, 128, 256, 384, ..., 1024]$ and $\sigma_x = 128$. The outputs from the first 3 stages in the pipeline were used for the basis functions to model the sub-DAC non-linearity. Basis functions given by Equation 3.22 were implemented using $N = 2$ and $\theta = 1$. This gives six basis functions for the first stage output q_1. Defining q_2 as the second stage output and r_1 to be the residual output of the first stage, which is the input of the second stage, $r_1 = 2(X - \frac{q_1 - 1}{2})$. The basis functions included for the second stage are,

$$\begin{aligned} b_k(r_1, q_2) = (r_1)^l \delta(q_2 - i), \text{ for } i &= 0, 1, 2. \\ l &= 0, 1. \end{aligned} \tag{3.23}$$

There are six basis functions for the second stage. The third stage is similar to stage 2, with $r_2 = 2(r_1 - \frac{q_2 - 1}{2})$. q_3 and r_2 are the third stage output bits and third stage inputs respectively. The third stage basis functions are

$$\begin{aligned} b_k(r_2, q_3) = (r_2)^l \delta(q_3 - i), \text{ for } i &= 0, 1, 2. \\ l &= 0, 1. \end{aligned} \tag{3.24}$$

This stage provides another six basis functions for error characterizations. This gives a total of 18 basis functions for error characterization using intermediate stage outputs

and 9 basis functions using the slope dependent error characteristics. In total, 27 basis functions are used to characterize the unknown error mechanisms in the simulated pipeline ADC.

To maintain numerical stability and to control any offsets in $\vec{\alpha}$ calculations, the fundamental and DC component of the calibration signal with $F_t = 1.319$ MHz (-0.5 dBFS) was extracted from its FFT (same method as in Section 3.2) and added to the $\tilde{T}\vec{x}$ and $\tilde{T}\tilde{B}$ arrays during the minimization. Since the basis function coefficients may have many solutions, these new terms help orientate $\vec{\alpha}$ so that it does not have drastic offsets or scaling, hence providing a better and more stable solution [1].

The simulated ADC was then tested using test signals to show the results of the compensation scheme. The calibration test frequencies were similar to the calibration frequencies but twice as many were used. The test frequencies were sinusoidal, starting at 318 kHz and stepped in 91 kHz intervals up to 2.047 MHz. Test signals had an amplitude of -2 dBFS. The converter's performance was recorded in SFDR measurements and is shown in Figure 3.3. There are 4 curves in this plot showing various compensation schemes. The 'Uncompensated' curve shows the calculated SFDR values using data from the ADC without any compensation. 'Code and Slope' curve represents compensation using only the slope error basis functions when solving for $\vec{\alpha}$. It can be seen from the uncompensated curve that the performance starts to decrease in the middle of the band due to slope error. This error is picked up by the 'Code and Slope' curve in the second half of the Nyquist band, where the slope dependent error is dominant. The 'Code and Bits' curve represents compensation using only the basis functions that use the intermediate stage outputs. From the plot, it is evident that the compensation is most effective in the beginning of the Nyquist band where the sub-DAC error is dominant. Note that the upper SFDR limit for an ideal ADC would be determined by the noise floor because there will not be any harmonics in an ideal conversion. The maximum SFDR value in the plot shown in Figure 3.3 is not typical of real 10-bit converters as

the simulation model only included the common error mechanisms in an ADC. Real converters include various other non-linear errors that can be due to process and physical design issues.

Plots shown in Figure 3.4 are an array of output spectra for a single test frequency. Figure 3.4(b) shows the output spectrum with only code/slope compensation. Note that most of the low power harmonics are still present, but the 2^{nd} harmonic is untouched by this scheme. This harmonic is not introduced by a slope dependent error mechanism. Figure 3.4(c) shows the output spectrum with compensation using the intermediate stage outputs. Notice that the 2^{nd} harmonic is reduced; this shows that the harmonic was introduced by sub-DAC or gain error. The final plot, Figure 3.4(d), shows a fully compensated ADC. The distortion introduced by S/H and sub-DAC have been compensated for.

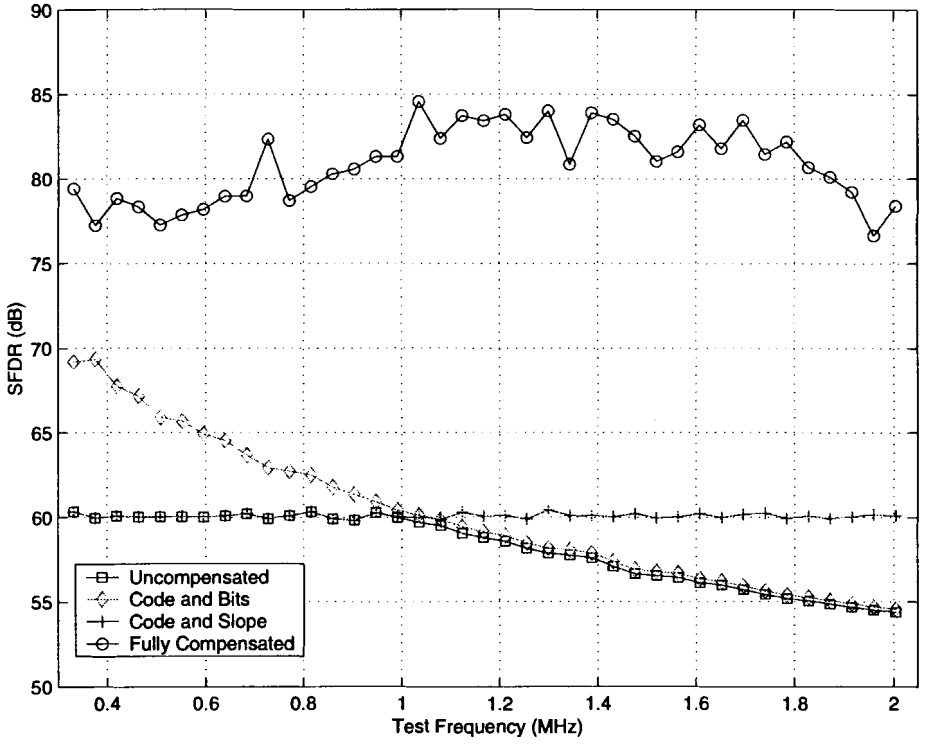
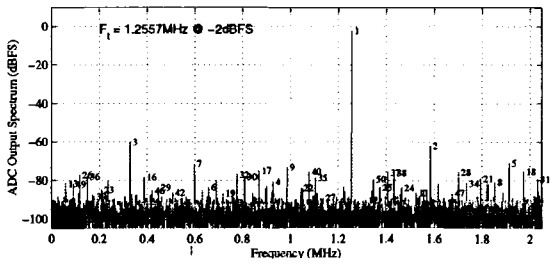
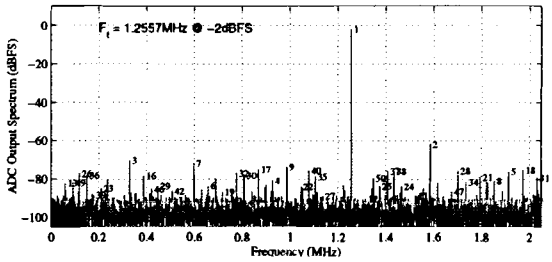


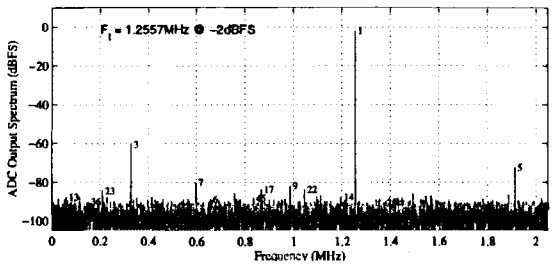
Figure 3.3: SFDR performance for Simulated ADC using proposed scheme.



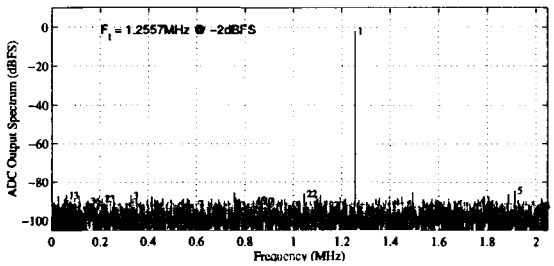
(a) Uncompensated output spectrum.



(b) Output spectrum with slope compensation.



(c) Output spectrum with intermediate stage bits compensation.



(d) Output spectrum with full compensation.

Figure 3.4: Simulated ADC output spectra illustrating different compensation schemes.

CHAPTER 4

Hardware Implementation

Previous chapters outlined the compensation scheme and its methods of implementation in software. To demonstrate the technique in hardware a 10-bit 5 MHz pipeline ADC was designed and fabricated. Intermediate sub-ADC outputs for the first three stages were made available to aid in the compensation. This chapter summarizes the design and performance of the converter with and without the proposed compensation scheme. The ADC was designed and fabricated in 0.5 μm CMOS technology in conjunction with a VLSI design course.

4.1 A 10-bit 5 MHz Pipeline ADC

The ADC is implemented using a nine stage 1.5 bits/stage pipeline architecture. The design is based on switched-capacitor circuitry adapted from [19]. The converter accepts -1 to 1 volt (2.5V common mode offset) fully differential input. The stage outputs are digitally corrected to obtain a 10-bit digital output. The device was fabricated through MOSIS in AMI's C5N process technology. The AMIC5N process is a single n-well, single poly and triple metal process. 14-masks were used in the final layout. Poly capacitors were used for all capacitors including the sampling capacitors in the design. The 0.5 μm minimum feature size is physically designed on a 0.15 μm grid. This constrains the gate length to a minimum of 0.6 μm. The die size is 1.5 $\times$ 1.5 mm and is packaged in a 40 pin DIP. Figure 4.1 shows a die photograph of the final chip and Figure 4.2 provides a floorplan of the die. A detailed description of this ADC can be found in [14].

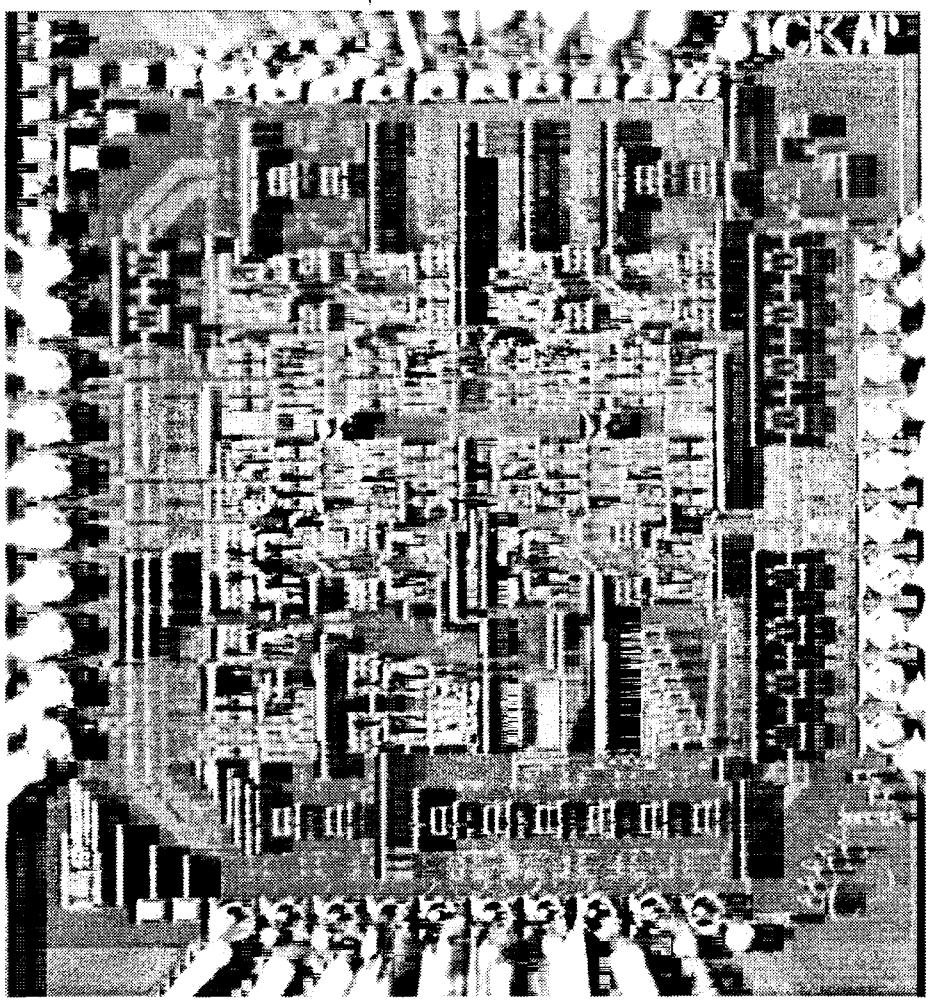


Figure 4.1: Digital image of the final packaged die.

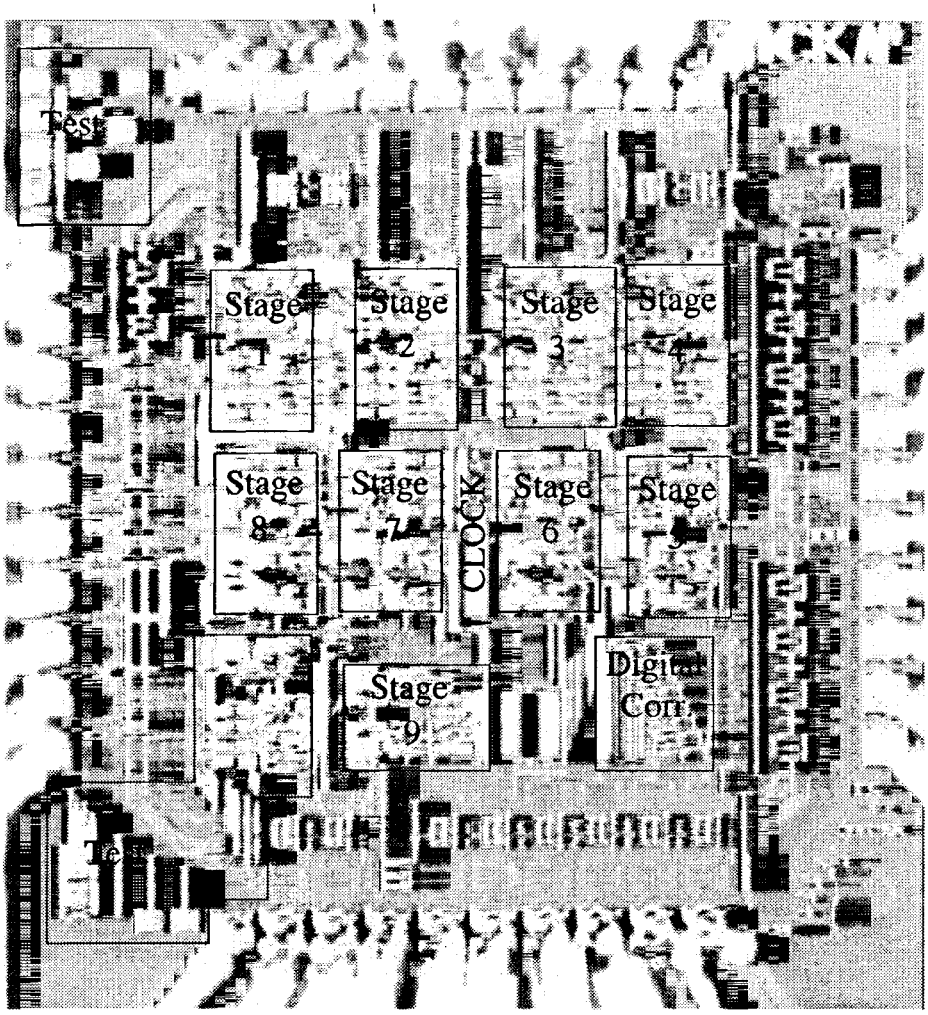


Figure 4.2: Floorplan of the final packaged die.

4.2 Design Overview

The 1.5 bits/stage architecture was used in the design of this ADC. Outputs from the nine stages are routed through a shift register before being digitally corrected. Each stage consists of an operational transconductance amplifier (OTA), differential comparators (sub-ADC) and a sub-DAC. The digital correction is a series of logic gates used to implement the addition operation discussed in Chapter 2. A shift register is used to keep the stage outputs in memory until the last stage completes its conversion before being corrected. The intermediate stage outputs from the first four stages are routed off-chip before they enter the shift register. Hence, when the chip is tested, the bits need to be realigned in software after the data is acquired. Due to the nature of the pipeline architecture, there is a six clock cycle delay at startup before the first output sample is valid. This switched-capacitor design also incorporates a fully differential concept for all signal paths.

4.2.1 Clock Description

This design relies on a two phase non-overlapping clock for operation. The two phases, ϕ_1 and ϕ_2, have a 180^o phase shift and a delay between the clock transitions. Since a digital word is available for output every clock cycle (after the initial inherent delay) and alternate stages are clocked by alternate clock phases, the components in a stage must settle within half the period of the main clock. For example: if the main clock is running at 5 MHz, then the longest time for any component to settle would be $\frac{200}{2} = 100$ ns. To implement bottom-plate switching [19], additional clock signals ϕ_1' and ϕ_2' are created which turn off slightly before ϕ_1 and ϕ_2 respectively. Figure 4.3 illustrates the four clock signals used in the design. The figure shows two timing requirements: the non-overlap period t_{nov} and the lag time between transitions for ϕ_k and ϕ_k', t_{lag}.

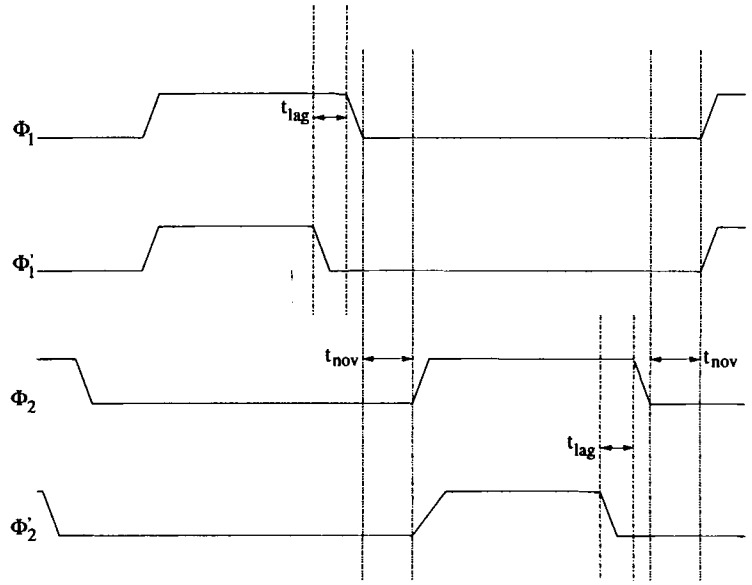


Figure 4.3: 2-Phase Non-Overlapping Clock Timing.

4.2.2 Pipeline Stages

The pipeline stages are divided into Odd and Even stages. The stages are identical except that the clock phases are reversed. The following stage description is based on an Odd Stage. For an Odd Stage, ϕ_1 is used to sample and quantize the input signal. During ϕ_2, the gainstage generates the residual to be passed to the next stage. The next (Even) stage is sampling its input during this second clock phase ϕ_2.

Figure 4.4 shows the architecture of a single stage of the converter. The sub-ADC for the 1.5 bits/stage converter consists of two differential comparators with thresholds placed at $\pm\frac{V_R}{4}$. Figure 4.5 shows the structure of one of these comparators. In this design, roles of the S/H and gain stage are implemented in the same block, shown in Figure 4.6. Figure 4.7 shows the circuit for the sub-DAC.

For an Odd Stage the sub-ADC (Figure 4.5) capacitors are charged to V_{ref}^+ and V_{ref}^- during ϕ_2. V_{ref}^+ and V_{ref}^- are set at 3 and 2 V respectively in this design and are reversed in the second comparator. When the clock transitions and ϕ_1 becomes active, the input signal is switched into the sub-ADC. During ϕ_1 and ϕ_1', the ADC output is

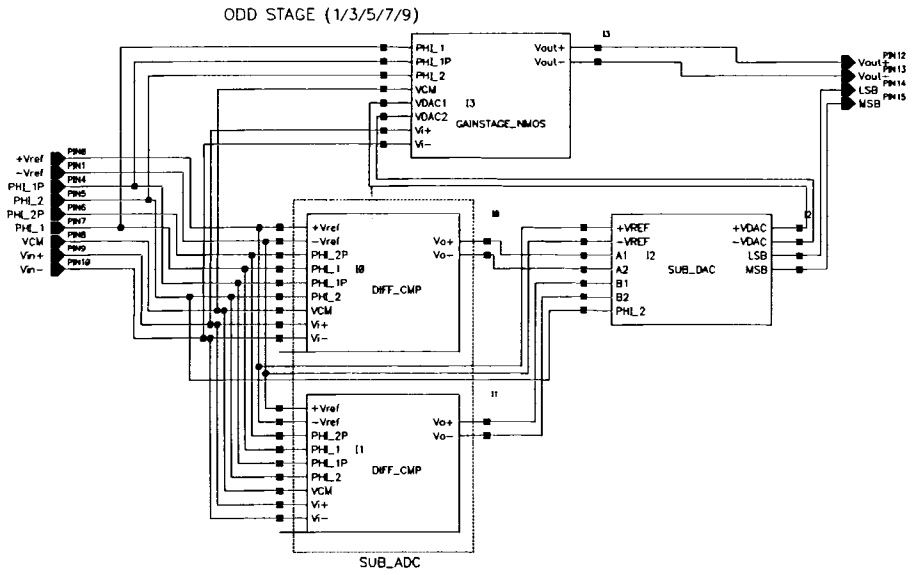


Figure 4.4: Single stage block diagram in 10-bit ADC.

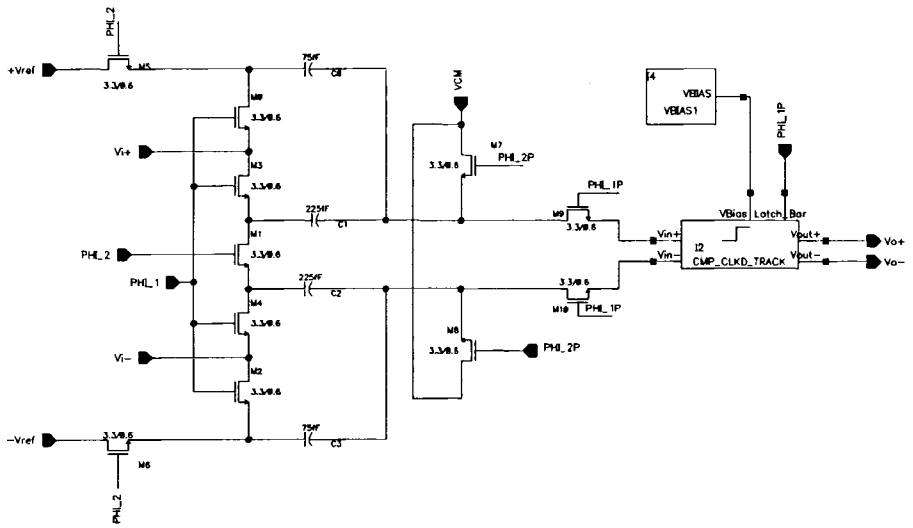


Figure 4.5: Differential Comparator implementation.

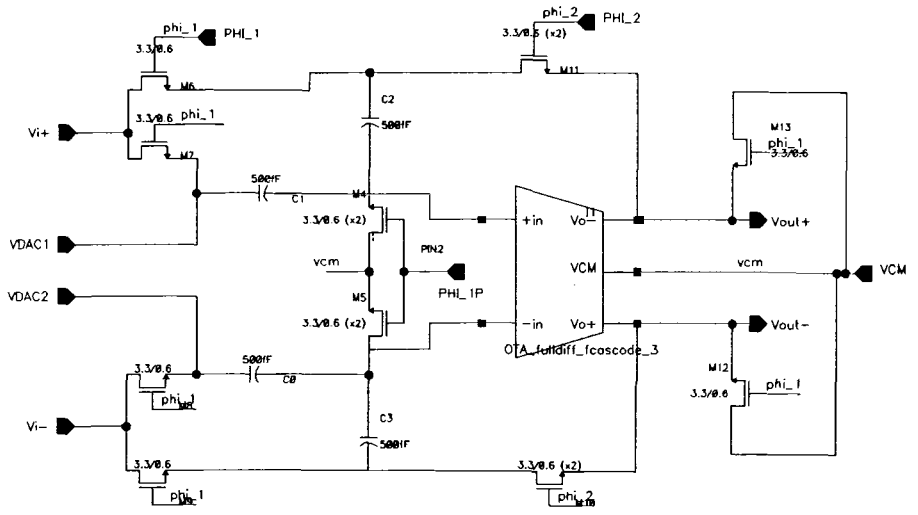


Figure 4.6: S/H and gain portions of the stage.

generated and latched (Figure 4.5). During this same clock phase, the input signal to the stage is sampled across the the gainstage S/H capacitors (Figure 4.6). The output and input pins of the OTA are tied to V_{cm} (2.5 V) during this period.

During ϕ_2 and ϕ_2', the sub-DAC is active and calculates the corresponding digital and analog outputs. The digital output is sent to the shift register and held for digital correction. The analog outputs are routed into the gainstage as V_{dac1} and V_{dac2}. In the gainstage, V_{dac1} and V_{dac2} are forced onto the capacitor network across the OTA to provide the appropriate residual for the next stage. In this next Even Stage, the capacitors are in a S/H mode to receive the residue from the Odd Stage.

Note that the last stage is different from all other stages in construction and also in timing. The last stage is switched with two clock phases and has a third threshold that is compared to its input sample. According to this design, the last stage will charge the capacitors with V_{ref}^{+} and V_{ref}^{-} during ϕ_1 and ϕ_1'. During ϕ_2, the outputs from each

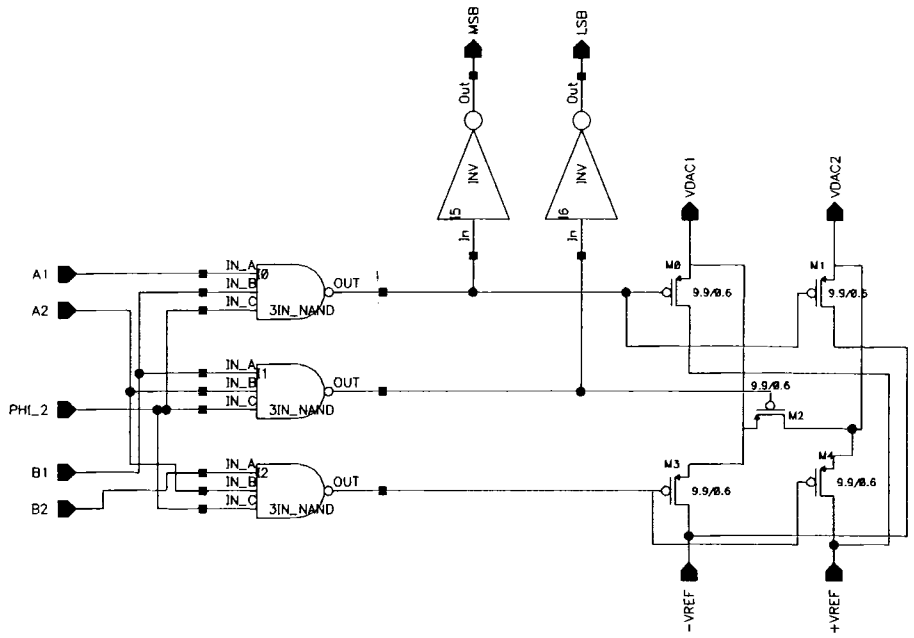


Figure 4.7: Sub-DAC circuit implementation.

differential comparator is switched through logic circuitry to provide the digital output for this stage. A more comprehensive discussion about this ADC can be found in [14].

4.3 Performance

The ADC was tested using the ADC/DAC test bed in the Communication Devices Applications Laboratory, at the University of Maine in Orono. Static power tests ($F_s = 5$ MHz and $F_t = 0$) confirmed power consumption predictions from earlier simulations, which was approximately 60 mW. Some functionality tests were carried out to characterize the ADC. These tests include SFDR, ENOB, SINAD, INL and DNL tests. INL and DNL are nonlinearity tests used in the industry to rate ADC performance. The ADC is functional at 5 MHz, however for this research, the sampling rate was kept at 4.096 MHz due to some issues with clock alignment in taking the data.

Figure 4.8 shows the non-linearity measurements from the ADC. The test used 16K samples and F_t was set at 1 MHz. Figure 4.9 shows a spectral plot of the ADC

output for this test frequency. Other test measurement plots are incorporated with the compensation plots in the next section.

4.4 Calibration and Compensation

The ADC was calibrated using the scheme outlined in Section 3.4. The same 20 trajectories were used for calibration. Initially, the slope and intermediate stage outputs were used to compensate the converter. This was successful but seemed to have a few dead zones at which the compensation results were obviously lower than those in the rest of the test frequencies across the band. A closer look at the schematic revealed that the S/H had significant dependence on the previous sample.

A previous sample error means that the charge on the sampling capacitors has some residue left over from the previous sample that may be affecting the new input sample. Figure 4.10(a) shows the input sampling capacitors located in the stage schematic shown in Figure 4.6. In this diagram, during the first clock phase, the input is sampled across both capacitors, hence the voltage at the input is $V_{in(i)}$. Note that for this analysis, only one side of the differential path is examined because the opposite side is a negative image. Figure 4.10(b) shows the second clock phase, where the difference between the input sample and DAC output is generated by the OTA. During this second clock phase, the voltage at the input can be written as $2(V_{dac})$ and the output is $2(V_{in(i)} - V_{dac})$. At the start of the next clock cycle (Figure 4.10(c)), both the capacitors are returned to the previous $V_{in(i)}$ sample and begin changing to the new sample, $V_{in(i+1)}$. This analysis suggests that the previous sample may have an effect on the new sample.

To compensate for the previous sample error, a new set of basis functions are defined and added into the compensation scheme. The previous sample error is characterized using Gaussian basis functions, similar to S/H error characterization. Defining $f_p(X_{k-1}, X_k - X_{k-1})$ as the previous sample error for the k^{th} sample which depends on

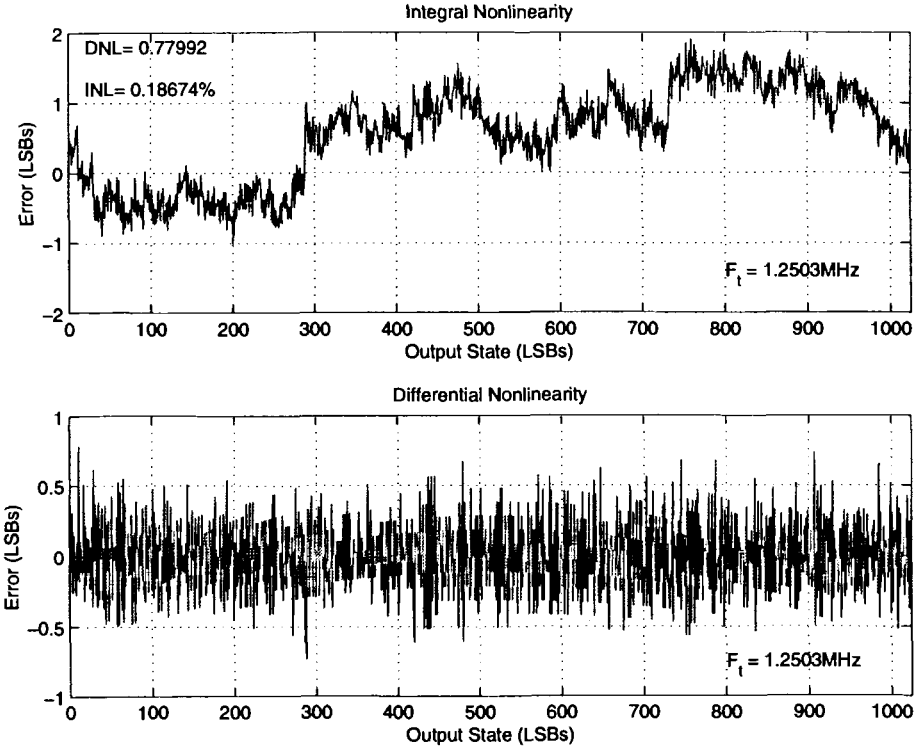


Figure 4.8: INL and DNL plots for real ADC.

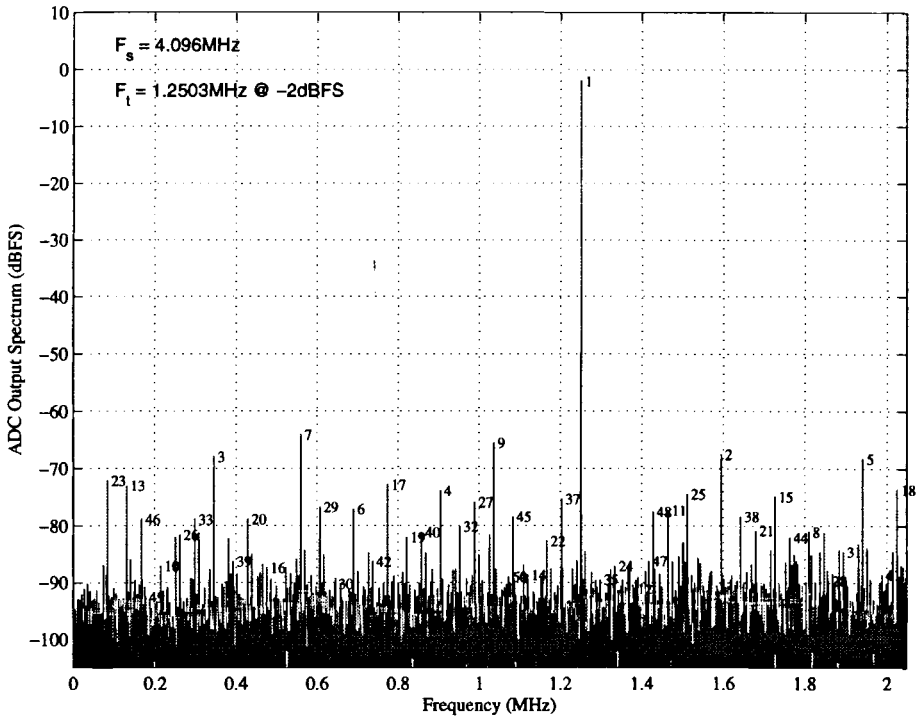


Figure 4.9: Output spectral plot of ADC for $F_t = 1.2$ MHz.

the previous sample and the difference between the current and previous sample, then

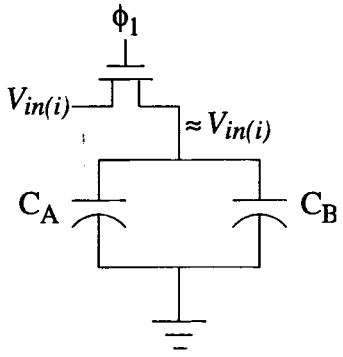
$$f_p(X_{k-1}, X_k - X_{k-1}) = g(X_{k-1})(X_k - X_{k-1}). \tag{4.1}$$

The unknown function $g(X_{k-1})$ can be written as unknown Gaussian functions using the form

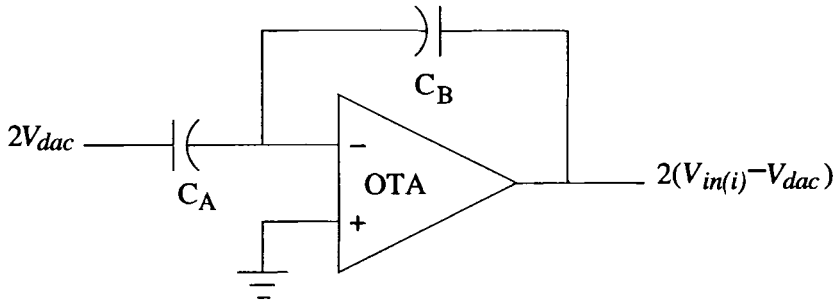
$$g(X_{k-1}) \approx g\left(\frac{x_{k-1} - x_{off}}{S}\right) = \sum_{i=1}^{L} \alpha_i e^{-\frac{1}{2}\left(\frac{x_{k-1}-p_i}{\sigma_p}\right)^2}, \tag{4.2}$$

where p_i are centers for these Gaussian functions, σ_p is the standard deviation and L is the total number of functions. The basis functions for the previous sample dependent error for each sample can defined as

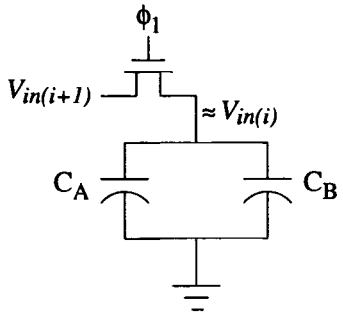
$$b_i(x_{k-1}, x_k - x_{k-1}) = \left(\frac{x_k - x_{k-1}}{S}\right) e^{-\frac{1}{2}\left(\frac{x_{k-1}-p_i}{\sigma_p}\right)^2} \quad \text{for } i = 1, 2, ..., L. \tag{4.3}$$



(a) End of ϕ_1.



(b) End of ϕ_2.



(c) Start of ϕ_1.

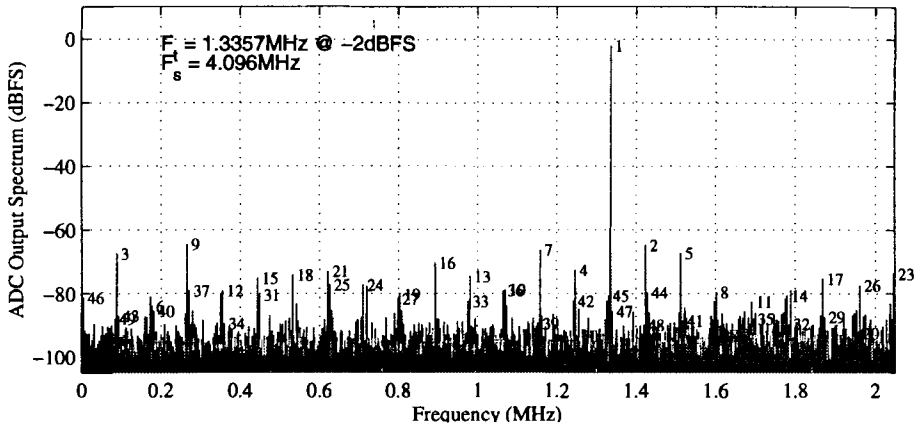
Figure 4.10: Sampling capacitor status for different clock phases showing previous sample dependence.

For compensating this ADC the centers, p_i values were chosen evenly over the output codes, $p_i = [0, 128, 256, 384, ..., 1024]$ and $\sigma_p = 128$ to provide sufficient overlap between functions. Since there were 9 centers, 9 basis functions were added to the existing 27 from Section 3.5. Hence 36 basis functions were used for calibration and compensation of the real ADC.

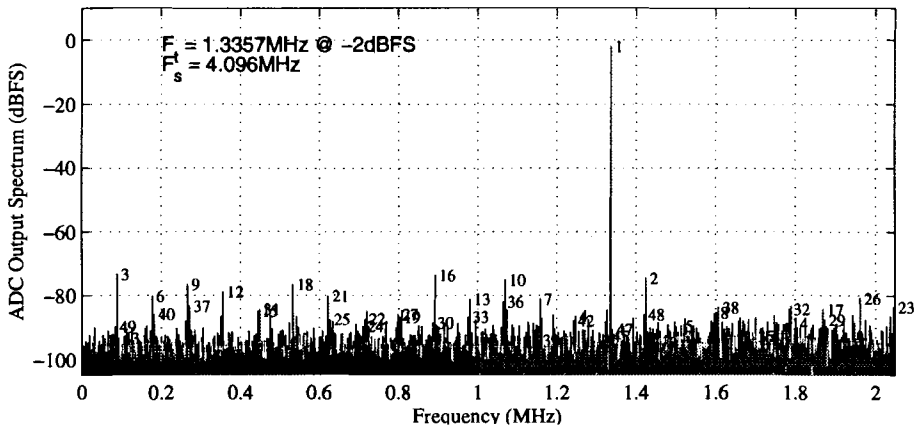
The ADC was calibrated using identical conditions listed in Section 3.5. All sample sets had 2^{14} samples. $\vec{\alpha}$ was calculated for various combinations of basis functions. A full compensation set has 36 basis functions, 9 for slope, 9 for previous sample and 18 for intermediate stage outputs.

Figure 4.11 shows the comparison between an uncompensated and compensated output spectrum for a single test frequency. Notice that the large harmonics are significantly reduced (7 to 12 dB) once the compensation is applied. Figure 4.12 shows the comparison between individual basis function sets used for the same test frequency. The ADC was compensated using the same $\vec{\alpha}$ values across the Nyquist band. Figure 4.13 shows the SFDR measurement for different compensation schemes. Note that all tests use signals with amplitudes of -2 dBFS. The uncompensated curve is also included in the plot. On average, the full set of basis functions seem to increase the SFDR by about 8.6 dB and as much 12.5 dB at some points. Individual error mechanism basis functions only work when that particular error is dominant. To get a uniform gain in performance throughout the band, all error mechanisms must be taken into consideration.

An increase in SINAD and effective number of bits is also observed after compensation. Figure 4.14 compares the uncompensated vs. compensated SINAD performance for the real ADC. An average of 3.15 dB increase in SINAD which translates to an ENOB increase of 0.53 bits was observed. These results are compared to simulation and discussed in Chapter 5. These are promising results and show a significant gain in using the intermediate stage outputs for compensation.

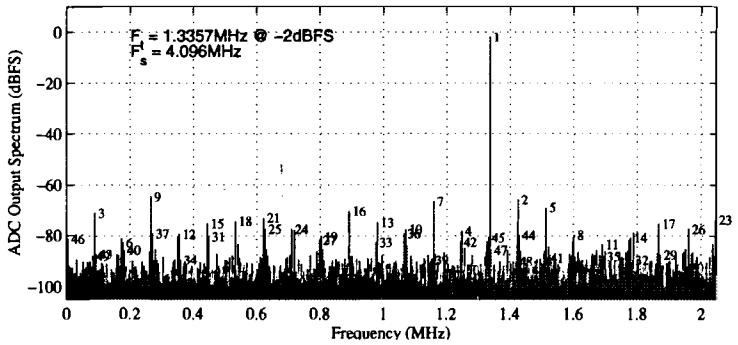


(a) Uncompensated output spectrum.

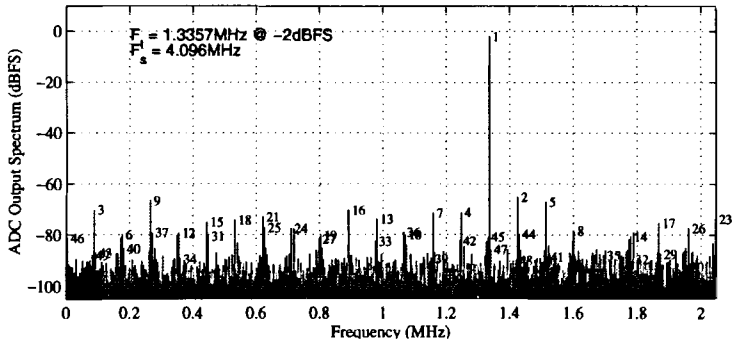


(b) Compensated output spectrum.

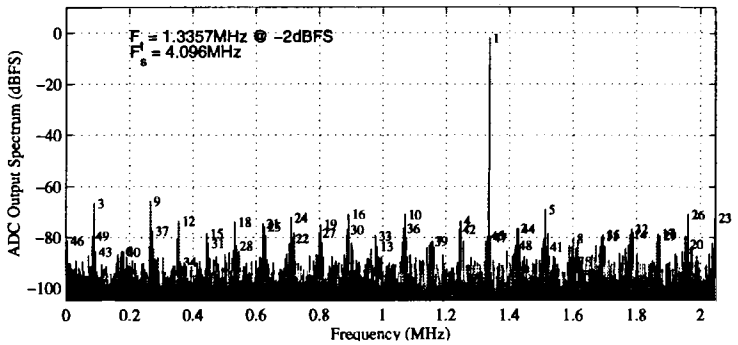
Figure 4.11: Uncompensated and compensated output spectrum for a single test signal using a 10-bit ADC.



(a) Previous sample basis functions only.



(b) Slope dependence basis functions only.



(c) Intermediate stage outputs basis functions only.

Figure 4.12: Compensated output spectrum for 10-bit ADC using different basis function sets.

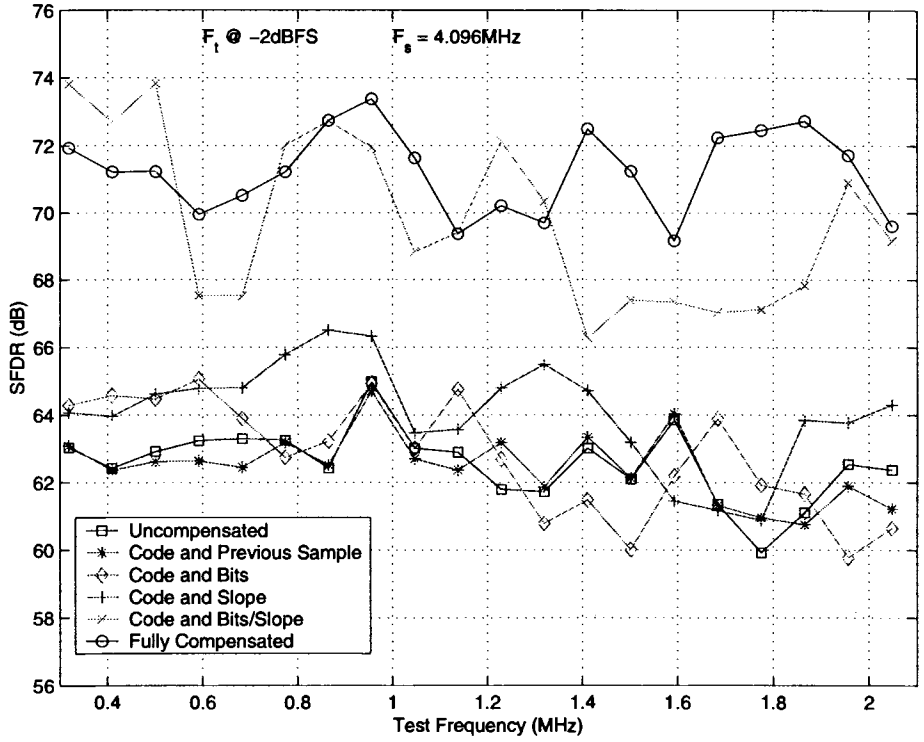


Figure 4.13: SFDR performance across the Nyquist band for a 10-bit ADC.

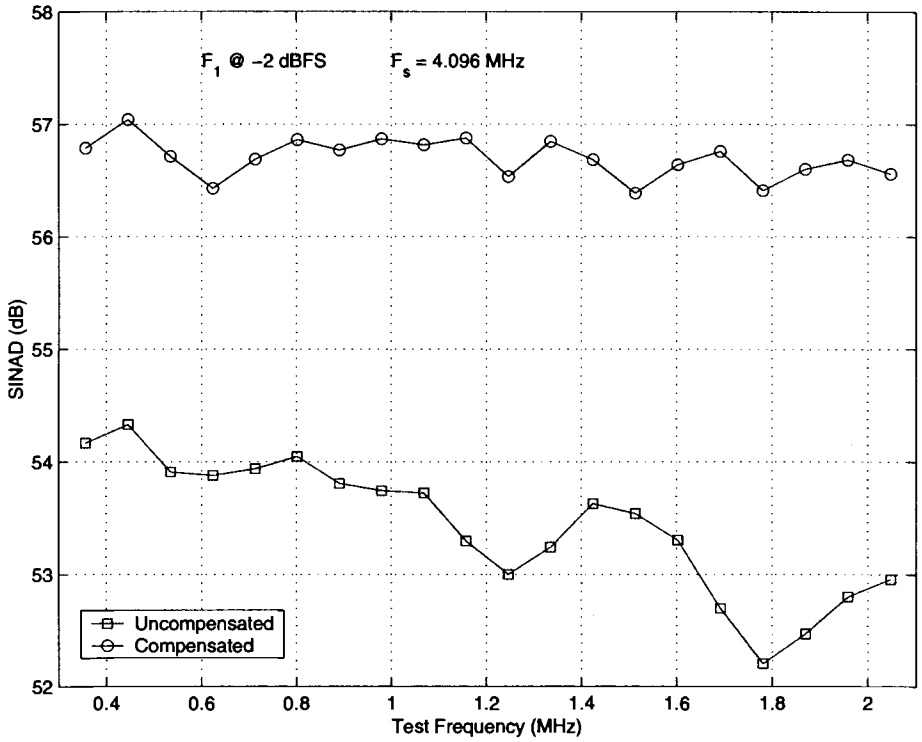


Figure 4.14: SINAD performance across the Nyquist band for a 10-bit ADC.

CHAPTER 5

Conclusion

This thesis has discussed, developed and implemented error detection and compensation schemes for pipeline ADCs. The idea of using intermediate stage bits for compensation has proved effective in simulations and also on a real ADC. Traditional dynamic compensation techniques have shown to be only marginally effective on pipeline converters. This thesis has successfully demonstrated that there are significant gains in performance by using the intermediate stage outputs for calibration and compensation.

Error mechanisms unique to pipeline ADCs were investigated for the general case and also the 1.5 bits/stage topology. The error mechanisms in the S/H, sub-ADC, sub-DAC and gain portion of the stage were given functional representations. Previous calibration work [5] was modified with new basis functions to efficiently model error characteristics in the pipeline architecture. The calibration and compensation scheme was tested on a simulated ADC with error mechanisms of interest. The results verified that the intermediate stage outputs are needed to characterize errors.

The developed calibration and compensation scheme was implemented on a real 10-bit 5 MHz ADC. The ADC was designed to provide the intermediate stage outputs for the first three stages to demonstrate the scheme presented in this thesis. The ADC was calibrated and compensated using both schemes, with and without the intermediate stage bits, to show the effectiveness of this new method. Without the intermediate stage outputs (traditional scheme), the average SFDR gain was approximately 1.5 dB with a maximum of 4.1 dB at certain test frequencies. Using the intermediate stage outputs provides an average gain of 8.6 dB of SFDR and 0.53 bits of ENOB across the Nyquist band. There are test frequencies that showed almost 12.5 dB of improvement. These are significant gains in the dynamic range of the converter that increase the performance of the 10-bit ADC to dynamic ranges typical of 12-bit ADCs. An increase in SINAD

performance was also observed. SINAD was increased in average by 3.2 dB in the first Nyquist band.

Clearly the hardware results indicate that the intermediate stage outputs are important to characterize error mechanisms of the sub-DAC and gain portions of a pipeline stage. The results from simulation and hardware are promising but at the same instant, demonstrate a need for further work to implement the schemes in hardware and not require the use of expensive test equipment. Current results do not include tests to show performance variation with temperature, power supply instability, aging etc. Another key test would be to show performance of the ADC when operating outside the first Nyquist band. A feasibility study on implementing compensation in production (economically and technically) can be useful in determining the effectiveness of this scheme for commercial devices. Economic suitability may depend upon the use of a single error table to compensate several devices. Further work is needed to implement this scheme in hardware or possibly as a Built-In Self Test (BIST) system. A self-calibrating ADC would greatly benefit the wide-bandwidth receiver designs.

REFERENCES

[1] D. Hummels, "Linearization of ADCs and DACs for all-digital wide-bandwidth receivers," in *IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation and 4th Workshop on ADC Modeling and Testing*, (Bordeaux, France), Sept. 1999.

[2] F. Irons and T. Rebold, "Characterization of high frequency analog-to-digital converters for spectral applications," tech. rep., Project Report AST-2, M.I.T. Lincoln Laboratory, Lexington, Massachusetts, Nov. 1986.

[3] S. Kennedy, "Improved ADC phase plane error compensation," Master's thesis, University of Maine, Dept. of Electrical and Computer Eng., Orono, Maine, May 1992.

[4] F. Irons, D. Hummels, and S. Kennedy, "Improved error compensation for analog-to-digital converters," *IEEE Trans. Circuits and Systems*, vol. 38, pp. 958–961, June 1991.

[5] D. Hummels, F. Irons, R. Cook, and I. Papantonopoulos, "Characterization of ADCs using a non-iterative procedure," in *Proceedings of IEEE International Symp. on Circuits and Systems*, (London, England), May 1994.

[6] D. Hummels, F. Irons, and J. Larrabee, "ADC compensation using sinewave histogram methods.," in *Proceedings of IEEE Instrum. and Meas. Technology Conference*, (Ottawa, Canada), pp. 628–631, May 1997.

[7] G. Kelso, "Fast calibration of analog to digital converters," Master's thesis, University of Maine, Dept. of Electrical and Computer Eng., Orono, Maine, Aug. 1999.

[8] G. Kelso, D. Hummels, and F. Irons, "Fast compensation of analog-to-digital converters," in *Proceedings of IEEE Instrum. and Meas. Technology Conference*, (Venice, Italy), May 1999.

[9] J. Tsimbinos and K. Lever, "Applications of higher-order statistics to modeling, identification and cancellation of nonlinear distortion in high-speed samplers and analog-to-digital converters using the Volterra and Wiener methods," in *IEEE Signal Processing Workshop on Higher-Order Statistics*, (South Lake Tahoe, CA, USA), June 1993.

[10] S. Acunto, P. Arpaia, D. Hummels, and F. Irons, "A new bi-dimensional histogram for the dynamic characterization of ADCs," in *Proceedings of IEEE Instrum. and Meas. Technology Conference*, (Budapest, Hungary), Nov. 2001.

[11] S. Acunto, P. Arpaia, D. Hummels, and F. Irons, "A new bi-dimensional histogram for the dynamic characterization of ADCs," *Submitted to IEEE Trans. on Instrum. and Meas.*, vol. IM-3395, May 2001.

[12] L. Lin, "Design techniques for parallel pipelined ADC," Master's thesis, University of California, Berkeley, California, May 1996.

[13] Texas Intruments Inc., Dallas(TX),USA, *12-bit 40 MSPS IF Sampling Communications Analog-to-Digital Converter*, Jan. 2001. Datasheet - SLAS279D.

[14] K. Sockalingam and R. Thibodeau, *Design and Layout of a 10bit 5MHz Pipeline A/D Converter.* University of Maine, Dept. of Electrical and Computer Eng., Orono, Maine, Jan. 2002. ECE547 VLSI Design Report.

[15] TC-10, "Standard for terminology and test methods for analog-to-digital converters." IEEE Instrum. and Meas. Society, Std. 1241 (draft vs 101700), Oct. 2000.

[16] D. Hummels, W. Ahmed, and F. Irons, "Measurement of random sample time jitter for ADCs," in *Proceedings of IEEE International Symp. on Circuits and Systems*, (Seattle), pp. 708–711, May 1995.

[17] E. Swanson, "Investigation of shifted bit dependent error in analog-to-digital converters," Master's thesis, University of Maine, Dept. of Electrical and Computer Eng., Orono, Maine, May 1998.

[18] L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis.* Reading, Massachusetts: Addison-Wesley, 1991.

[19] A. Abo, *Design for Reliability of Low-voltage Switched-capacitor Circuits.* PhD thesis, University of California, Berkeley, California, May 1999.

BIOGRAPHY OF THE AUTHOR

Kannan Sockalingam was born in Kuala Lumpur, Malaysia on October 15, 1978. He received his high school education from St. Johns Institution in Kuala Lumpur in 1995. He entered the University of Maine in 1996 and obtained his Bachelor of Science degree in Electrical Engineering in 2000. During his undergraduate education, he completed Co-op assignments at Fairchild Semiconductor (S. Portland, ME.) and Texas Instruments (Dallas, TX.). He worked on wafer and package level reliability while at Fairchild Semiconductor. In Texas Instruments he worked on ADC/DAC characterization and wafer level testing.

In January 2001, he was enrolled for graduate study in Electrical Engineering at the University of Maine and served as a Research Assistant in the Communications Devices Applications Laboratory. His current research interests include communications and signal processing.

He is a member of IEEE, Tau Beta Pi, Eta Kappa Nu, and his interests include emergency medicine and traveling. Kannan is a candidate for the Master of Science degree in Electrical Engineering from The University of Maine in August, 2002.