1-1-1997

An accuracy bootstrapped digitally self calibrated non-radix-2 analog-to-digital converter

Tapas Ray
Iowa State University

An accuracy bootstrapped digitally self calibrated non-radix-2 analog-to-digital converter

by

Tapas Ray

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Electrial Engineering
Major Professor: Marwan M. Hassoun and William C. Black, Jr.

Iowa State University
Ames, Iowa
1997

Graduate College
Iowa State University

This is to certify that the Master's thesis of

Tapas Ray

has met the thesis requirements of Iowa State University

Signatures have been redacted for privacy

For the Graduate College

To my parents

who have made my ambitions their dreams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

No words can completely express my appreciation for my research advisor Dr. Marwan M. Hassoun. I am grateful to him for his guidance and support throughout my stay at Iowa State University. He developed an excellent rapport with all his students which created a very friendly and congenial atmosphere for research. He has been a good friend to me and to us all. Because of him I have learned to believe in others. I am forever indebted to him. I wish to thank Dr. William C. Black who helped me understand data converters better and sat with me for some memorable discussions. A special thanks is also due to John Moulder at Center for Non Destructive Evaluation, Iowa State University who gave me an assistantship for three semesters without which all this wouldn't have been possible. I also wish to thank Dr. Gurpur Prabhu for being in my committee. I owe my knowledge of computer architecture to him.

At ISU, I came in touch with quite a few professors and from each of them I have learned some thing different about life. Dr. William C. Black's modesty makes me wonder about my own place in the universe. His brilliance and odd working hours were a boon to students especially the night before the deadline. Dr. Sachin S. Sapatnekar taught us just not about VLSI circuits but also how to teach. His jokes in the classroom and last minute reprieves on the homework deadlines endeared him to all of us. Dr. Randall Geiger's courses made me appreciate analog design and developed in me a sense to feel the circuit. His laughter still rings in my ears. I consider myself privileged to have worked with such great people and to be able to say one day that yes, I knew them.

During my stay at Ames I made many friends who helped me whenever they could. Each one of them had certain qualities that I came to admire. Anindya Roy's brilliance, Naresh Kumar's discipline, Ankur Gupta's managerial ability, Jayant Kumar's helpful nature, Subramanyam Narayan's focussed goals, Naresh Maheshwari's ability to question everything and anything, Sabita Pilli's hardworking nature, Pavan Bassetty's nonchalance, Lakshmikant Bhupathi's calm brilliance, Venkata Navin's knowledge about analog circuits and easy going manner, Jatan Shah's ability to finish homework well before deadline, Manoj Sahu's company, Pradipta Sarkar's cheerfulness, Prasoon Surti's interest in things

outside the course of study, Jeff Shi's studiousness, Rajesh Parekh's good manners, Sunil Mishra's cooking abilities and last but not least Soon Seng Lau's knowledge of LaTeX. I thank them all.

I would like to thank, my colleague and office mate, Venkata Navin with whom I had many fruitful discussions, on data converters in particular and life in general. It was great to have worked with him. Thanks are also due to Hon Chi Ng for his work on the plotting capabilities of the simulator. I also appreciate the help of my officemate, Suchi, in finishing up the final paperwork.

A special thanks is also due to Wayne Dietrich and Defense Systems Electronics Group at Texas Instruments, Dallas for their financial and technical support to the project.

I thank all those people whom I could not name here but have helped me in numerous ways. Lastly, I would like to thank my parents for encouraging me to reach higher and being content in seeing the world through my eyes. To them I dedicate my thesis.

ABSTRACT

As data conversion interfaces are designed for higher precisions and higher speed, the nonidealities that accompany monolithic devices become more problematic. In this thesis an improved algorithm is presented to correct the nonidealities in a non-radix-2 based pipeline. The pipeline analog-to digital converter (ADC) based on non-radix-2 architecture has the advantage of a simple design and algorithm, but a higher number of stages are required to obtain the required linearity. To improve the linearity of the non-radix-2 pipeline ADC the proposed algorithm uses digital self-calibration technique along with accuracy bootstrapping. The digital self-calibration technique presented in [1], for non-radix-2 converters suffers from the drawback that it does not utilize the whole pipeline for calibration. Accuracy bootstrapping [2] allows the utilization of all the stages of the pipeline for calibration, but it is meant for radix-2 converters. The proposed algorithm, in this thesis, works in two steps. During the calibration of stages with integer gain, all the stages are switched to integer gain, bootstrapped and calibrated. When calibrating non-integer gain stages, only stages upstream of the stage being calibrated are switched to integer gain, the rest being maintained at their nominal value, bootstrapped and calibrated. Since all the stages of the pipeline are used during calibration the results obtained are better by as much as 50% when compared to the digital self-calibration algorithm presented in [1], at little extra cost.

1 INTRODUCTION

An increasing demand for high performance Digital Signal Processing(DSP) system motivates an increasing demand for high-resolution, high-speed Analog-to-Digital Converters(ADCs). These ADCs are used in a wide variety of environments, including industrial, communication, medical and aerospace. The use of ADCs in such systems permits sophisticated DSP algorithms to be utilized which alleviates complexity in the analog signal processing required otherwise.

Traditional high-resolution, high-speed ADCs rely on expensive, hybrid or discrete implementation and thus are not amenable to low cost manufacturing. The use of monolithic Integrated Circuit techniques has been repeatedly demonstrated in the digital and analog circuit domains to be economical for synthesizing complex systems. Thus, monolithic integration of high-accuracy, high-speed ADCs with other systems requires that the ADC architecture be compatible with properties of the host system. However the desire to integrate high-speed, high resolution ADCs in low cost, IC processes presents algorithmic, circuit and process challenges.

Pipeline analog-to-digital converters present advantages compared to flash or successive approximation techniques because potentially high resolution and high speed can be achieved at the same time . A 1-bit per stage design is particularly desirable because each stage is very simple and fast. One advantage of the pipeline ADC architecture is that it exhibits linear growth in hardware, compared to exponential growth with flash ADCs, for increasing resolution. In addition, pipeline ADCs require fewer clock cycles than successive approximation or oversampled ADCs, which is important for high-speed. However, pipeline ADCs are capable of achieving only 8-10 bits of linearity in most IC processes without the use of component trimming or self-calibration techniques.

This thesis covers the study and analysis of non-radix-2 pipeline Analog-to-Digital Converter architecture. With the demand for high-speed devices increasing at the explosive rate there is a need to look for alternative architectures which have the potential for such applications but have not yet been studied in detail. One such architecture was proposed in [1] which instead of having pipeline stages with integer gain , has a non-integer gain for majority of the stages. Since it's a pipelined converter

it needs to be calibrated to compensate for the errors in its components. The pipeline is calibrated only for the non-integer gain stages. In this thesis, an alternative algorithm for calibration has been proposed for the same architecture. The proposed algorithm uses digital self-calibration algorithm proposed in [1] along with accuracy bootstrapping algorithm [2] to use all the stages of the pipeline during calibration. Since all the stages are now being used, the calibration results are better compared to [1] and the calibrated ADC has a higher degree of linearity. Besides non-radix-2 architecture, in this thesis experiments on a simplified cell architecture for accuracy bootstrapping, which is an improved version of the architecture in [2], have been performed. The experiments were helpful in determining the exact method of calibration to be used for the new architecture.

This work was funded by the Defense Systems Electronics Group at Texas Instruments, Dallas, Texas. This thesis uses results presented in the thesis of Venkata Navin [25], which has been referenced at appropriate places. In particular, the mathematical foundation for accuracy bootstrapping and Global Matching algorithm for parallel pipelined converters was developed by Venkata Navin.

1.1 Nonlinearity Specifications

There is a wide variation in the literature in the definition of the nonlinearities, especially Integral Nonlinearity and Differential Nonlinearity . The definitions [19] given below are the ones used in the program and in the rest of the thesis.

Integral Nonlinearity: The maximum deviation of the actual transition points in an ADC's transfer curve from the straight line drawn between the end points i.e. first and last code transitions, is defined as the Integral Nonlinearity (INL).

Differential Nonlinearity: The maximum deviation in the output step size from the transition point is defined as the Differential Nonlinearity (DNL). DNL error describes the difference between two adjacent analog signal values compared to the step size (LSB) of a converter generated by transitions between adjacent pairs of digital code numbers over the full range of the converter. The DNL is zero if every transition to its neighbors equals 1 LSB.

In case of non-binary radix architecture, since the decision levels are not uniformly spaced due to the non-binary radix, a different mechanism, called digital output reduction, was developed in [33] which maintains the value of INL and DNL after correct calibration within 1 LSB. If the INL and DNL calculations are performed using the end points, values within 1 LSB cannot be achieved. A detailed explanation of digital output reduction method is given in Chapter 3.

INL and DNL are very significant indicators of the linearity of the ADC. The DNL reflects the

resolution of the ADC whereas the INL reflects the accuracy of the ADC.

Offset: Input amplifiers, output amplifiers, and comparators in practical circuits inherently have a built-in offset voltage and offset current. The offset is caused by the finite matching of components. The offset results in a non-zero input or output voltage, current or digital code although a zero signal is applied to the converter.

Global Gain: Due to the nonlinearities present in various components of the ADC, the resultant transfer characteristic of the ADC has a gain other than unity. The gain of the transfer characteristic is defined as the slope between the two end points. This slope is the global gain.

Signal to Noise Ratio (S/N): The most important dynamic specification of a converter is the signal-to-noise ratio. This signal-to-noise ratio depends on the resolution of the converter and automatically includes specifications of linearity, distortion, sampling time uncertainty, glitches noise and settling time. Over half the sampling frequency, this signal-to-noise ratio must be specified and should ideally follow the theoretical formula:

$$S/N_{max} = 6.02n + 1.76dB \tag{1.1}$$

This S/N ratio is calculated for a sine wave input with a maximum amplitude, and the ratio between the frequency of the sine wave, and the sampling frequency should be irrational.

Spurious Free Dynamic Range: When converters are used with large oversampling ratios or the spectral purity of the converter is important, an additional specification determining the ratio between the maximum signal component and the largest distortion component can be obtained. SFDR is defined as the ratio in decibels between the magnitude of the fundamental component and the magnitude of the largest harmonic or the inter-modulation product. It is an indication of the dynamic operating range of the ADC [25].

1.2 Analog-to-Digital Converter Architectures

1.2.1 Flash Converters

Conceptually the simplest and potentially the fastest, flash architectures employ parallelism and "distributed" sampling to achieve a high conversion speed. Figure 1.1 shows a block diagram of an m-bit flash ADC. The circuit consists of 2^m comparators, a resistor ladder comprising 2^m equal segments, and a decoder. The ladder subdivides the main reference into 2^m equally spaced voltages. For example, if the analog input is between V_j and V_{j+1}, comparators A_1 through A_j produce ONE's at their outputs

while the rest generate ZERO's. Consequently, the comparator outputs constitute a thermometer code, which is converted to binary by the decoder.

Full flash architectures in principle do not need explicit front-end sample-and-hold circuits and their performance is determined primarily by that of their constituent comparators. Since comparators do not require linear amplification and hence typically achieve much higher speeds than sample-and-hold amplifiers, flash ADCs can operate faster than those that demand front-end SHAs.

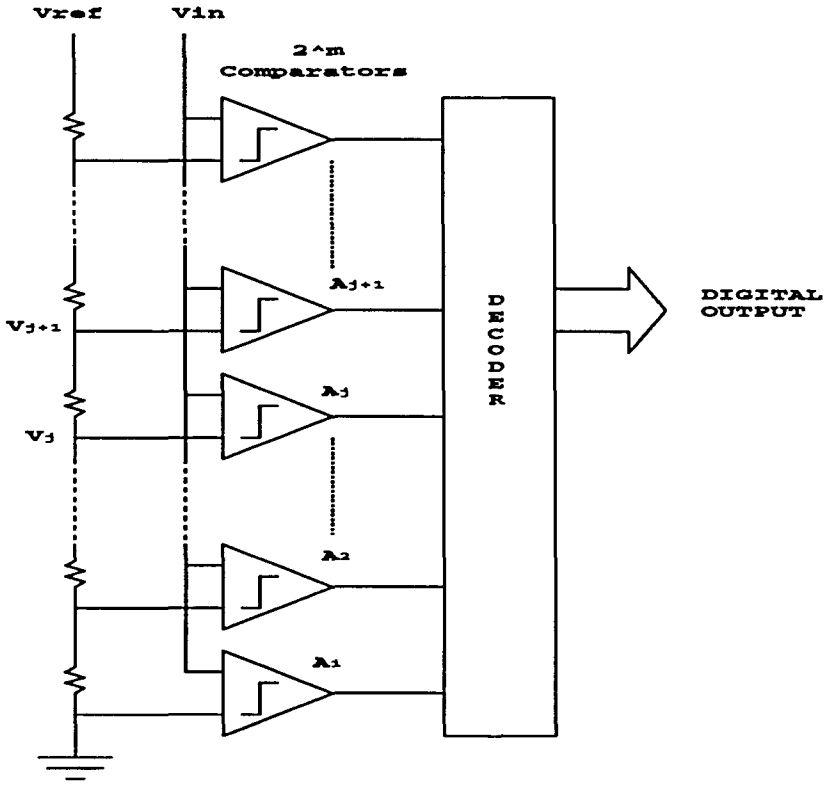

Figure 1.1 Block diagram of m-bit flash A/D converter

Despite these features, flash topologies suffer from a number of drawbacks due to massive parallelism or lack of a front-end sampling circuit. Since the number of comparators grows exponentially with the resolution, these ADCs require excessively large power and area for resolutions above 8 bits. Furthermore, the large number of comparators gives rise to problems such as dc and ac deviation of the reference voltages generated by the ladder, large nonlinear input capacitance and kickback noise at the analog input. In addition, the lack of a front-end sample and hold amplifier makes the converter susceptible to sparkles and slew-dependent sampling points [22].

1.2.2 Two-Step Architecture

The exponential growth of power, area and input capacitance of flash converters as a function of of resolution makes them impractical for resolutions above 8 bits, calling for other architectures that provide a more relaxed trade-off among these parameters. Two-step architectures trade speed for power, area, and input capacitance [8] [13].

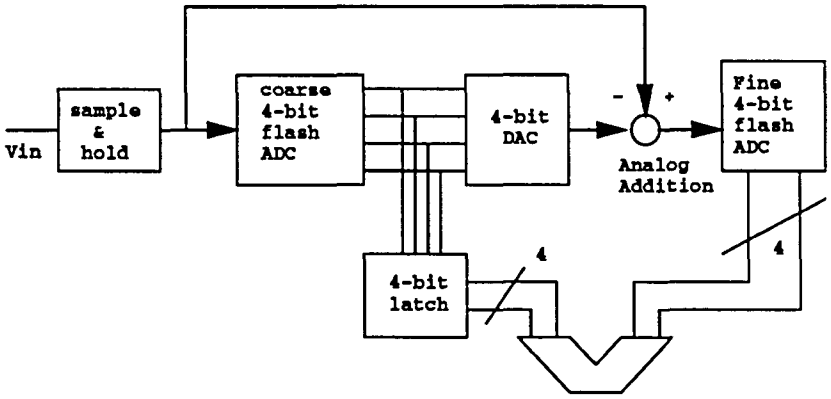

Figure 1.2 Two-step A/D converter structure

The two-step method uses a coarse and fine quantization to increase the resolution of the converter as shown in Figure 1.2. Consider, for example, an 8-bit system that uses a 4-bit coarse quantization. After the coarse quantization has been performed, the 4-bit digital data are converted into an analog value again using a 4-bit D/A converter. This analog value is subtracted from the input signal and the difference is applied to a 4-bit fine converter which generates the fine code.

In this system an ideal coarse-fine signal level matching is expected. In practical applications, however, timing and accuracy limitations can result in conversion problems resulting in "missing" codes [19]. To avoid such a problem, the full-scale range of the fine converter is increased with respect to the LSB step size of the coarse system. In this way a compensation for errors between coarse and fine conversion is obtained.

In this system only 40 comparators are needed to achieve 8-bit resolution. The (4-bit) D/A converter in these applications, however, needs to have an 8-bit accuracy and linearity to obtain the full 8-bit overall linearity. Furthermore, a sample-and-hold amplifier is needed to compensate for the time delay in the coarse quantization/reconstruction step. In this way the dynamic performance of the A/D converter is mostly determined by the performance of the sample-and-hold amplifier.

1.2.3 Successive Approximation Architecture

The successive approximation architecture is the best known and most commonly employed A/D conversion technique. They are used to achieve relatively high resolution (13-15 bits) at conversion rates of 100KHz-1MHz [23] [24].

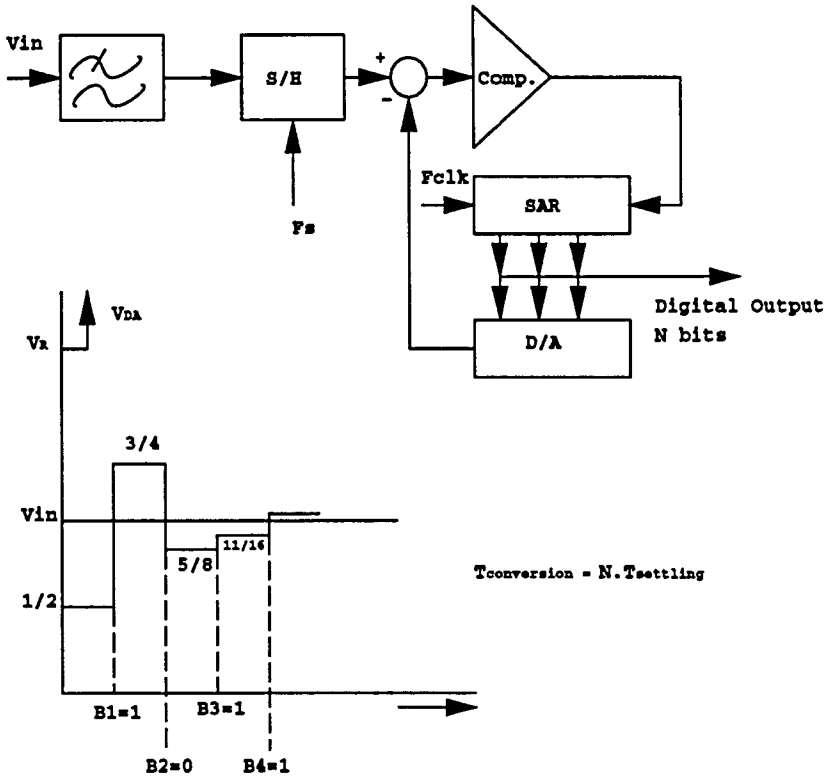

Figure 1.3 Basic successive approximation A/D converter

The basic architecture of a successive approximation analog-to-digital converter is shown in Figure 1.3. The basic converter consists of a comparator stage, the successive approximation register (SAR) and the D/A converter. A sample-and-hold and an anti-aliasing filter are added to limit the maximum analog input frequency and to convert the continuous time input signal into a discrete time signal. At the beginning of the conversion the MSB is switched on and the input signal is compared to the output signal of the D/A converter. When the input signal is larger than the output of the signal of the D/A converter, then the MSB remains on and the next bit is switched on and a comparison will be performed. A bit by bit operation is performed in this system to bring the D/A output signal within 1 LSB to the time discrete input signal. In the lower part of the Figure 1.3 the conversion procedure as a function of bit weighing is shown. The output value in the Figure 1.3 equals 1011. A complete

conversion in this system requires N switching and comparison operations to convert the input signal into a N-bit digital output value. The conversion time equals:

$$T_{conversion} = N \times T_{settling}. \tag{1.2}$$

The settling time is defined as the time required to settle within 1/2 LSB of the D/A converter. The linearity and accuracy of this system depends on the D/A converter [19].

1.2.4 Sigma-Delta A/D Converters

In Figure 1.4 a general form of sigma-delta A/D converter system is shown. The system uses a multi-bit quantizer (analog-to-digital converter) to reconstruct the analog signal. When multi-bit D/A converters are used to reconstruct the analog signal, then the linearity of such a converter is important. In case of high-resolution converters an accuracy problem in the D/A system is encountered. To overcome this accuracy problem a 1-bit system is used. In a 1-bit D/A converter the linearity is determined by the accuracy of the switching between the reference signals. If a high switching accuracy can be guaranteed, then a very linear system is obtained. From the input signal the output signal of the 1-bit D/A converter is subtracted. The difference of these two signals is filtered by the loop filter, and the output signal of the loop filter is applied to the 1-bit quantizer or A/D converter. The clock frequency of the system is high compared to the maximum analog input frequency while the order of the loop filter determines the dynamic range of the system. The output of the 1-bit A/D converter is usually applied to a digital low-pass filter which rejects signals above the signal band of interest. Then sub-sampling or decimation is applied to obtain a multi-bit output code. The whole operation results in a binary-weighted digital output signal that can have a minimum sampling ratio equal to the twice the signal bandwidth [19].

1.2.5 Algorithmic A/D Converter

A block diagram of the converter is shown in the Figure 1.5 [9]. The algorithmic A/D converter consists of an analog signal loop which contains

1) a sample-and-hold amplifier.

2) a multiply-by-two amplifier

3) a comparator, and a

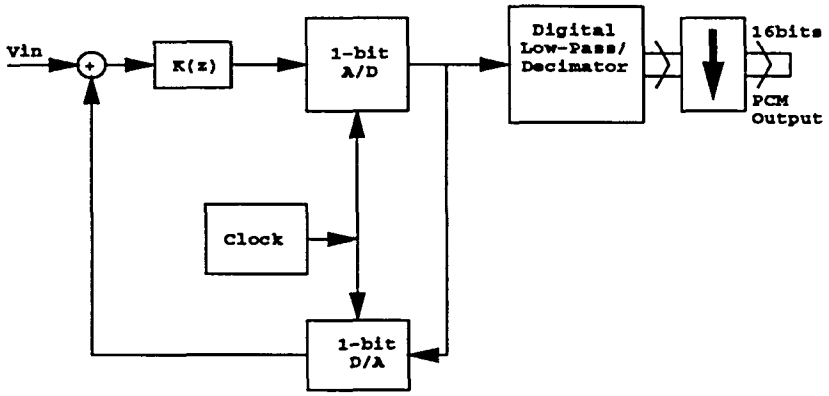

Figure 1.4 Sigma-delta A/D converter system

4) reference subtraction circuit.

The operation of the converter consists of first sampling the input signal onto the sample/hold amplifier. This is done by selecting the input signal instead of the loop signal using the select switch S1. The input signal is then passed to the multiply-by-two amplifier where it is amplified. To extract the digital information from the input signal, the resultant signal, denoted V_a, is compared to the reference. If it is larger than the reference, the corresponding bit is set to 1 and the reference is then subtracted off from V_a. Otherwise, this bit is set to 0 and the signal V_a is kept unchanged. The resultant signal, denoted V_b is then transferred, by means of switch S1, back into the analog loop for further processing. This process continues until the desired number of bits have been obtained, whereupon a new sampled value of the input signal will be processed. Thus, the digital data come out from the converter in a serial manner, most significant bit first.

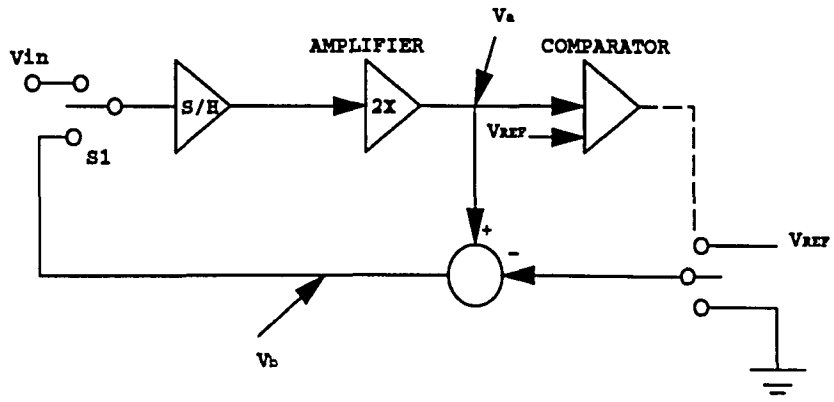

Figure 1.5 Block diagram of the algorithmic A/D converter

The algorithmic A/D converter can be constructed with very little precision hardware. Its implementation in a monolithic technology can therefore be relatively area-sparing. It also possesses inherent sample/hold capability because the sample/hold amplifier is an integral part of the converter.

1.2.6 Pipeline Converters

Integral linearity of data converters usually depends on the matching and linearity of integrated resistors, capacitors, or current sources and is typically limited to approximately 10 bits with no calibration. For higher resolutions means must be sought that can reliably correct nonlinearity errors. This is often accomplished by either improving the effective matching of individual devices or correcting the overall transfer characteristics.

The concept of pipelining often used in digital circuits, can also be applied in the analog domain to achieve higher speed where several operations must be performed serially. Each stage of a pipelined converter consists of a sample-and-hold amplifier, a coarse flash ADC and DACs. The sample-and-hold amplifier samples the input signal and holds it for some time while the conversion is being performed. After the ADC output settles, a voltage determined by the ADC output code is subtracted (or added) from the sampled signal and the residue is calculated. This residue is then sampled by the sample-and-hold amplifier of the next stage. Therefore the first stage operates on the most recent sample while the second stage operates on the residue of the previous stage and so on.

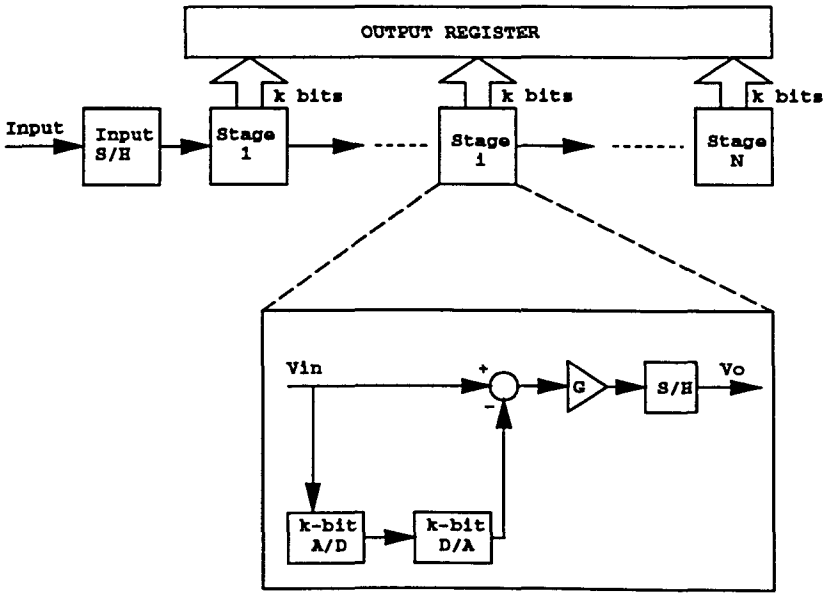

Figure 1.6 Pipeline A/D converter

Figure 1.6 shows a general pipeline system. Here, each stage carries out an operation on a sample, provides the output for the following sampler, and, once that sampler has acquired the data, begins the same operation on the next sample. Thus at any given time, all the stages are processing different samples concurrently, and hence the throughput rate depends only on the speed of each stage and the acquisition time of the next sampler. In sharp contrast to the flash ADC structure, pipeline ADCs only require linear growth in hardware for increasing resolution. Adding another stage to the pipeline will increase the resolution by at least 1-bit, depending on the resolution of the stage.

While the concurrent operation of pipelined converters makes them attractive for high speeds, their linear processing of the analog input relies heavily on operational amplifiers, which are relatively slow building blocks in analog design. The maximum allowable gain error and nonlinearity of the sample-and-hold amplifier in each stage is commensurate with the number of bits resolved afterward and must be maintained well below 1 LSB in the first few stages, thus mandating the use of high gain op amps in high resolution converters. Single-supply op amp designs with such gains and large output swings suffer from slew rate and phase margin degradation.

The number of bits resolved in each stage and hence the number of stages of a pipelined ADC depend on various considerations such as overall resolution, speed, and technology. Practical implementations vary from cascaded flash stages to 1-bit-per-stage topologies, each of which has its own merits and drawbacks. This work will be concerned mostly with 1-bit-per-stage implementation.

1.3 Simulator Terms

Most of the theory was developed and tested using a simulator, called SIMPAD which is explained in detail in Chapter 5. During this testing and developing phase SIMPAD itself was modified and features were added which expanded its capabilities. The results presented in the thesis were solely obtained from SIMPAD. As such, some of the terms used in the simulator need to be defined so that the experimental results can be better understood.

Percentage Component Error: This is the percentage of error introduced in the DAC, flash ADC and the amplifier. This error is present in all the stages of the pipeline and is different for each stage. The error distribution can be either uniform or extreme. In extreme error, if the specified component error is 1%, then the error on each component would be either +1% or -1%. In case of uniform error, a random number is generated (the probability of number being generated is uniformly distributed between the two extremes) which is then factored to the percentage value and is then applied to the components. For example, if an amplifier has an ideal gain of 2 and the component error is 1%, then the actual gain

of the amplifier will lie between 1.9 and 2.1.

Percentage opamp non-linearity: It refers to the equivalent output-referred mid-point nonlinearity introduced by the stage, expressed in fraction of the analog range. The nonlinearity may be either single bow or double bow. A more detailed explanation of the single bow and double bow nonlinearity is done in Chapter 5.

1.4 Conclusion

In this chapter an introduction to various ADC architectures was given and advantages and limitations of different architectures was presented. In Chapter 2, pipeline architectures and accuracy bootstrapping algorithm are discussed in detail. The experimental results with simplified cell architecture , developed in association with Defense Systems Electronics Group of Texas Instruments, are presented and the conclusions from the results are also derived. In Chapter 3, the non-binary architecture is explained and digital self-calibration is revisited. Building on the drawbacks of digital self-calibration algorithm, an improved algorithm for non-binary radix architecture called the Accuracy Bootstrapped Digital Self Calibration (ABDiSC) algorithm is proposed. In Chapter 4, experimental results comparing digital self-calibration and ABDiSC are presented. In Chapter 5, the software developed for the simulation of pipelined converters called SIMPAD is explained and some of its features are highlighted. In Chapter 6, conclusions are made and future direction of further research laid down.

2 PIPELINE ARCHITECTURES AND ACCURACY BOOTSTRAPPING

In this chapter, pipeline architectures, the effect of errors on the transfer curve and a digital calibration algorithm referred to as accuracy bootstrapping which removes these errors is explained in detail. Results with a simplified cell architecture for the pipelined converter are presented.

The three main components of a cell in a pipeline ADC and therefore the error sources are the flash A/D converter, the D/A converter and the sample/hold amplifier as shown in Figure 2.1. These errors are classified as flash ADC errors, DAC errors and gain errors [3]. The errors in the flash ADC are mainly due to comparator offsets and reference level errors. The errors in the DAC are mainly due to the reference level errors and switch feed-through effects. Gain errors mainly occur because of capacitor mismatches and component mismatches. The sample/hold is usually implemented as a switched capacitor gain stage with gain being the ratio of two capacitors as shown in Figure 2.2. Since there is always capacitor mismatch, this contributes to the gain error.capacitors, commonly known as kT/C noise, also introduces errors at resolutions higher than 10 bits.

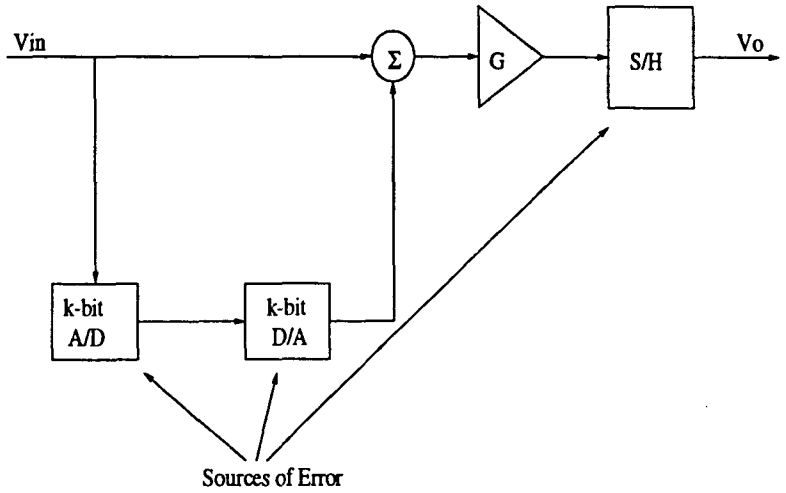

Figure 2.1 Sources of error in a pipelined A/D converter

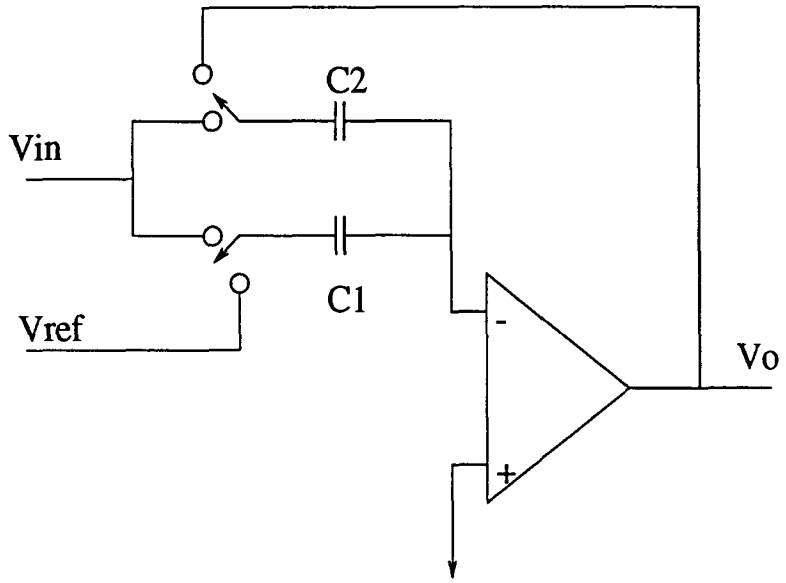

Figure 2.2 Switched capacitor sample-and-hold

2.1 Effect of Errors

Figure 2.3(a) shows the ideal residue plot and the effects of principal errors on the residue plot [1]. The dashed box represents the *reference boundary* that passes through the coordinates $\pm V_{ref}$ along the V_{out} axis and $\pm V_{ref}$ along the V_{in} axis. Charge injection offset, Figure 2.3(b), causes a vertical shift of the residue plot. Near the major carry transition point, the residue exceeds the reference boundary resulting in missing decision levels. This is because the remaining pipeline section is saturated so that the output code does not change for the corresponding range of analog input. Near the major carry transition point, the residue minimum does not extend to $-V_{ref}$ resulting in a gap from the minimum to the reference boundary. As a result, missing codes result. This is because the full input range of the remaining pipeline section is not accessed. Comparator offset, Figure 2.3(c), causes a shift of the major carry transition point. This leads to the residue exceeding the reference boundary as well as leading to a gap to the reference boundary. Again, missing decision levels and missing codes result, respectively. Finally, capacitor mismatch, Figure 2.3(d), causes the residue to exceed the reference boundary near the major carry transition point resulting in missing decision levels. Capacitor mismatch could also lead to a gap from the residue extrema to the reference boundary near the major carry transition point in missing codes.

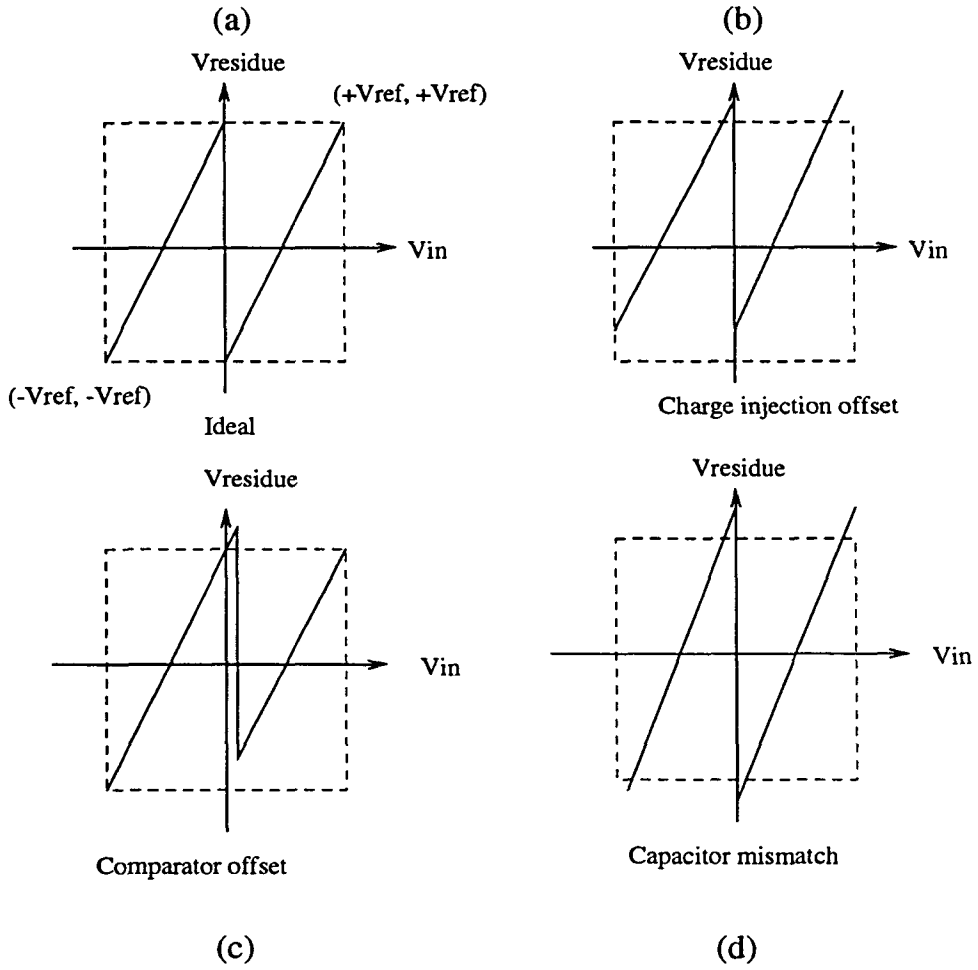

Figure 2.3 Effect of errors on the transfer curve. Missing decision levels and codes are present

2.2 ADC Calibration Techniques

High resolution ADC's when limited by component mismatches or circuit nonidealities, have relied on corrective measures such as trimming or electronic calibration. In general, most high-resolution converters recently developed can be classified into two groups. One group uses various circuit techniques which correct nonlinearities in the analog domain while the other group uses calibration techniques which quantize nonlinearity errors in digital forms and removes these premeasured errors during normal conversion. In analog calibration, individual errors of the capacitors are measured and stored in memory during the calibration cycle, and then read from the memory to estimate residue errors during normal conversion. However, in digital calibration, after the digital output is obtained (which is uncalibrated), the code errors are subtracted from the digital output to obtain the calibrated digital output.

2.3 Accuracy Bootstrapping Algorithm

The accuracy bootstrapping algorithm [2], [3] is a powerful technique for calibration of ADCs and best suited for multistage, pipelined converters with nominally identical stages. It corrects the most common causes of converter nonlinearity: errors on comparator trip points, incorrect DAC levels and incorrect amplifier gains. In the following sections, a mathematical description of the algorithm will be given.

2.3.1 Mathematical Description

Figure 2.4 shows a schematic of one converter stage. The input signal, V_{in} is compared against a number of reference levels,V^{ref}, using a flash converter. The output of the flash ADC is a digital code. The number of bits in the digital code depends upon the number of reference levels. If there are M comparators, then there will be M+1 possible codes in thermometer format. The flash section is followed by a reconstructing digital/analog converter (DAC). Depending on the digital code, one out of M + 1 possible voltages, V^{DAC}, is subtracted from the input signal. The difference signal $(V^{in} - V^{DAC})$ is amplified by a sample/hold amplifier. The resulting signal is called the residue, $V^{res} = (V^{in} - V^{DAC}) * A$. This residue becomes the input voltage for the next stage.

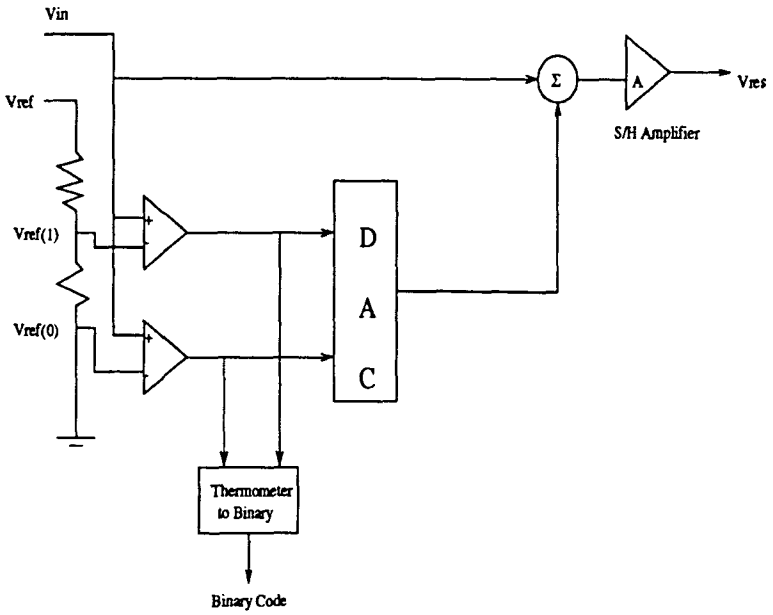

Figure 2.4 Schematic of single converter stage

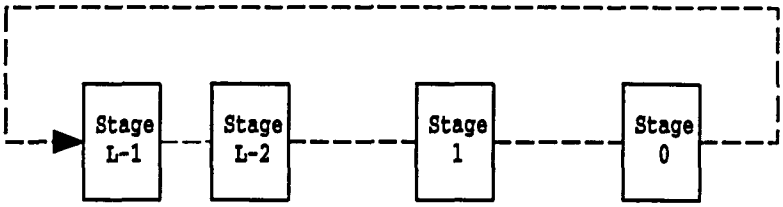

Figure 2.5 Pipeline stages

For a pipeline with L stages shown in Figure 2.5, let l denote the lth converter. Depending on the input voltage, each of the stages will have one of the following digital codes:

$$\begin{aligned} c_l \;=\; & 0 \;\; for \;\; V_l^{in} < V_l^{ref}[0] \\ & 1 \;\; for \;\; V_l^{ref}[0] \le V_l^{in} \le V_l^{ref}[1] \\ & \ldots\ldots\ldots \\ & M \;\; for \;\; V_l^{ref}[M-1] \le V_l^{in} \end{aligned} \tag{2.1}$$

where c_l is the digital code of stage l.

For any given c_l, a general equation can be written for the residue V_l^{res} as a function of V_l^{in}.

$$V_l^{res} = (V_l^{in} - V_l^{DAC}[c_l])A_l \tag{2.2}$$

Rearranging,

$$V_l^{in} = V_l^{DAC}[c_l] + \frac{V_l^{res}}{A_l} \tag{2.3}$$

Since the residue of one stage forms the input for the next stage, it is clear that

$$V_{l-1}^{in} = V_l^{res} \tag{2.4}$$

Therefore the expression for the input voltage of stage l can be expanded to

$$V_l^{in} = V_l^{DAC}[c_l] + \frac{V_{l-1}^{DAC}[c_{l-1}]}{A_l} + \frac{V_{l-1}^{res}}{A_l A_{l-1}} \tag{2.5}$$

For the L stage pipeline the input voltage can be represented as

$$V^{in} = V_{L-1}^{DAC}\frac{A^L}{A^L} + V_{L-2}^{DAC}\frac{A}{A_{L-1}}\frac{A^{L-1}}{A^L} + ... + V_0^{DAC}\frac{A^{L-1}}{A_{L-1}...A_1}\frac{A}{A^L} + \frac{V_0^{res}}{A_{L-1}...A_0} \tag{2.6}$$

Equation (2.6) can be rewritten as

$$V^{in} = V_{L-1}^{DAC}\frac{A^{(L)}}{A^{(L)}} + V_{L-2}^{DAC}\frac{A}{A_{L-1}}\frac{A^{(L-1)}}{A^{(L)}} + V_{L-3}^{DAC}\frac{A^2}{A_{L-1}A_{L-2}}\frac{A^{(L-2)}}{A^{(L)}}. + ... + V_0^{DAC}\frac{A^{(L-1)}}{A_{L-1}...A_1}\frac{A}{A^{(L)}} + \frac{V_0^{res}}{A_{L-1}...A_0} \tag{2.7}$$

$$V^{in} = [[[[[\frac{V_{L-1}^{DAC}}{A^{(L)}}]A + \frac{V_{L-2}^{DAC}}{A^{(L)}}\frac{A}{A_{L-1}}]A + \frac{V_{L-3}^{DAC}}{A^{(L)}}\frac{A^{(2)}}{A_{L-1}A_{L-2}}]A + ...]A + \frac{V_0^{DAC}}{A^{(L)}}\frac{A_{(L-1)}}{A_{L-1}....A_0}]A + \frac{V_0^{res}}{A_{L-1}....A_0}. \tag{2.8}$$

Equation (2.8) expresses the conversion result in a recursive way. When translated into hardware, it results in identical logic for each stage. The digital output of the ADC can be written as

$$Digital\ \ Output = [[[[[\frac{W_{L-1}}{A^L}]A + \frac{W_{L-2}}{A^L}]A + \frac{W_{L-3}}{A^L}]A + ...]A + \frac{W_0}{A^L}]A \tag{2.9}$$

where W_l, called *weight* is the value of DAC for that stage.

Comparing the above two equations

$$W_l^i[c] = \frac{V_l^{DAC}[c]A^{(L-1-l)}}{A_{L-1}....A_{l+1}} \tag{2.10}$$

It is clear that the weight associated with each stage are nominally identical, and nominally equal to the V^{DAC} values, since the correction factors $\frac{A}{A_{L-1}}, \frac{A^{(2)}}{A_{L-1}A_{L-2}}$ etc are nominally equal to unity. In practice, component variations are small and the values of weights will remain close to nominal. A more detailed explanation of the accuracy bootstrapping algorithm can be found in [3], [2].

2.3.2 Modified Architecture of the Single Cell

From (2.9), a modified architecture of the single cell was developed [2], which is shown in Figure 2.6. Depending on the value of *cal*, the input V_{in} or a fixed voltage V_{fix} can be input to the pipeline stage (*cal* is used for running the pipeline in calibration mode). Depending on the code output by the flash ADC one of the weights (W[0], W[1] or W[2]) stored in the RAM will be selected. The weights are

used as weighted coefficients whose sum is used to determine the digital output code. As can be seen in the figure, the output of the RAM is divided by 2^L and added to the output of the previous stage. This is then multiplied by 2 to give the output of the present stage. The digital hardware implements (2.9). Note that if A = 2, the multiplication and division can be done by simple shifting operations. Since the input range for all the stages is identical, the digital hardware can be duplicated from stage to stage.

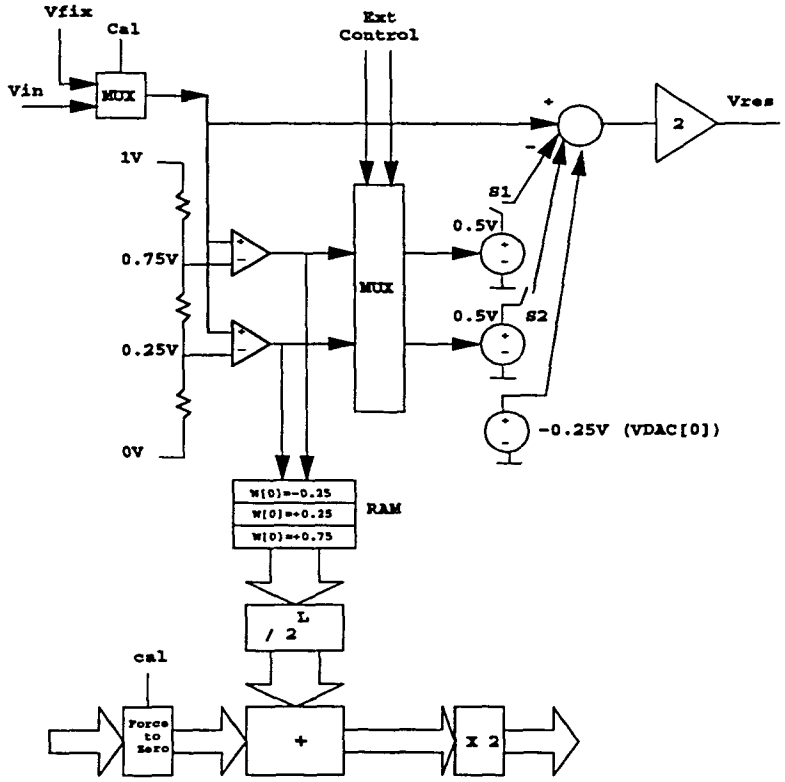

Figure 2.6 Single stage, modified for calibration

Figure 2.6 also shows three voltage sources of values -0.25V, +0.5V and +0.5V. The two voltage sources of value +0.5V are termed as voltage increments and are represented as $\Delta V_l^{DAC}[1]$ and $\Delta V_l^{DAC}[2]$ and the voltage source of -0.25V is represented as $V_l^{DAC}[0]$. Note that

$$V_l^{DAC}[1] = V_l^{DAC}[0] + \Delta V_l^{DAC}[1] \tag{2.11}$$

$$V_l^{DAC}[2] = V_l^{DAC}[0] + \Delta V_l^{DAC}[1] + \Delta V_l^{DAC}[2] \tag{2.12}$$

The voltage increments are switched on by the switches S1 and S2. From the Figure 2.6 it is clear that 3 possible results/stage are possible. Thus bits per stage are 2 but only 3 out of four possible codes are

used. Also, since it has 0.25V over range capability, it is in effect a 1.5 bits/stage architecture with 0.5 bits overlap between stages. The calibration procedure is described in next section.

2.3.3 Procedure for Calibration

The basic idea of accuracy bootstrapping is to individually measure all the DAC levels of each converter stage i.e. points N1 and N2 shown in Figure 2.7, using the remaining stages of the pipeline. The measurements are used to update the values in the look-up tables (RAM in Figure 2.6) of that stage, and the process is repeated until each stage has been calibrated. The advantage of the algorithm is that it uses the hardware already present to do the calibration. Experimentally, it has been observed that the calibration results in an iterative numerical problem that converges to the result very fast. The procedure is as follows:

(1) The values in the look-up table, called weights, are initialized to their nominal values. Since the weights are actually the value of the DAC's, they are initialized to -0.25V, 0.25V and 0.75V i.e. it is assumed that there are no errors in the DAC's.

(2) The last stage is calibrated first. An analog input, V_{fix} is applied to the last stage, with the help of *cal*, while none of the voltage increments are enabled. Note that the voltage source of -0.25V ($V^{DAC}[0]$) is always enabled. The output of this stage is connected to the input of the first stage (as shown in Figure 2.5) and this circular structure is used to determine the conversion result or digital code, called N_1.

(3) Continuing with the calibration of the last stage, the first voltage increment ($\Delta V_0^{DAC}[1]$) is enabled by switch S1. The same circular structure is used to determine this conversion result, N_2. Next, the second voltage increment ($\Delta V_0^{DAC}[2]$) is enabled, while the first is disabled, to obtain N_3.

(4) The new weights for the last stage are then calculated. $W_0[0]$ is fixed to its nominal value. The rest of the weights are calculated as follows.

$$W_0[1] = W_0[0] + \left(\frac{N_1 - N_2}{A}\right) \tag{2.13}$$

$$W_0[2] = W_0[1] + \left(\frac{N_1 - N_3}{A}\right) \tag{2.14}$$

All the original (nominal) weights of the last stage are now replaced by the newly determined ones. This concludes the calibration of the last stage. The new weights reflect the values of the DAC's more accurately.

(5) The procedure for calibrating stage 0 is now repeated to calibrate stage 1. The values N_1, N_2 and N_3

are measured, using the converter formed by stages 0, L-1, L-2,...2 and the comparator of stage 1. The new values of $W_1[c]$ are calculated and updated. The same procedure is applied to stages $l = 2.....L-1$ i.e. the stage l is calibrated using the converter formed by stages $l-1, l-2, ...L-1, ..l$. The previously calibrated stage is always the first (most significant) stage in the converter used to calibrate the next one.

The calibration process stops once all the stages have been calibrated.

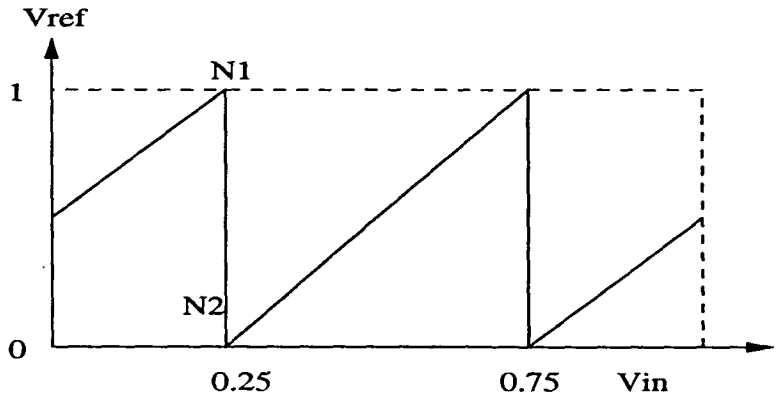

Figure 2.7 DAC levels to be measured

Correct calibration can be obtained even when the analog components of the pipeline are noisy. The influence of the noise upon the calculation of the weights can be reduced to an arbitrarily low level by averaging a number of successive measurements for each voltage increment or DAC value.

2.3.4 Explanation for the Calibration Technique

The fixed voltage V_{fix} is of value 0.25V. In the calibration procedure, first none of the voltage increments are enabled and the value N_1 is obtained, then the first voltage increment is enabled and value N_2 is obtained and finally the second voltage increment is enabled (and the first one is disabled) and N_3 is obtained. Mathematically,

$$N_1 = (0.25 - V_l^{DAC}[0])A_l \tag{2.15}$$

$$N_2 = (0.25 - V_l^{DAC}[0] - \Delta V_l^{DAC}[1])A_l \tag{2.16}$$

$$N_3 = (0.25 - V_l^{DAC}[0] - \Delta V_l^{DAC}[2])A_l \tag{2.17}$$

In the first analysis, let us consider the ADC to full accuracy. Then the weights would be calculated as follows. For the stage 0 the weights would be

$$W_0[1] = W_0[0] + \left(\frac{N_1 - N_2}{A}\right) \tag{2.18}$$

and

$$W_0[2] = W_0[1] + \left(\frac{N_1 - N_3}{A}\right) \tag{2.19}$$

Therefore if there was no error in calculating the final result in the ADC,

$$\frac{N_1 - N_2}{A} = \frac{\Delta V_0^{DAC}[1] A_0}{A} \tag{2.20}$$

$$\frac{N_1 - N_3}{A} = \frac{\Delta V_0^{DAC}[2] A_0}{A} \tag{2.21}$$

Rewriting (2.18) and (2.19) we have,

$$W_0[1] = W_0[0] + \frac{\Delta V_0^{DAC}[1] A_0}{A} \tag{2.22}$$

$$W_0[2] = W_0[0] + \frac{\Delta V_0^{DAC}[1] A_0}{A} + \frac{\Delta V_0^{DAC}[2] A_0}{A} \tag{2.23}$$

In the ideal case i.e. when there are no component errors then A_l is equal to nominal A and the terms $N_1 - N_2$ and $N_1 - N_3$ give the exact value of the voltage increments $\Delta V_0^{DAC}[1]$ and $\Delta V_0^{DAC}[2]$, respectively. As a result (2.22) and (2.23) are exactly equal to (2.11) and (2.12).

A more detailed analysis of the properties of accuracy bootstrapping algorithm in non-ideal cases can be found in [25].

2.4 Simplified Cell Architecture

2.4.1 Operation of the Single Stage

The architecture of the single stage of the pipelined analog-to-digital converter is shown in Figure 2.8. It consists of two comparators of value $\pm 0.25V$ ($\pm V_{ref}/2$), two latches, some decoding logic, two voltage sources of value $\pm 0.5V$ ($\pm V_{ref}$), a summer and an amplifier with a gain of 2. The nominal input range of a single stage is between -0.5V to 0.5 V ($-Vref$ to $+Vref$). The transfer curve for the single stage is shown in Figure 2.9. When the input is between -0.5V to -0.25V (region "00"), E1 is enabled

and a voltage equivalent to E1 is added to the input. When the input voltage is between -0.25V to +0.25V (region "10"), none of the voltage sources is enabled and the input voltage is passed through i.e. subtract zero from the input. When the input is greater than 0.25V (region "11"), then the voltage source E2 is enabled and a voltage equivalent to E2 is subtracted from the input.

Since accuracy bootstrapping has proved to be a very useful technique for calibration we have used it for calibration here also. For calibration, the points $N0, N1, N2$ and $N3$ need to be determined as shown in the Figure 2.9. When compared to the architecture described in [2], it has reduced hardware - two reference voltages compared to three, and two look-up table registers compared to three, at each stage. Moreover it has a range of $\pm 0.5V$. On the other hand, for calibration of each stage, four measurements are needed compared to three. But that does not effect the performance of the converter in real time as the calibration measurements need not be done on the fly.

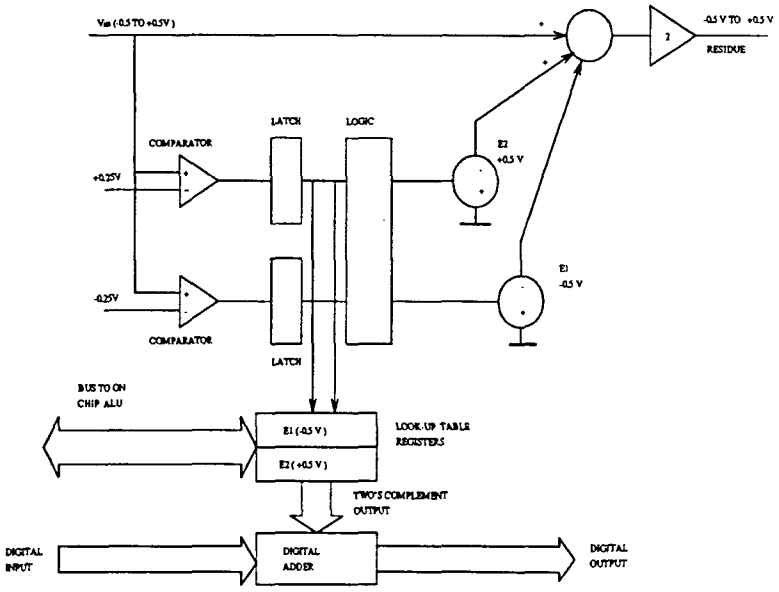

Figure 2.8 Simplified cell architecture

2.4.2 Passing Zero

When the input voltage is between -0.25V to +0.25V (region "10"), none of the voltage sources is enabled and the input voltage is passed through i.e. zero is subtracted from the input. The issue that was left to be decided was how to pass a zero i.e. how to subtract zero in the analog domain and its effect in the digital domain, during calibration. Ideally the equation for V_{res} during calibration would

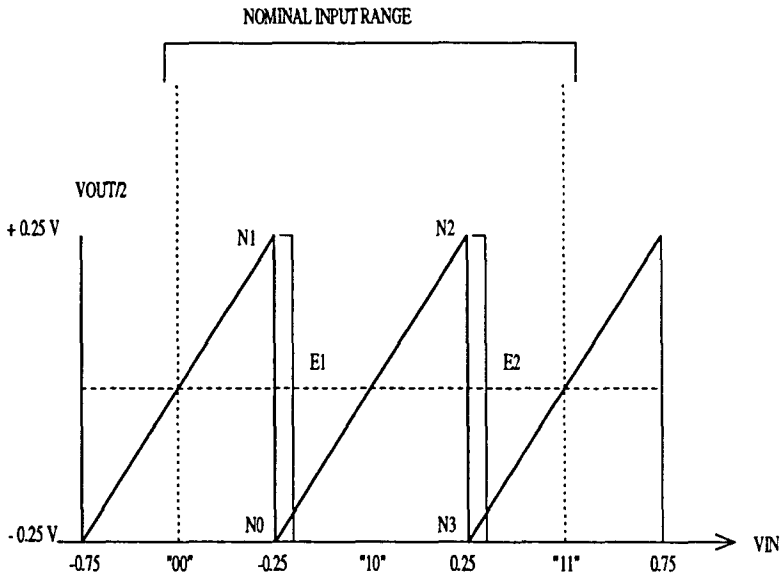

Figure 2.9 Transfer characteristic

be,

$$V_{res} = V_{fix} - 0 \tag{2.24}$$

The following options were considered to subtract zero in the analog domain and represent it in the digital domain:

Option 1:

During Calibration $\Rightarrow V_{fix} - Ground$

In Analog Domain $\Rightarrow V_{in} - Ground$

In Digital Domain $\Rightarrow 0$

Option 2:

During Calibration $\Rightarrow V_{fix} - Ground$

In Analog Domain $\Rightarrow V_{in} - Ground$

In Digital Domain $\Rightarrow W0 + W1$

Option 3:

During Calibration $\Rightarrow V_{fix} - Ground$

In Analog Domain $\Rightarrow V_{in} - DAC0 - DAC1$

In Digital Domain $\Rightarrow W0 + W1$

Option 4:

During Calibration $\Rightarrow V_{fix} - DAC0 - DAC1$

In Analog Domain $\Rightarrow V_{in} - Ground$

In Digital Domain $\Rightarrow W0 + W1$

Option 5:

During Calibration $\Rightarrow V_{fix} - DAC0 - DAC1$

In Analog Domain $\Rightarrow V_{in} - DAC0 - DAC1$

In Digital Domain $\Rightarrow W0 + W1$

The reason the calibration operation is considered separately even though it comes in the analog domain was the fact that its operation can be independent of the input signal. In the above options $V - Ground$ was used to simulate the fact that subtracting zero from the input voltage is equivalent to grounding the capacitors used for sampling the reference voltages. Since there is going to be some residual charge left on the capacitor, this subtraction is not going to be perfect. Therefore, some random value equal to the charge left on the capacitor has to be subtracted during the simulation. This value of the ground called "random_zero" in our simulations was assumed to be about the same magnitude as the error on the DACs (of course, the value of random error for the DAC and the value of random_zero were different). Moreover the value of random_zero was assumed to be different for each stage.

Instead of grounding one of the capacitors we can also enable both the voltage sources as their ideal sum is zero. So $V - DAC[0] - DAC[1]$ is used to simulate the fact that both the voltage sources are enabled. This can be done either only during calibration or only during the normal operation or at all times.

When the output of the flash A/D is "00" , W[0] is added to the digital input of Figure 2.9. If the output is "11", W[1] is added to the digital input. When the output of the flash A/D is "10" then 0 is added to the digital input. This is shown as a simple zero in the digital domain. In the above options a zero in the digital domain indicates that nothing is being added to the digital input of Figure 2.8. On the other hand, W[0] and W[1] when added should yield zero under ideal conditions, then this sum can

be added to the digital input. This is represented by W[0]+W[1] in the digital domain.

The results for each of the options and the explanation of the results is given in the next section.

2.5 Results with Different Options

A random error of 1% of full scale value was applied to each of the components of a 12-bit pipelined converter and then a linear sweep was performed. Then the converter was calibrated using accuracy bootstrapping and the linear sweep was performed again. For each option, the results were obtained for two different converters. The results are shown in Table 2.1. The first column , "Option", stands for the options described in the previous section; column "Linear Sweep" stands for the number of linear sweep. For each linear sweep, maximum INL and DNL was calculated before calibration and are shown in columns 3 and 4. After calibration, another linear sweep was performed and maximum INL and DNL was obtained which is shown in columns 5 and 6.

We can clearly see from the Table 2.1 that if we enable the two DACs and use them as zero instead of grounding one of the capacitors, then we get better results. An explanation of these results is given in the next section.

Table 2.1 Results for uncalibrated and calibrated pipeline with different options

Option	Linear Sweep	Max.INL(LSB) Before Calibration	Max.DNL(LSB) Before Calibration	Max.INL(LSB) After Calibration	Max.DNL(LSB) After Calibration
Option 1	1	50.3914	35.5733	93.7588	71.6894
Option 1	2	-26.2272	-24.8607	-45.4635	37.5297
Option 2	1	25.0508	30.3456	-31.8691	26.4825
Option 2	2	-24.9460	-31.8399	-16.0252	-12.9485
Option 3	1	29.8472	27.2701	17.6909	-9.7212
Option 3	2	30.1227	-39.1311	-18.4298	-15.7671
Option 4	1	24.1238	32.2601	46.7047	27.2160
Option 4	2	-31.0528	-46.9728	52.9695	52.8286
Option 5	1	42.5345	58.1017	-0.9121	1.0230
Option 5	2	21.9081	26.0422	-1.0426	0.9904

2.6 Explanation of the Results

In this section, the error due to grounding the capacitor will be referred to as $\delta 0$, which stands for a random number close to zero, but not zero. So $V - Ground$ will be written as $V - \delta 0$.

If $\delta 0$ is used for calibration, then

$$N0 = V_{fix1} - \delta 0 \tag{2.25}$$

$$N1 = V_{fix1} - D0 - \delta D0 \tag{2.26}$$

$$N2 = V_{fix2} - \delta 0 \tag{2.27}$$

$$N3 = V_{fix2} - D1 - \delta D1 \tag{2.28}$$

where

$V_{fix1} = -0.25V$

$V_{fix2} = +0.25V$

$D0 = DAC[0]$

$\delta D0 =$ error on DAC[0]

$D1 = DAC[1]$

$\delta D1 =$ error on DAC[1]

$N0, N1, N2, N3$ are the points as shown in Figure 2.9.

Therefore,

$$W0 = N0 - N1 \tag{2.29}$$

$$W1 = N2 - N3 \tag{2.30}$$

Rewriting $W0$ and $W1$ in it's components, we have

$$W0 = D0 + \delta D0 - \delta 0 \tag{2.31}$$

$$W1 = D1 + \delta D1 - \delta 0 \tag{2.32}$$

The above two equations will also have a gain error term which has been removed for simplicity without

any loss of generality.

It is clear from the above equations that every time the DACs are calibrated by grounding one of the capacitors, an error term of some finite value, shown as $\delta 0$ is added to the calibrated weight of the DACs. Therefore the calibrated value has an error.

On the other hand, if both the DACs are enabled for calibration, then

$$N0 = V_{fix1} - D0 - \delta D0 - D1 - \delta D1 \tag{2.33}$$

$$N1 = V_{fix1} - D0 - \delta D0 \tag{2.34}$$

$$N2 = V_{fix2} - D0 - \delta D0 - D1 - \delta D1 \tag{2.35}$$

$$N3 = V_{fix2} - D1 - \delta D1 \tag{2.36}$$

As a result, the weights now are obtained as shown below:

$$W0 = N3 - N2 = D0 - \delta D0 \tag{2.37}$$

$$W1 = N1 - N0 = D1 - \delta D1 \tag{2.38}$$

One can see now that during calibration, if zero has to be subtracted, then enabling both the DACs gives the right result as the calibrated weights do not contain any error terms present in 2.30 and 2.31.

A summary of all the options and cause of errors is shown below:

Option 1:

During Calibration $\Rightarrow V_{fix} - \delta 0$

In Analog Domain $\Rightarrow V_{in} - \delta 0$

In Digital Domain $\Rightarrow 0$

Cause of Errors: 1) Calibration procedure adds a $\delta 0$ to the weights. 2) Though we are subtracting $\delta 0$ in the analog domain, we are adding an ideal zero to the digital code.

Option 2:

During Calibration $\Rightarrow V_{fix} - \delta 0$

In Analog Domain $\Rightarrow V_{in} - \delta 0$

In Digital Domain $\Rightarrow W0 + W1$

Cause of Errors: 1) Calibration procedure adds a $\delta 0$ to the weights. 2) A value equal to $\delta 0$ is being subtracted in analog domain, and the value that is added in digital domain is $W0 + W1$ which is random with respect to $\delta 0$.

Option 3:

During Calibration $\Rightarrow V_{fix} - \delta 0$

During Normal Operation $\Rightarrow V_{in} - DAC0 - DAC1$

In Digital Domain $\Rightarrow W0 + W1$

Cause of Errors: 1) Calibration procedure adds a $\delta 0$ to the weights.

Option 4:

During Calibration $\Rightarrow V_{fix} - DAC0 - DAC1$

In Analog Domain $\Rightarrow V_{in} - \delta 0$

In Digital Domain $\Rightarrow W0 + W1$

Cause of Errors: 1) A value equal to $\delta 0$ is being subtracted in analog domain, and the value that is added in digital domain is $W0 + W1$ which is random with respect to $\delta 0$.

Option 5:

During Calibration $\Rightarrow V_{fix} - DAC0 - DAC1$

In Analog Operation $\Rightarrow V_{in} - DAC0 - DAC1$

In Digital Domain $\Rightarrow W0 + W1$

This works because the calibration is right and the value being subtracted in the analog domain is the value being added in the digital domain.

2.7 Calibration Results for the Single Pipeline

With the insight gained in previous two sections the pipeline was calibrated by accuracy bootstrapping, enabling both the voltage sources whenever zero needed to be subtracted from the input voltage. In the digital domain, both the weights, W[0] and W[1] will have to be added to the digital input of Figure 2.8. Figure 2.10 shows the transfer characteristic of an uncalibrated 16-bit ADC having 1% component error. Figure 2.11 and Figure 2.12 show the DNL and INL plot of the uncalibrated ADC, respectively. The transfer characteristic after calibration is shown in Figure 2.13, the DNL plot is shown in Figure 2.14 and the INL plot is shown in Figure 2.15.

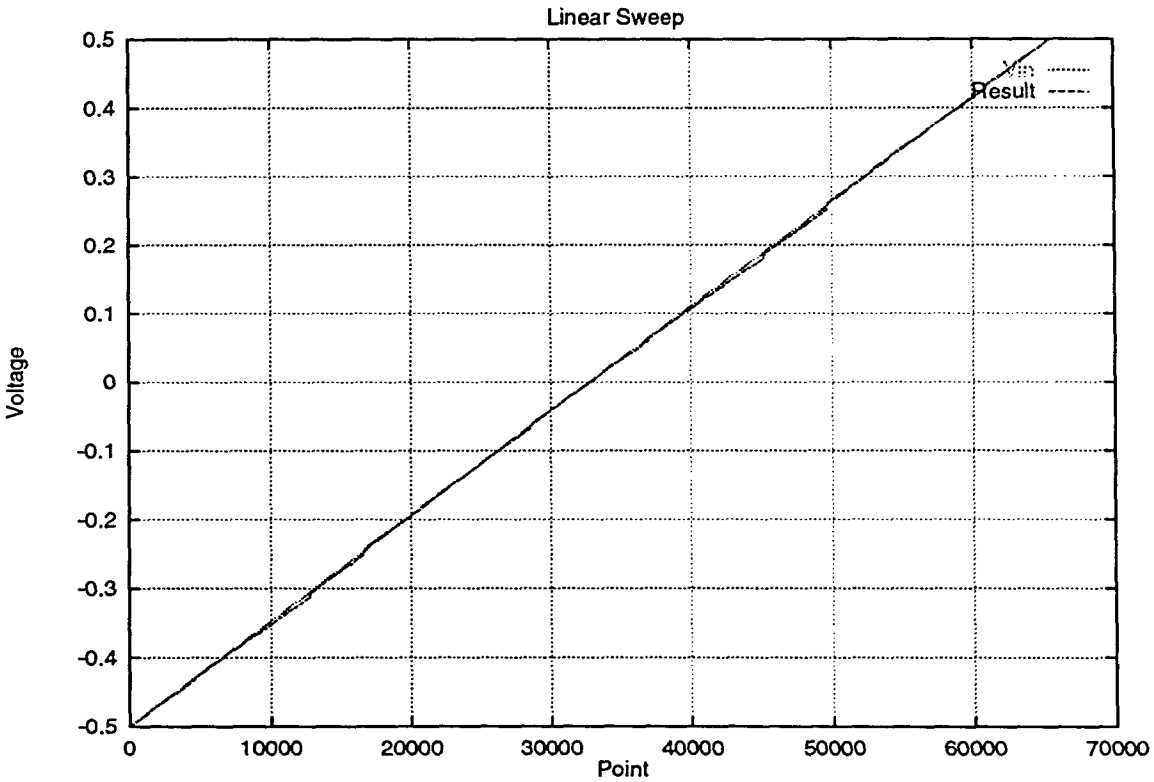

Figure 2.10 Output of the uncalibrated 16-bit ADC

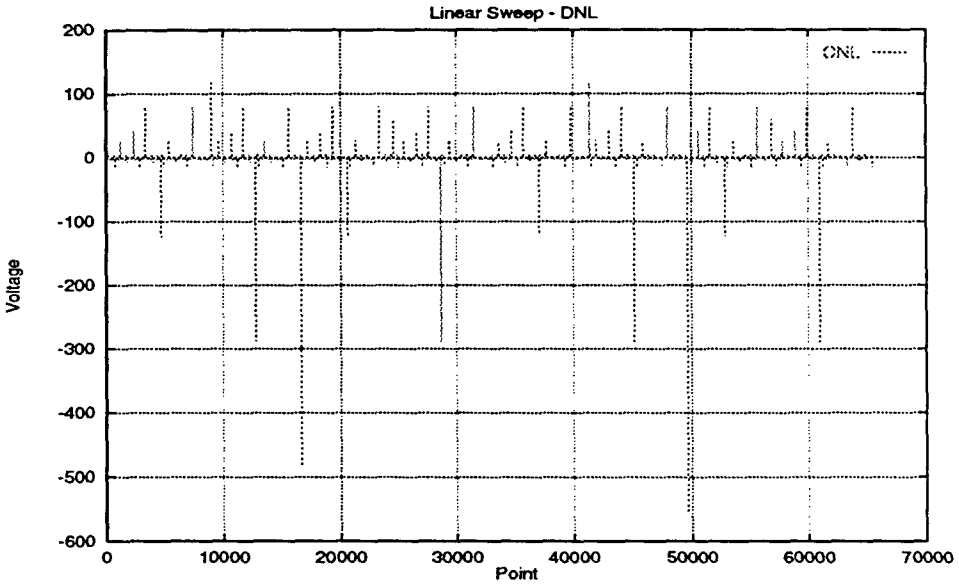

Figure 2.11 DNL of the uncalibrated 16-bit ADC

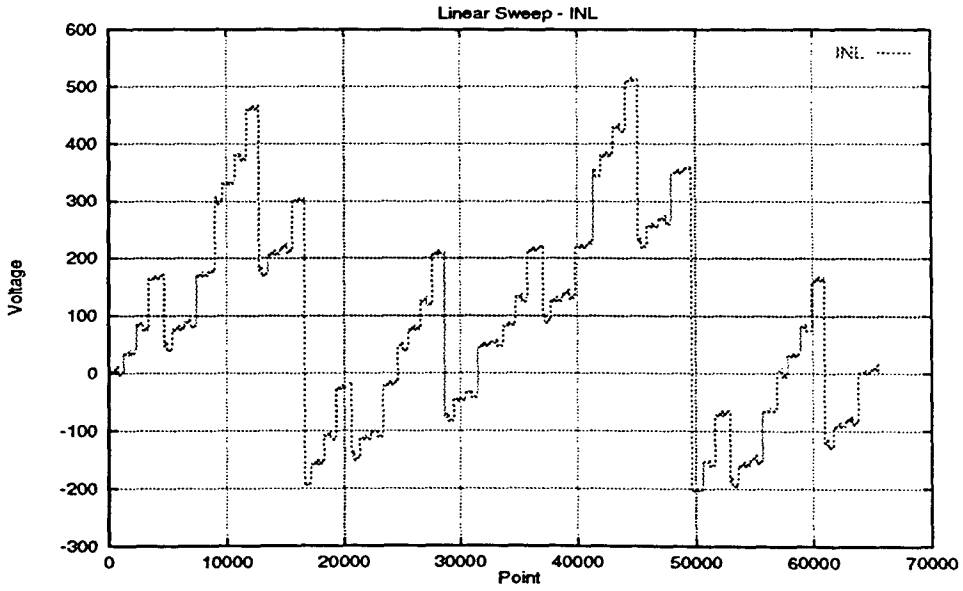

Figure 2.12 INL of the uncalibrated 16-bit ADC

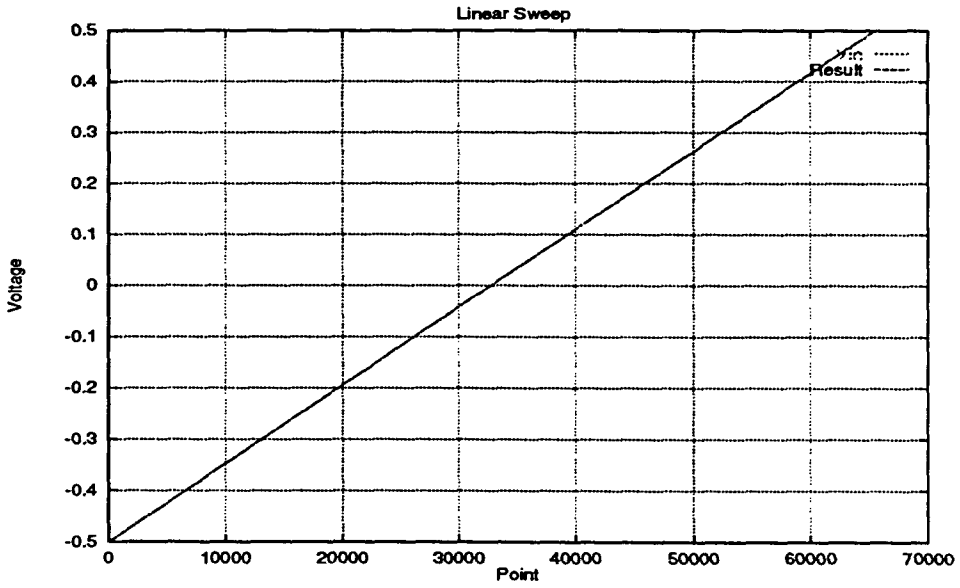

Figure 2.13 Output of the 16-bit ADC after calibration

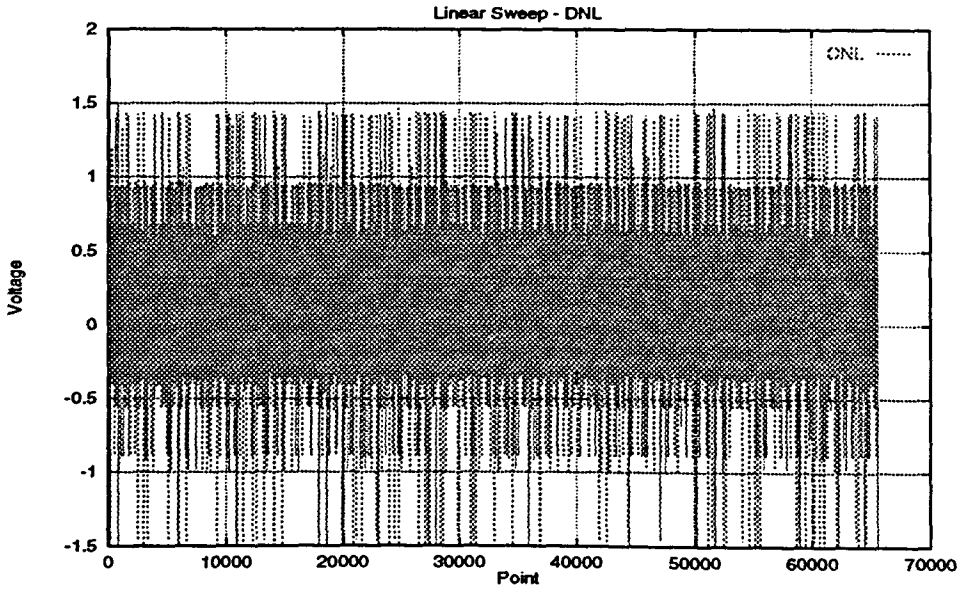

Figure 2.14 DNL of the 16-bit ADC after calibration

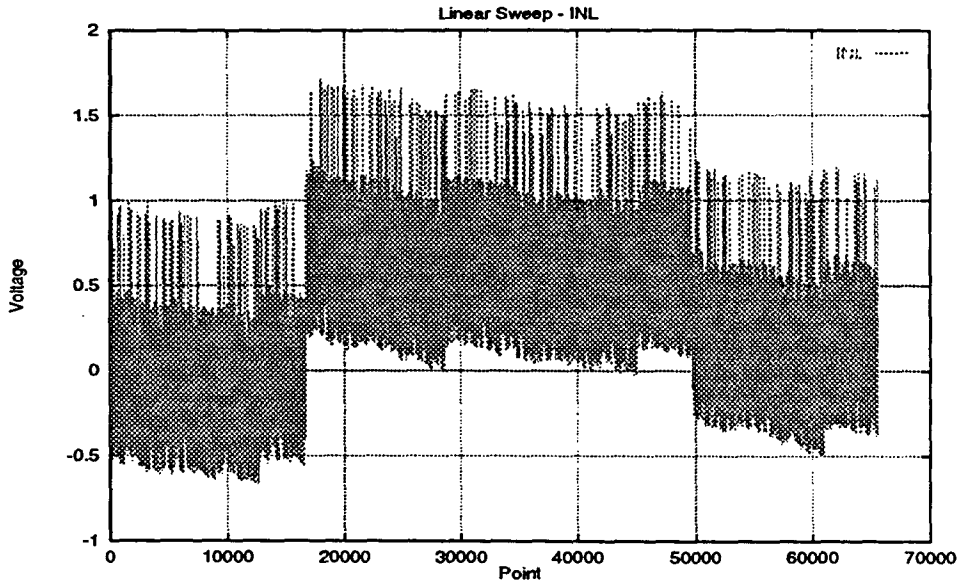

Figure 2.15 INL of the 16-bit ADC after calibration

2.8 Calibration Results with the Parallel Pipeline

The accuracy bootstrapping algorithm with modifications as developed in previous sections, and Global Matching Algorithm [25] were then applied to a four pipe parallel 16-bit A/D converter shown in Figure 2.16. In the Global Matching Algorithm (GMA), the bulk of the linearization is done by accuracy bootstrapping and the pipes (each of which has been already calibrated) are used to linearize the resulting curve. If the transfer characteristic of each of the ADC was linear with around 1 LSB of INL and DNL error, then the slope and intercept of each transfer characteristic can be calculated and then normalized with respect to a fixed characteristic. Since the INL and DNL of each of the pipes is linear to an INL and DNL error of 1 LSB at the 16-bit level, the resulting normalized curve would also be linear to INL and DNL errors of around 1 LSB (assuming that the slope and intercept are calculated correctly). The slope and intercept are calculated by two point measurement. A more detailed explanation of the GMA with the results can be found in [25].

2.8.1 Result of the Linear Sweep

The transfer characteristic of an uncalibrated four pipe parallel 16-bit ADC with 1% component errors is shown in Figure 2.17. The maximum DNL with the uncalibrated ADC was 555.6079 LSB and maximum INL was 550.0792 LSB. After calibration, a linear transfer characteristic was obtained and

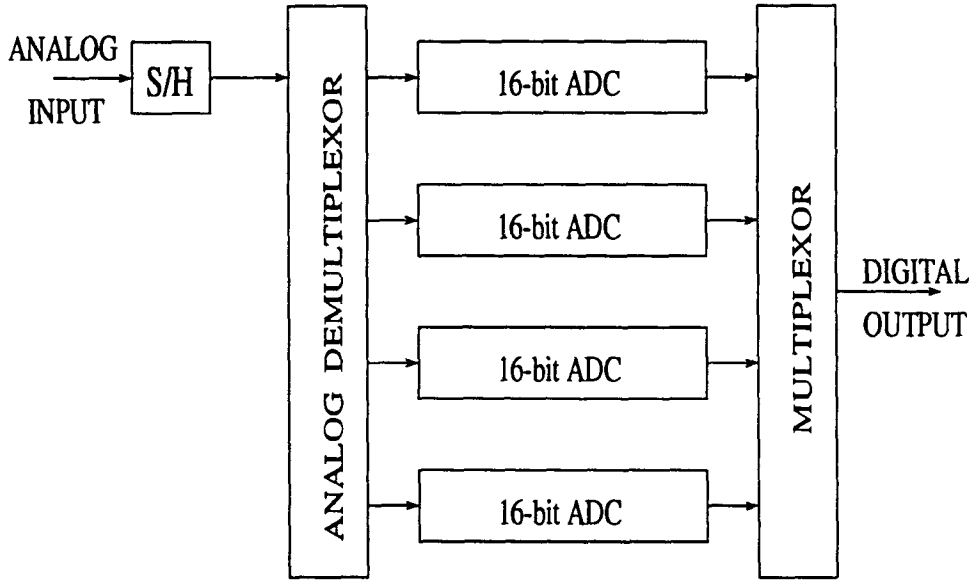

Figure 2.16 A four pipe parallel 16-bit ADC

is shown in Figure 2.18. The corresponding DNL and INL measurements are shown in Figure 2.19 and Figure 2.20, respectively. The maximum DNL after calibration is 0.9 LSB and maximum INL is 0.9 LSB.

2.8.2 Result of the Sinusoidal Sweep - FFT

Two types of tests were performed on the parallel pipes. The first one is called single tone testing and the second one, two tone testing. In single tone testing the input signal contains a single sinusoid, and in our case it was of frequency 1.50390625 MHz. Single tone testing is used to test the spectral purity of the ADC output. In two tone testing the input signal contains two sinusoids, where the frequency of the two sinusoids are not a rational multiple of each other. Two tone testing is performed as it is often useful to measure the third order inter-modulation products for two sine-waves of frequency f_1 and f_2 [34]. These products occur at frequencies $2f_1 + f_2$, $2f_2 + f_1$, $2f_1 - f_2$, $2f_2 - f_1$. Most inter-modulation products can be filtered out. However, if the two tones are of similar frequencies, the third order inter-modulation product ($2f_1 - f_2$, $2f_2 - f_1$) will be very close to the fundamental frequencies and cannot be easily filtered [34]. For the two tone testing the frequencies where 1.50390625 MHz and 2.36328125 MHz. The two tone tests help in determining the Spurious Free Dynamic Range (SFDR) which are important in communication applications as it is often desired to know the maximum ratio achievable between the amplitude of the input signal and the amplitude of the maximum spur. As an example,

both the tests were performed on 2 pipe and 4 pipe A/D Converters.

Figure 2.21 shows the FFT of the input, for single tone testing, to a four pipe parallel 16-bit ADC having 1% component error and no opamp nonlinearity. Figure 2.22 shows the FFT of the output of the ADC with no calibration, Figure 2.23 shows the FFT of the output with accuracy bootstrapped pipes and Figure 2.24 shows the FFT of the output with global matching. The SFDR for the ADC is 110 dB. Single tone testing was also performed on a two pipe parallel 16-bit ADC. Figure 2.25 shows the FFT of the single tone input to the ADC. Figure 2.26 shows the FFT of the output with no calibration. Figure 2.27 shows the FFT of the output with accuracy bootstrapped pipes and Figure 2.28 shows the output with global matching. The SFDR for this ADC is 115 dB. Figure 2.29 shows the FFT of the input, for two tone testing, to a four pipe parallel 16-bit ADC having 1% component error and no opamp nonlinearity. Figure 2.30 shows the FFT of the output of the ADC without calibration, Figure 2.31 with accuracy bootstrapped pipes and Figure 2.32 with global matching. The SFDR of the four pipe 16-bit ADC is 110 dB. The two tone testing was then performed on a four pipe parallel 16-bit ADC having 0.1% opamp nonlinearity in addition to the 1% component error. Figure 2.33 shows the FFT of the input, Figure 2.34 shows the FFT of the output of the ADC without calibration, Figure 2.35 with accuracy bootstrapped pipes and Figure 2.36 with global matching. With a 0.1% opamp nonlinearity the SFDR of the four pipe 16-bit ADC falls down to 22 dB.

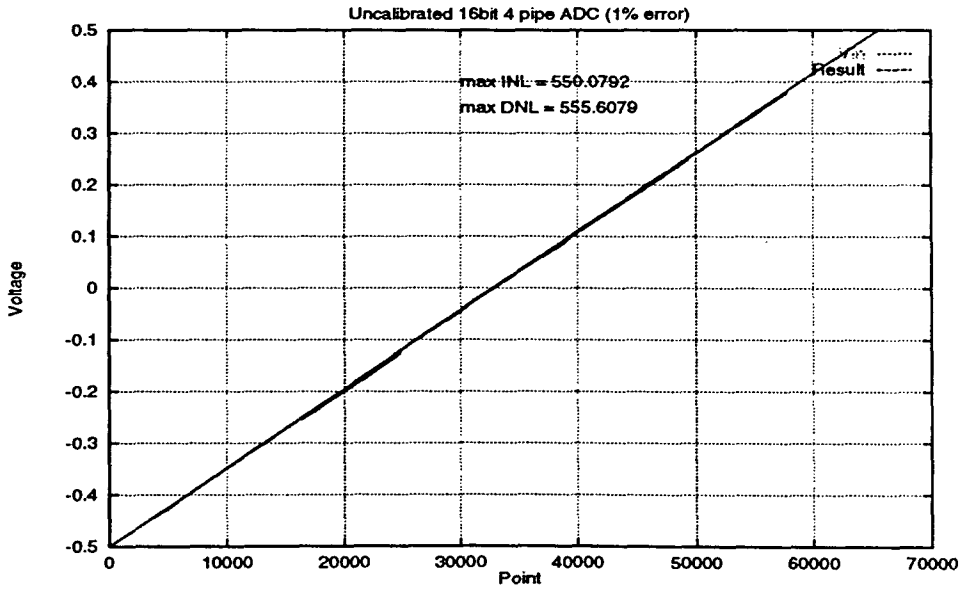

Figure 2.17 Output of the uncalibrated four pipe parallel 16-bit ADC

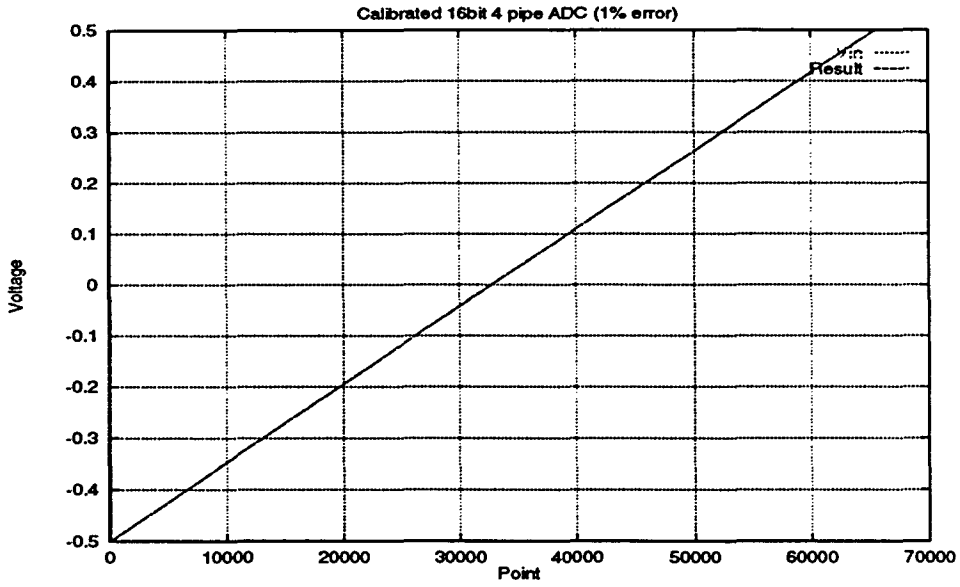

Figure 2.18 Output of the calibrated four pipe parallel 16-bit ADC

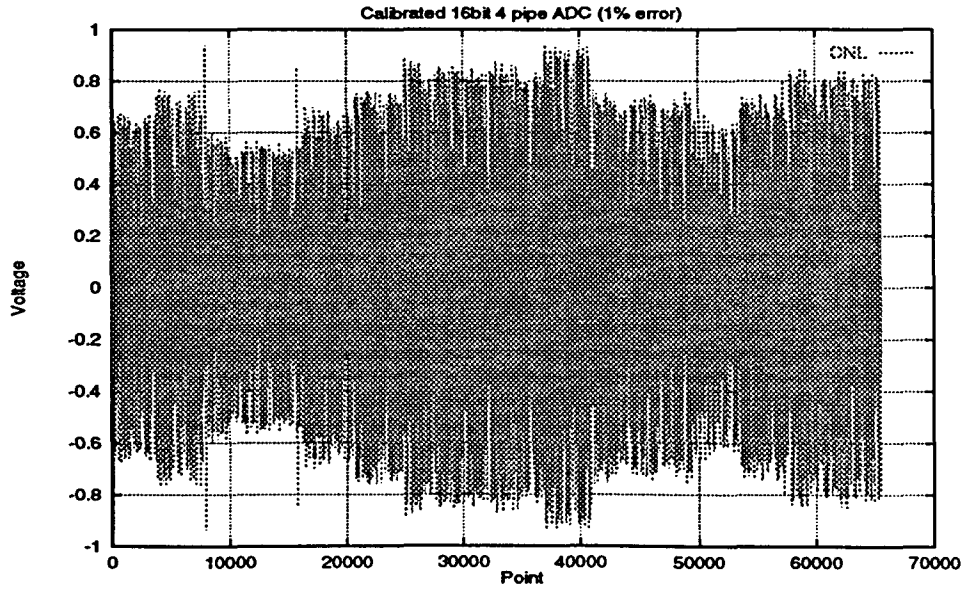

Figure 2.19 DNL of the four pipe parallel 16-bit ADC after calibration

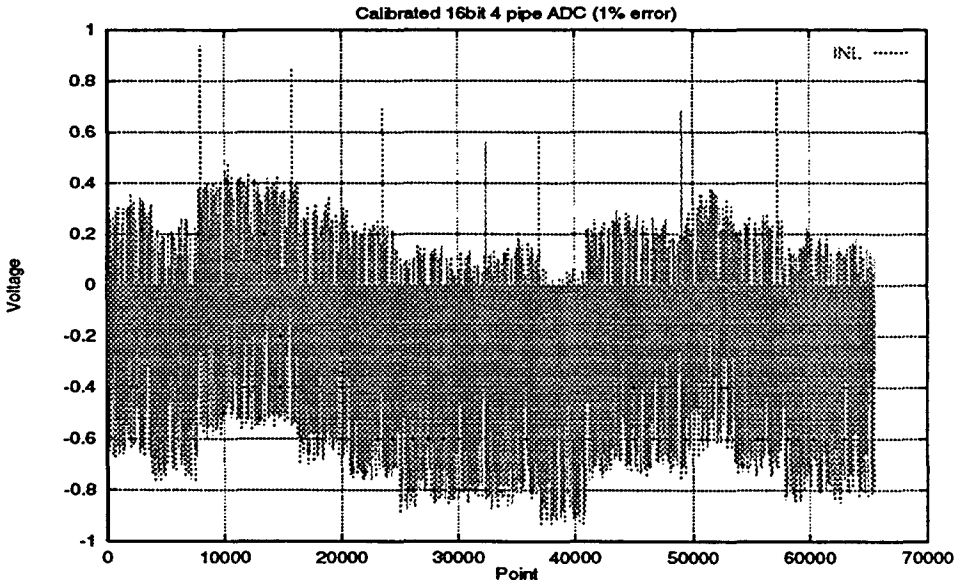

Figure 2.20 INL of the four pipe parallel 16-bit ADC after calibration

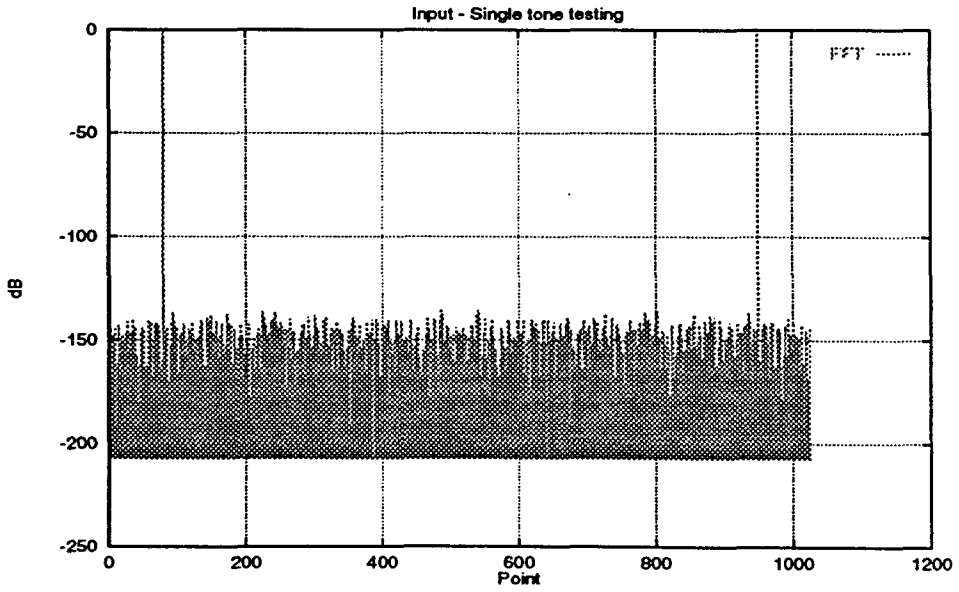

Figure 2.21 FFT of the input with no opamp nonlinearity

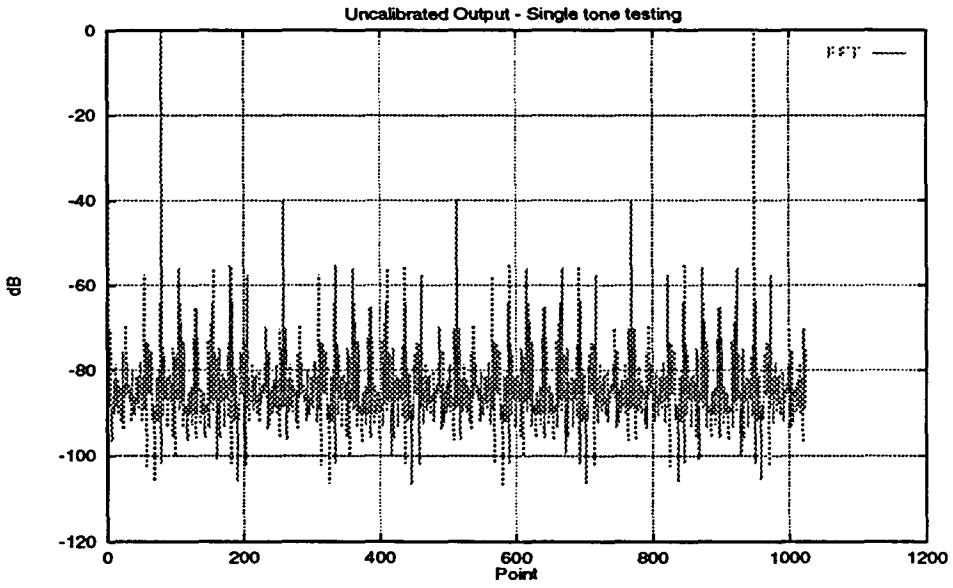

Figure 2.22 FFT of the output with no calibration (no opamp nonlinearity)

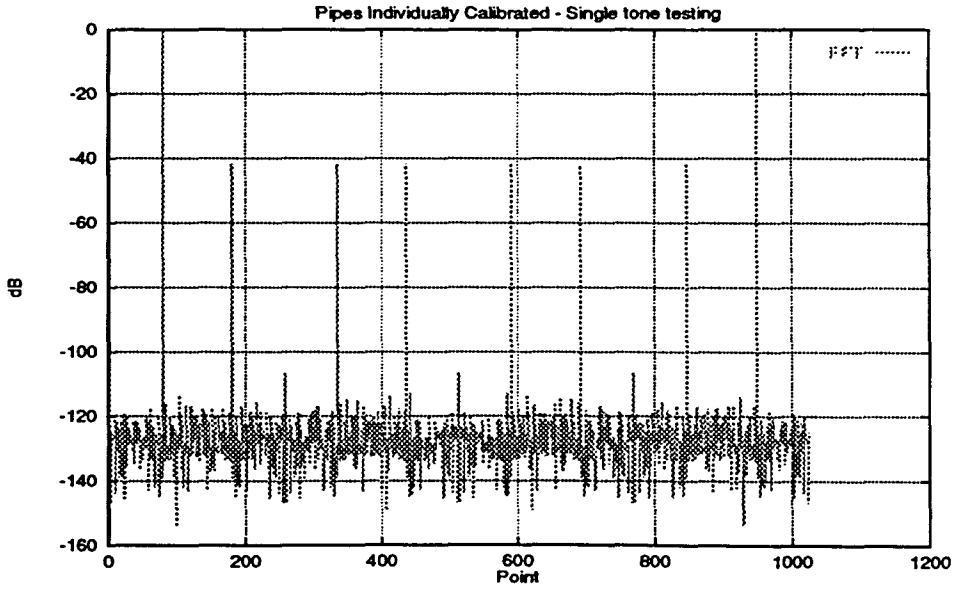

Figure 2.23 FFT of the output with accuracy bootstrapped channels (no opamp nonlinearity)

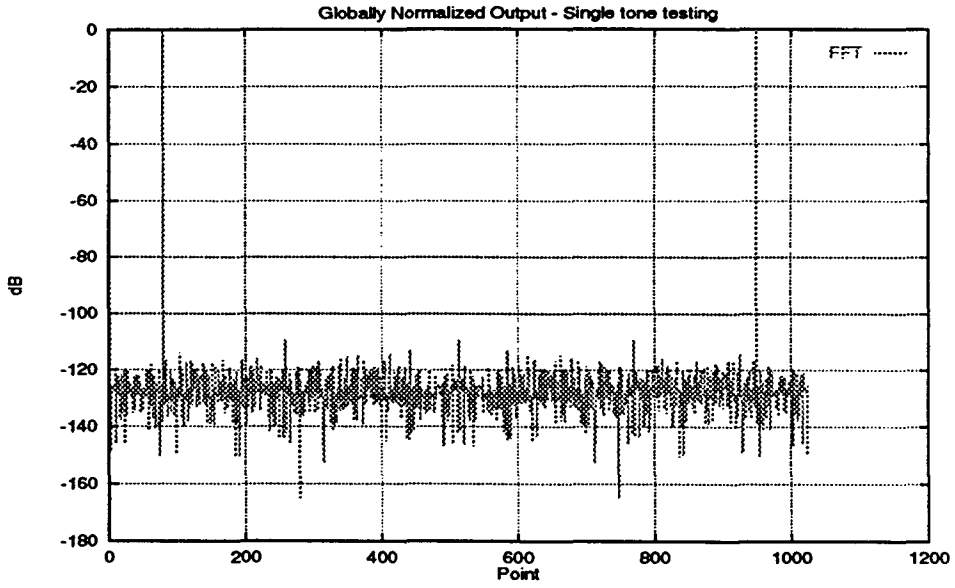

Figure 2.24 FFT of the output with global matching or normalization (no opamp nonlinearity)

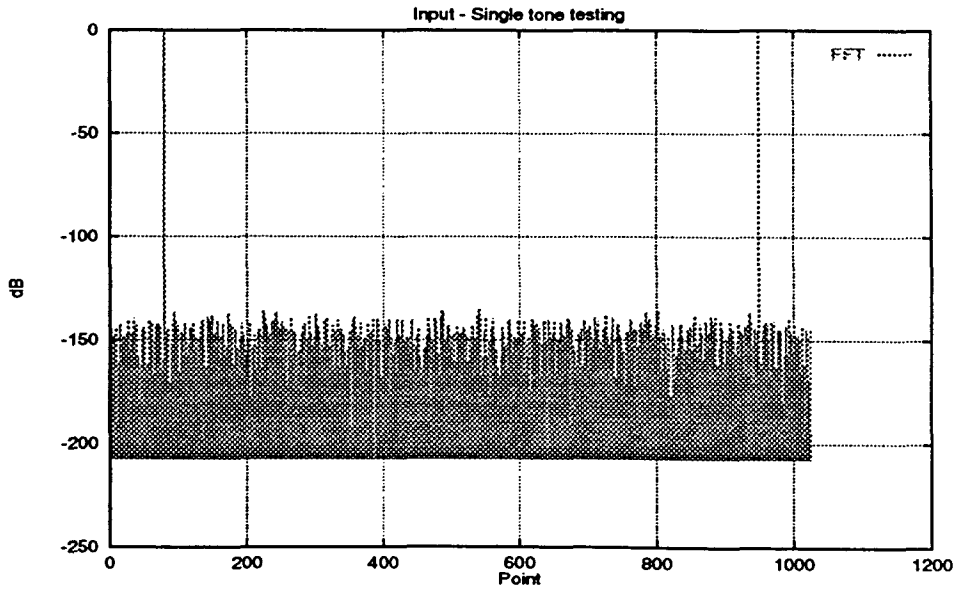

Figure 2.25 FFT of the input with no opamp nonlinearity

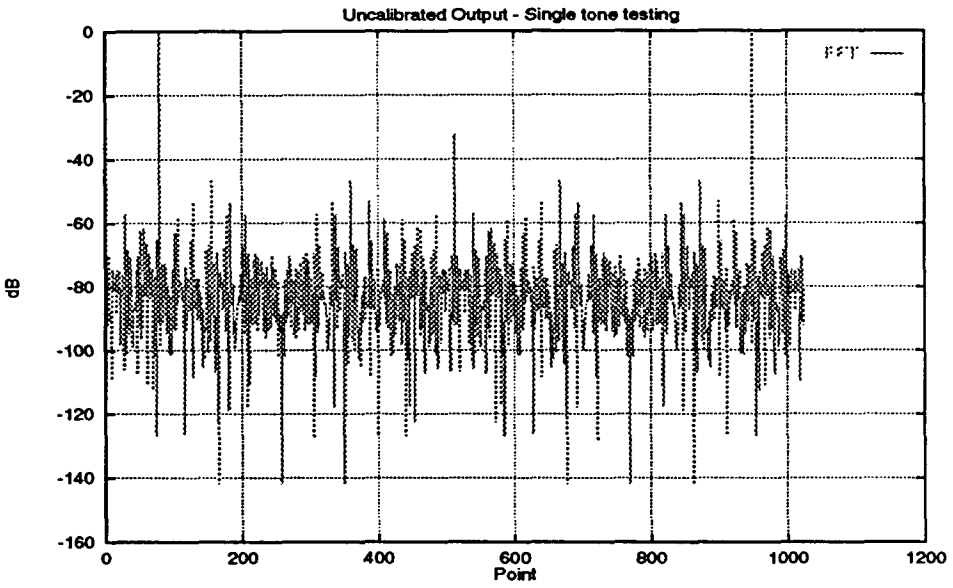

Figure 2.26 FFT of the output with no calibration (no opamp nonlinearity)

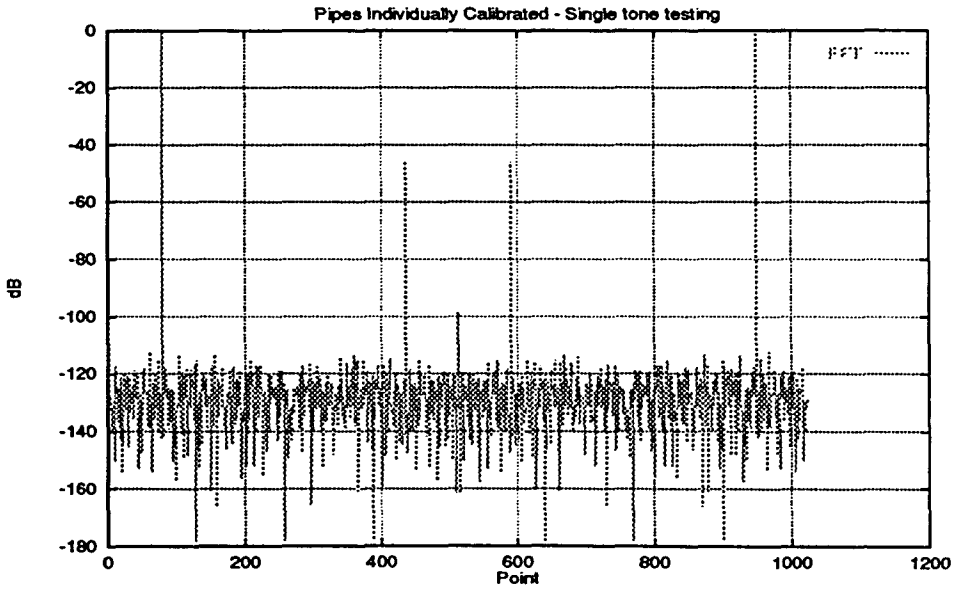

Figure 2.27 FFT of the output with accuracy bootstrapped channels (no opamp nonlinearity)

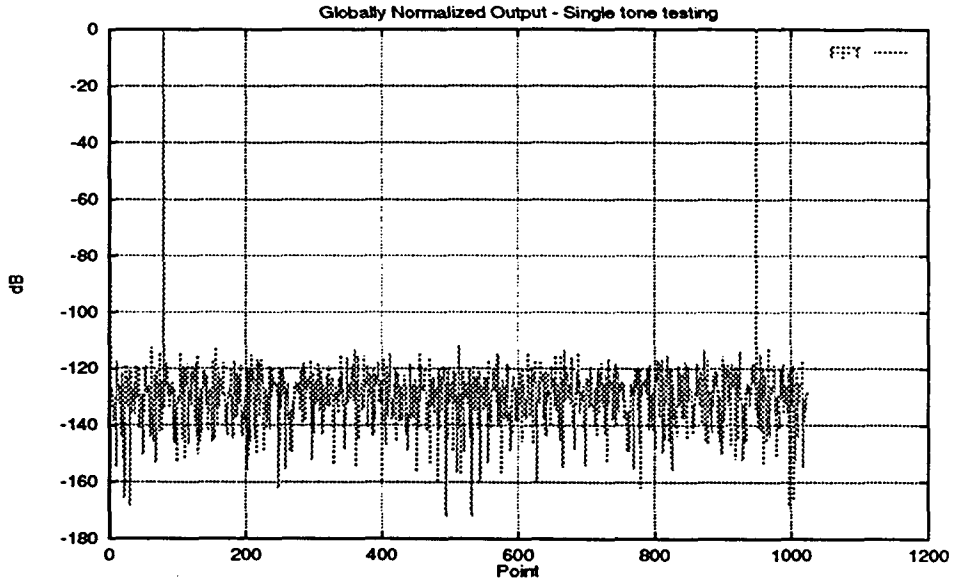

Figure 2.28 FFT of the output with global matching or normalization (no opamp nonlinearity)

2.9 Conclusion

A new architecture for accuracy bootstrapping called "simplified cell architecture" was analyzed and experiments were performed with it. The simplified architecture has less hardware. Each stage has one less voltage source and one less look up table memory. But one extra measurement is needed for calibration compared to previous architecture. Our analysis showed that direct application of accuracy bootstrapping was not advantageous as it does not take into consideration the effect of non-ideal zero. So the calibration was modified so that the nonidealities due to non-ideal zero can be removed. It should be noted that even though one extra measurement is required for calibration, it does not in anyway compromise the speed of the converter. The speed of the converter is, however, reduced due to more complex logic circuitry but it is balanced by the fact that there are only two look-up tables at each stage compared to three in previous architecture.

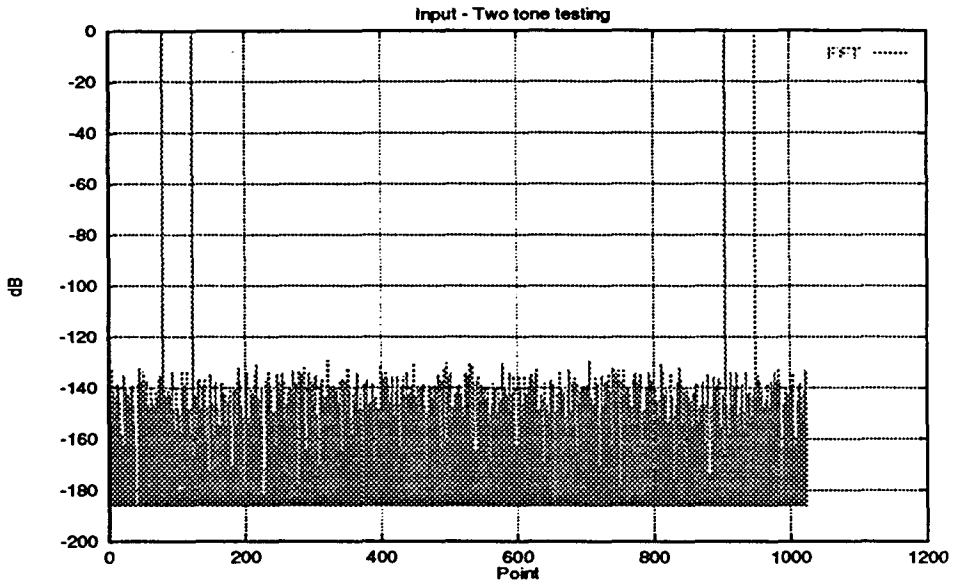

Figure 2.29 FFT of the input with no opamp nonlinearity

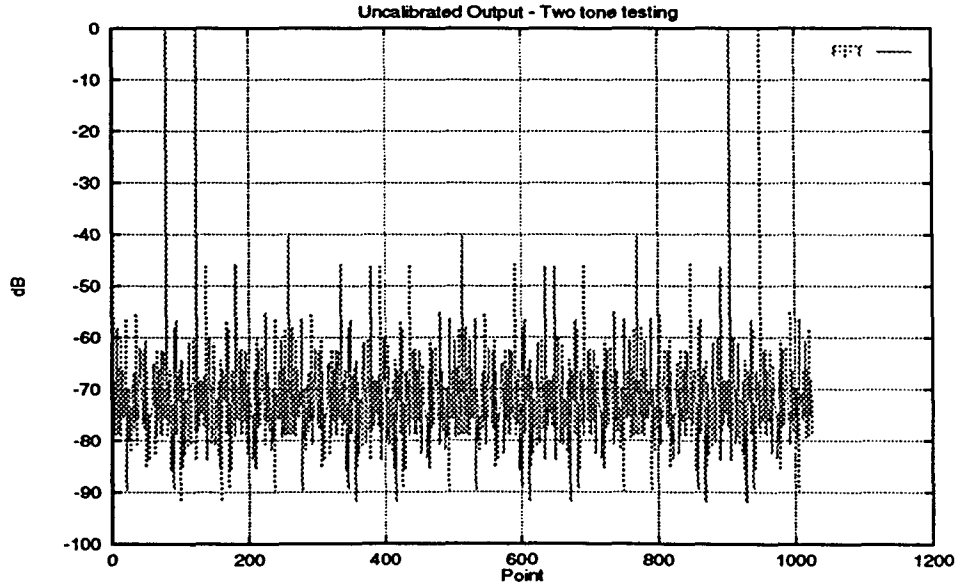

Figure 2.30 FFT of the output with no calibration (no opamp nonlinearity)

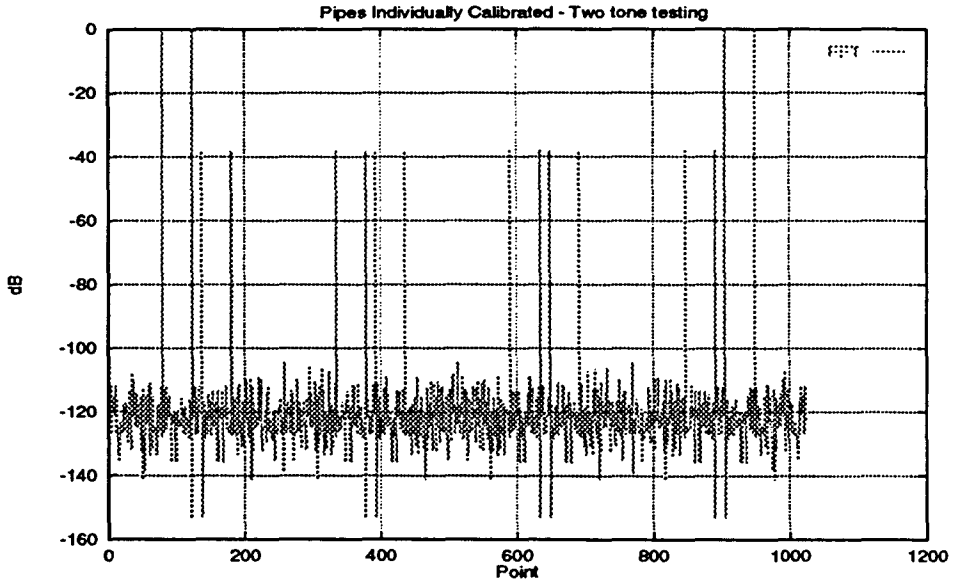

Figure 2.31 FFT of the output with accuracy bootstrapped channels (no opamp nonlinearity)

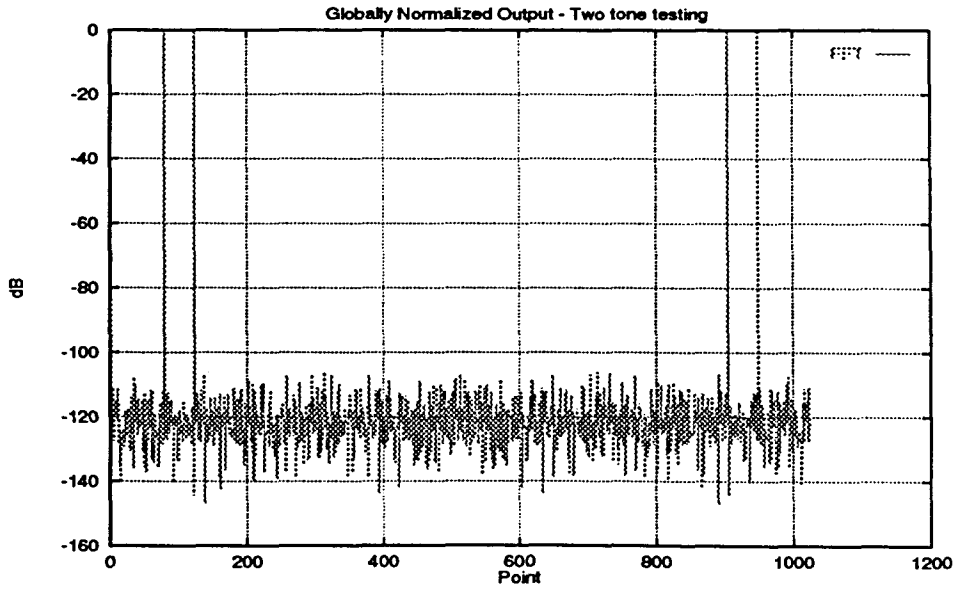

Figure 2.32 FFT of the output with global matching or normalization (no opamp nonlinearity)

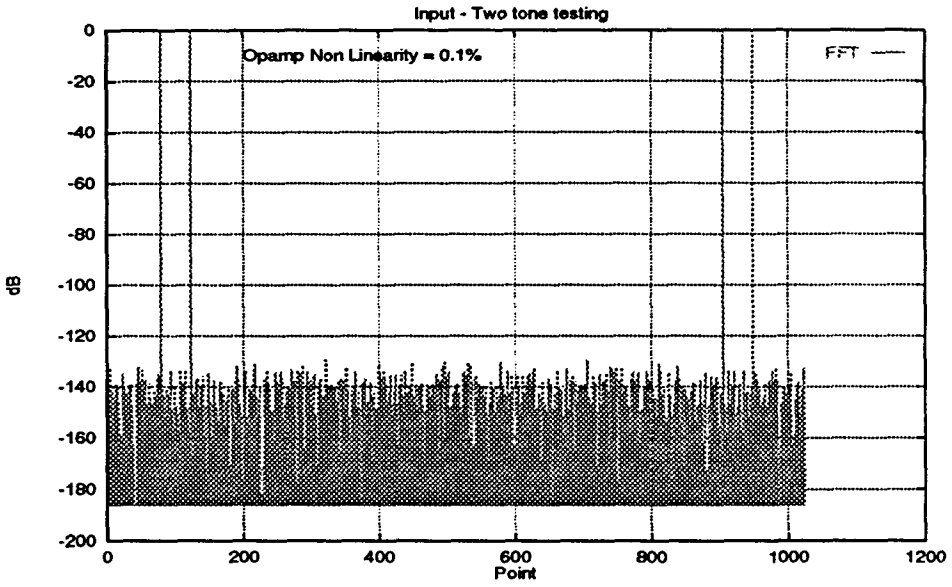

Figure 2.33 FFT of the input (opamp nonlinearity = 0.1%)

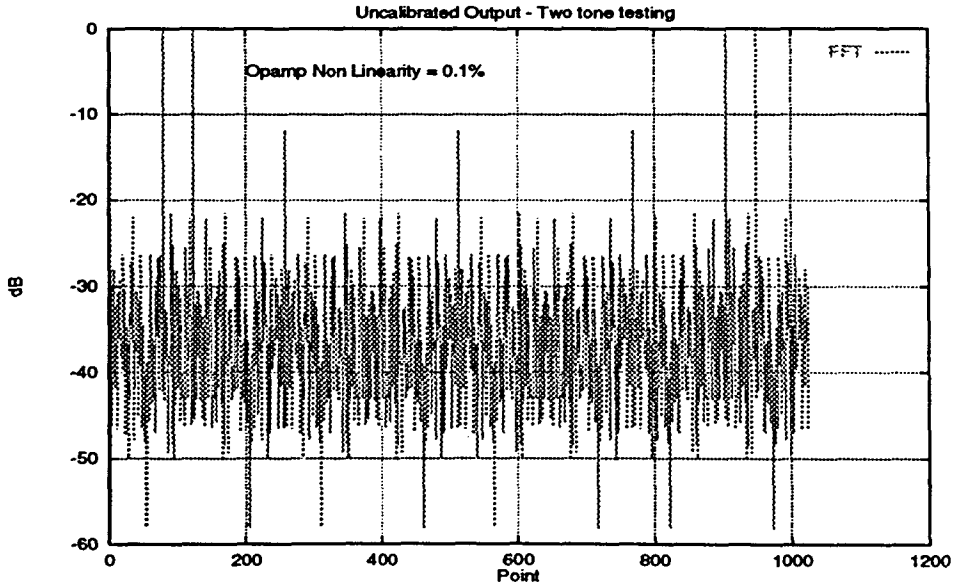

Figure 2.34 FFT of the output with no calibration (opamp nonlinearity = 0.1%)

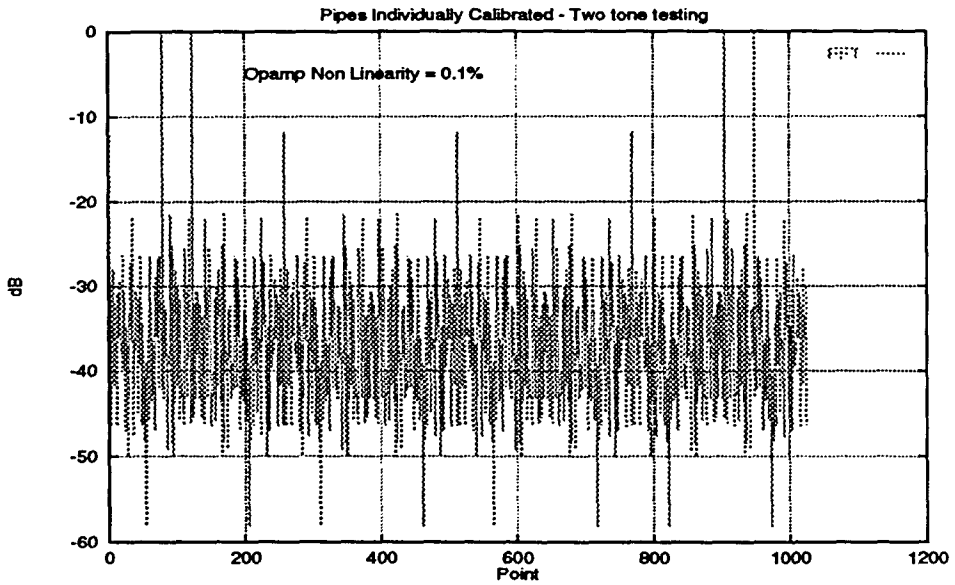

Figure 2.35 FFT of the output with accuracy bootstrapped channels (opamp nonlinearity = 0.1%)

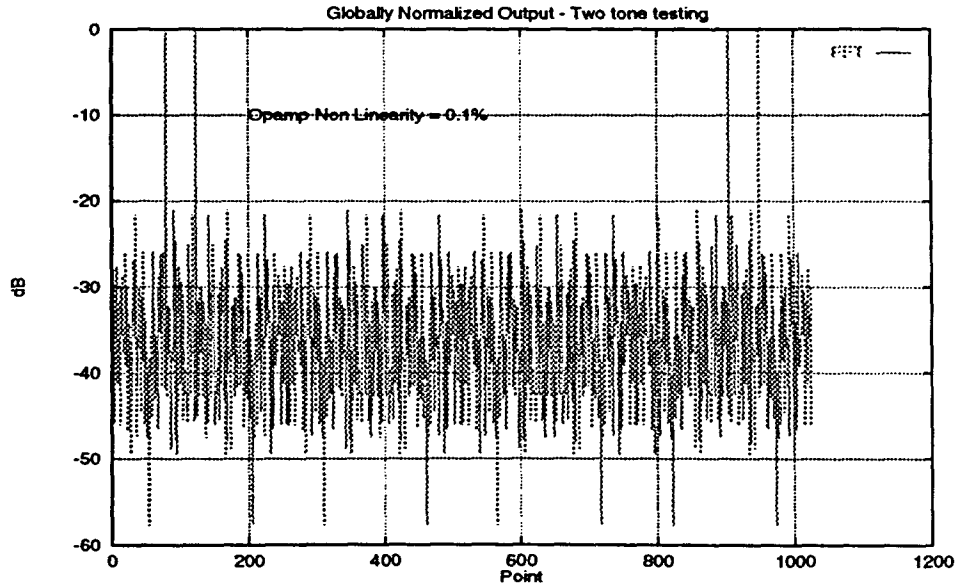

Figure 2.36 FFT of the output with global matching or normalization (opamp nonlinearity = 0.1%)

3 NON-BINARY RADIX ALGORITHM

3.1 Introduction

The algorithm discussed here is a digital self-calibration technique and is based on a radix 1.93 and one comparator per stage conversion algorithm [1]. The digital calibration automatically accounts for some capacitor mismatch, capacitor nonlinearity contributing to DNL, charge injection, finite opamp gain, and comparator offsets. In 1-bit-per-stage pipeline ADC, missing decision levels result when the input of any of the stage exceeds the full scale due to the component nonlinearities. The missing decision levels cannot be removed by digital calibration alone. Missing decision levels can be eliminated, however, by using gain less than 2 and two to three more stages of the pipeline, which gives enough redundancy in the analog decision levels. With gain less than 2, missing codes are introduced, rather than missing decision levels. The missing codes that result with a gain less than 2 are eliminated by digital calibration.

3.2 Architecture of a Single Cell

Figure 3.1 shows the architecture of a single cell. There is a sample-and-hold amplifier which has a gain of 1.93, a comparator which trips at 0 and two DAC levels ($\pm V_{ref}$). V_{in} is the input voltage to the cell, V_{out} the output residue of the cell, D_{in} the input digital bit used to select a reference voltage and D_{out} the output digital bit of the stage. D(i) stands for the output digital word of the converter for the ith stage. The output of the basic cell is given by the following equations:

$$V_{out} = A * V_{in} - V_{ref} \quad if \quad D_{in} = 0 \tag{3.1}$$

$$A * V_{in} + V_{ref} \quad if \quad D_{in} = 1 \tag{3.2}$$

where A is the gain of the amplifier.

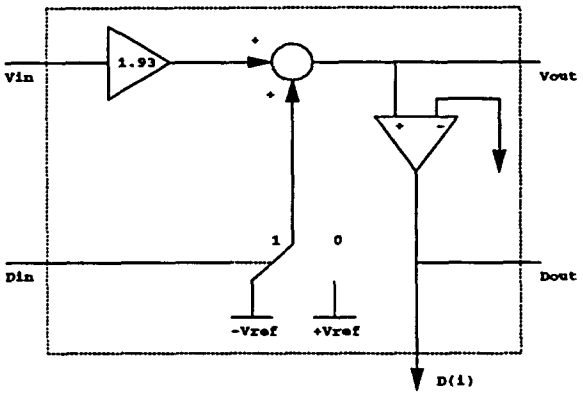

Figure 3.1 Architecture of the single cell for non-binary radix algorithm

3.3 Digital Output Reduction Method

Since a non-radix-2 architecture has non-uniform decision levels, a uniform decision level spacing no wider than 1 LSB must be defined for calculating INL and DNL, at the intended resolution. The concept of digital output reduction is shown in the Figure 3.2. The ideal unreduced and actual unreduced decision level locations are shown first. The ideal reduced and actual reduced decision level are shown next. It is clear from Figure 3.2 that actual reduced decision level spacing is more uniform than actual unreduced decision level. This improves the INL and DNL values. When using this method, the step size of the linear sweep needs to be smaller then 1 LSB. Every input for which the output code changes indicates a decision level. This is the unreduced decision level which is not uniform. The unreduced decision levels are then averaged by a factor determined by the user. In the calculations performed in this thesis, the averaging factor was 4. It is not necessary that the averaging factor be a power of 2. Mathematically it can be written as [33]:

$$RDNL(k) = \frac{1}{L}\sum_{i=kL}^{(k+1)L-1} DNL(i) \tag{3.3}$$

$$RINL(k) = \frac{1}{L} INL((k+1)L-1) \tag{3.4}$$

where L is the averaging factor, DNL(k) and INL(k) are the actual unreduced DNL and INL, RDNL(k) and RINL(k) are the actual reduced DNL and INL.

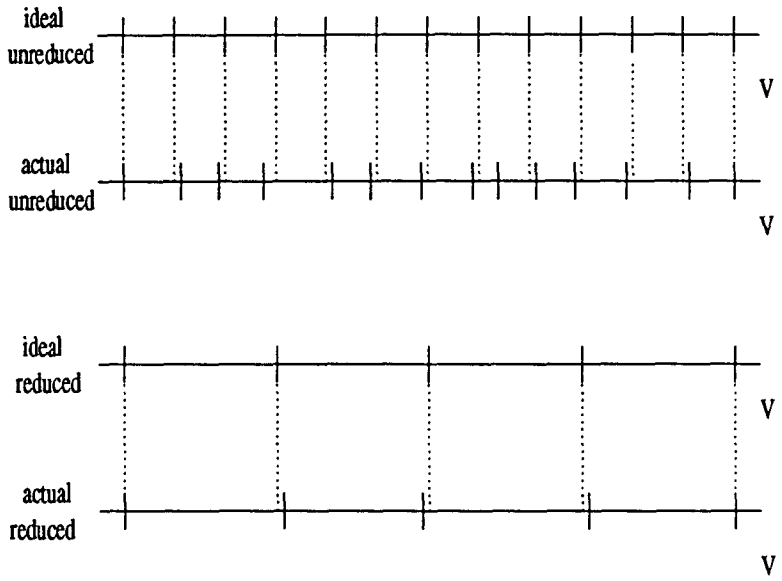

Figure 3.2 Decision level locations for unreduced and reduced ADC output

3.4 Derivation of the Digital Self Calibration Algorithm

Figure 3.3 shows an ideal pipeline with all the stages having a gain of 2 [1]. The idealized structure is shown with an analog input V_{in}, a digital input D_{in}, an analog output V_{out} and a digital output D_{out}. The output bit D_{out} when concatenated to resultant digital code of the last six stages forms quantity X. This quantity X represents the quantization of the residue V_{out}. The digital input D_{in} has the quantity X concatenated to result in the quantity $D_{in} : X$. This quantity $D_{in} : X$ represents the quantization of the input V_{in}. The residue plot is shown for two cases $D_{in} = 0$ and $D_{in} = 1$. The transfer characteristic is also shown with $D_{in} : X$ plotted against V_{in}. In this ideal pipeline ADC, the transfer characteristics are perfectly linear.

In Figure 3.4, the input stage of Figure 3.3 is replaced by an amplifier of gain G < 2 and the remaining last 6 stages have a gain, G $= 2$. The residue plot of this structure clearly indicates that the residue V_{out} is contained within the reference boundary box. The transfer characteristic is composed of two linear regions, corresponding to $D_{in} = 0$ and $D_{in} = 1$, respectively, that are disjoint near $V_{in} = 0$. Notice that there is a sudden change in code near $V_{in} = 0$. As a result, missing codes result in this region.

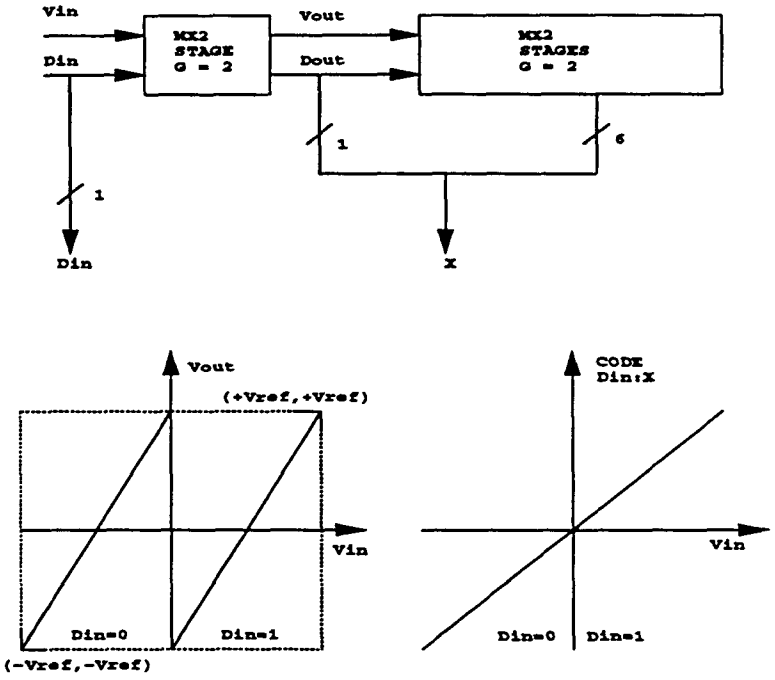

Figure 3.3 Ideal residue and transfer characteristic of a pipeline ADC

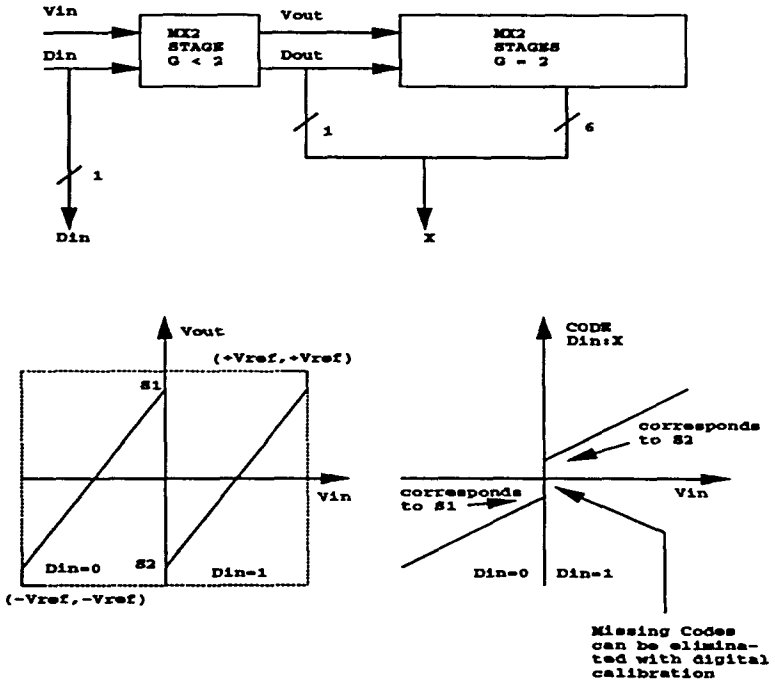

Figure 3.4 Residue and transfer characteristic of a pipeline ADC with missing codes

Table 3.1 Digital output of Vin with all stages having a gain of G = 2

Vin	Vout	Din	X	Din:X
+Vref	+Vref	1	127	255
.	.	1	126	254
.	.	.	.	.
.	.	1	1	129
0	-Vref	1	0	128
0	+Vref	0	127	127
.	.	0	126	126
.	.	.	.	.
.	.	0	1	1
-Vref	-Vref	0	0	0

Table 3.1 illustrates the output parameters of an ideal pipeline section with G = 2 as a function of V_{in} [33]. This table corresponds to Figure 3.3. Notice that X repeats for $D_{in} = 0$ and $D_{in} = 1$. This is because X represents the quantization of the residue V_{out}. The key point is to note that the output $D_{in} : X$ has no missing codes and count from 0 to $2^8 - 1 = 255$.

Table 3.2 Digital output of Vin with input stage G < 2

Vin	Vout	Din	X	Din:X
+Vref	< +Vref	1	125	253
.	.	1	124	252
.	.	.	.	.
.	.	1	3	131
0	> -Vref	1	2	130
0	< +Vref	0	125	125
.	.	0	124	124
.	.	.	.	.
.	.	0	3	3
-Vref	> -Vref	0	2	2

Table 3.2 illustrates the output parameters of a pipeline section with the input stage G < 2 as a function of V_{in} [33]. This table corresponds to Figure 3.4. In this case, it is seen that X does not reach the maximum or minimum value possible because the gain is reduced in the input stage. Thus $D_{in} : X$ has missing codes at the major carry transition point. This observation is central to the digital

self-calibration scheme. The quantity $D_{in} : X$ can be remapped to eliminate the missing codes without affecting decision levels. The interpretation of the output $D_{in} : X$ is that the input stage structure joins the two halfs of the residue plot together. Thus, the decision level spacing is essentially not affected. The reduced gain of the input stage means only that fewer than the maximum possible number of decision levels are accessed.

Table 3.3 Digital output of Vin with input stage G < 2 and digital calibration applied.

Vin	Vout	Din	X	Din:X	Y
+Vref	< +Vref	1	125	253	248
.	.	1	124	252	247
.	.	.	.	.	.
.	.	1	3	131	126
0	> -Vref	1	2	130	125
0	< +Vref	0	125	125	125
.	.	0	124	124	124
.	.	.	.	.	.
.	.	0	3	3	3
-Vref	> -Vref	0	2	2	2

Table 3.3 illustrates the output parameters of a pipeline section with input stage G < 2 and digital self-calibration system employed as a function of V_{in} [33]. This table corresponds to Figure 3.5. The digital calibration logic accepts two inputs D_{in} and X. The output Y is the calibrated output. The digital calibration logic accepts two inputs S_1 and S_2 which are digitized residue extrema indicated in Figure 3.5. Notice that the calibrated output has no missing codes. This is performed by arranging the output Y to count with X for $D_{in} = 0$ and to continue counting for $D_{in} = 1$. This eliminates the missing codes.

3.4.1 The Digital Self Calibration Algorithm

The digital self-calibration algorithm can now be simply stated as [1], [33]:

$$Y = X, \quad if \quad D_{in} = 0 \tag{3.5}$$

$$Y = X + S_1 - S_2, \quad if \quad D_{in} = 1 \tag{3.6}$$

There is no explicit digital multiplication needed for calibration. From Table 3.3, it is seen that

$S_1 = 125$ and $S_2 = 2$. As a result, $S_1 - S_2 = 123$. With $D_{in} = 0$, the output Y counts from 2 to 125. When $D_{in} = 1$, the output Y becomes X + 123 which begins at 125. The output continues to 248. If desired, the offset can be removed at this stage by determining $S_0 = 2$ and subtracting S_0 from Y. This would result in an output range from 0 to 246.

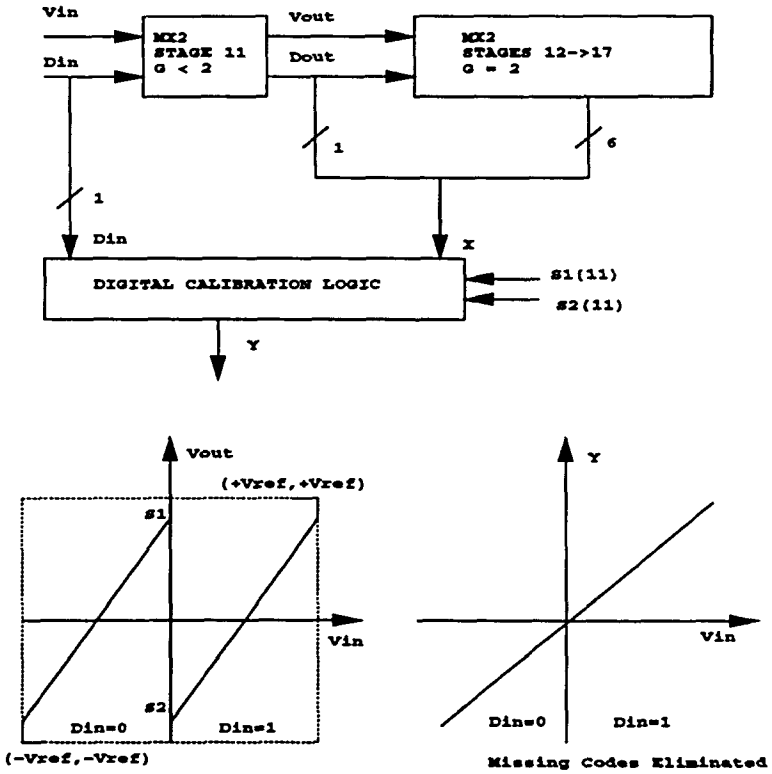

Figure 3.5 Pipeline ADC with digital calibration applied to the eleventh stage

In order to obtain S_1 for a particular stage, the analog input is grounded and the input bit is forced to be 0 and the residue from the stage is quantized using the successive pipeline stages. In order to obtain S_2 for a particular stage, the analog input is grounded and the input bit is forced to be 1 and the residue from the stage is quantized using the successive pipeline stages.

With the digital calibration of the one G < 2 stage accomplished, the digital calibration of higher levels can proceed as shown in Figure 3.6. The system within the dashed box has been of principal interest thus far. The last six stages with nominal gain of 2 and the digital calibration logic were used to calibrate stage 11. The calibrated system within the dashed box is now used as an ADC to measure stage 10. Calibration constants S_1 and S_2 for stage 10 are determined in a similar manner as for stage 11. The digital calibration logic system for stage 10 follows the same algorithm as for stage 11. The

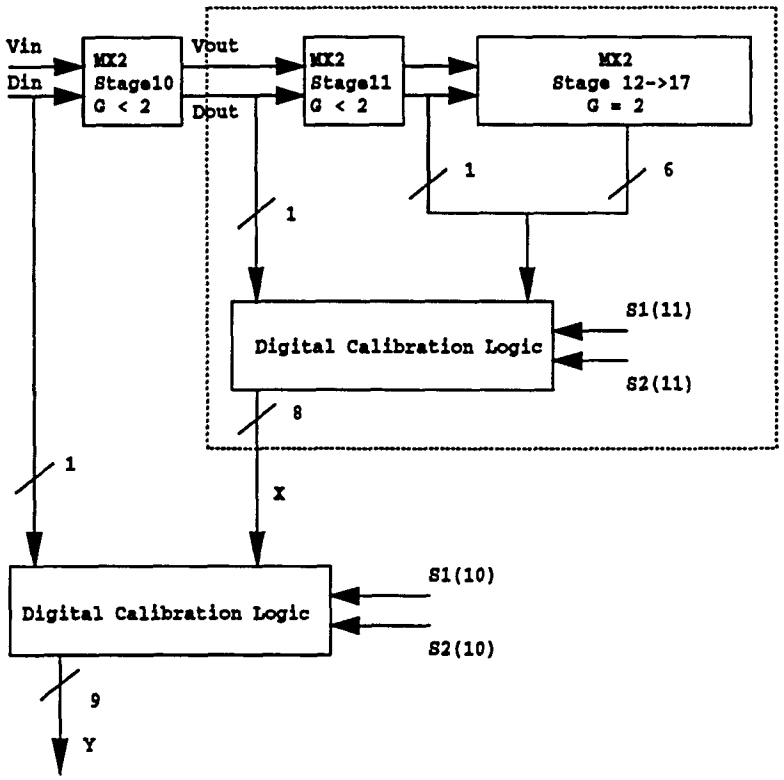

Figure 3.6 Digital calibration of higher level stages

calibrated system leading with stage 10 can then be used to calibrate stage 9. This process continues to the first stage. In this way the entire pipeline is calibrated.

3.5 Accuracy Bootstrapped Digital Self Calibration (ABDiSC)

In the digital self-calibration algorithm described in the previous section, when a stage is being calibrated, only those stages which are downstream of the calibration stage are being used. In accuracy bootstrapping, the algorithm makes use of all the stages for calibration i.e all the stages of the pipeline are being used for calibration and not just the downstream stages. Since all the stages are being used, the calibration results for each stage are more accurate thereby improving the overall linearity of the ADC. By bootstrapping the last stage of the pipeline to the first stage, all the stages of the pipeline can be used. This is the basic difference between the digital self-calibration and ABDiSC. Notice that the only expense on this account are some switches which are needed to connect the last stage to the first.

3.5.1 The ABDiSC Algorithm

The new algorithm for calibration is as shown below:

if (*calibrating stages with gain = 2*)

{

switch all the stages to gain = 2;

apply accuracy bootstrapping;

update the weights;

}

else if (*calibrating stages with gain < 2*)

{

switch all stages upstream of the stage being calibrated to gain = 2;

apply accuracy bootstrapping;

update the weights;

}

When the stages with gain = 2 are being calibrated, their weights need to be as close as possible to $\pm V_{ref}$. Therefore all the stages need to have a gain of 2 as shown in Figure 3.7. When the stages with gain < 2 are being calibrated, those stages with gain < 2 and are upstream of the stage being calibrated should be switched to a gain of 2. Those stages which are downstream of the stage being calibrated, and have a gain < 2 should not be switched to a gain of 2 as shown in Figure 3.8.

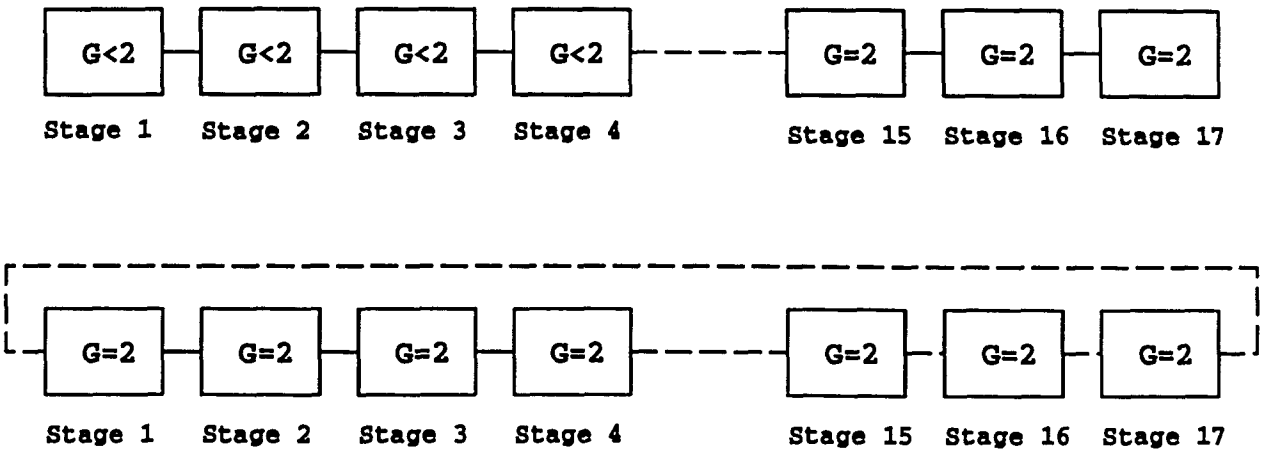

Figure 3.7 Calibration of stage with gain of 2

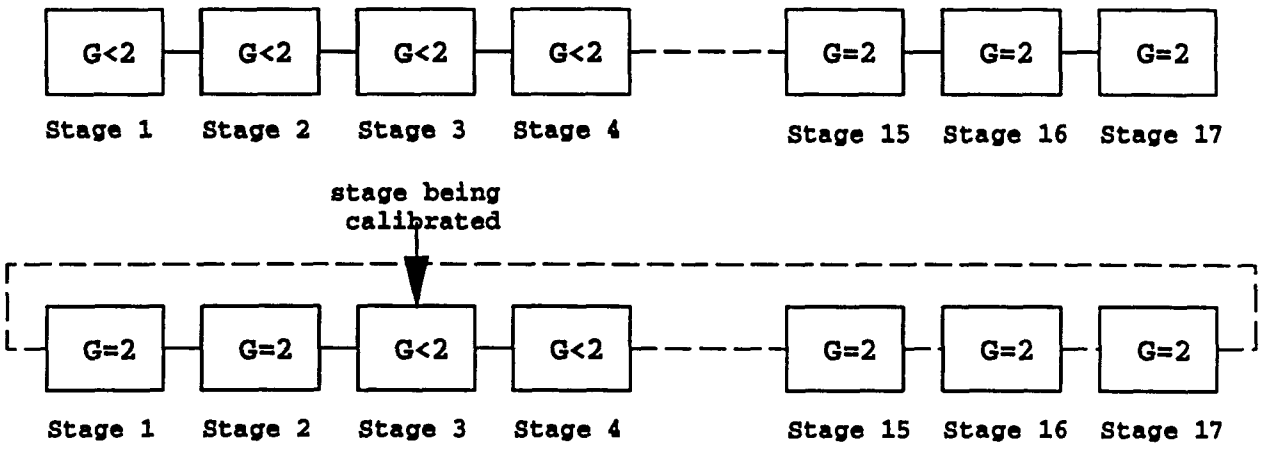

Figure 3.8 Calibration of stage with gain < 2

3.5.2 Proof of the Algorithm

The proof of the algorithm is a direct consequence of the results derived in Section 2.2.2 and were originally derived in [25]. It is presented here again for the purpose of completeness.

Let $+V_{ref}$ and $-V_{ref}$ be represented by V_{ref1} and V_{ref2}. The input voltage to the ADC can be written as:

$$V_{in} = \sum_{some\ i(D=1)} V_{ref1}\left(\frac{1}{A^i}\right) + \sum_{other\ i(D=0)} V_{ref2}\left(\frac{1}{A^i}\right) \tag{3.7}$$

$$= \sum_{some\ i(D=1)} (V_{ref1} - V_{ref2})\left(\frac{1}{A^i}\right) + \sum_{all\ i} V_{ref2}\left(\frac{1}{A^i}\right) \tag{3.8}$$

which can be expressed as

$$Ideal\ Result = \sum_{some\ i(D=1)} (V_{ref1} - V_{ref2})\left(\frac{1}{A^i}\right) - constant \tag{3.9}$$

where A^i is the product $A_0A_1...A_{L-l}$ since i=L-l.

Considering the first stage,

$$\frac{S_1 - S_2}{A^{L-1}} = \frac{V_{ref1} - V_{ref2}}{A^{L-1}} \tag{3.10}$$

and comparing it with the ideal result $\frac{V_{ref1}-V_{ref2}}{A_{L-1}...A_0}$, the overestimation is $\frac{A^{L-1}}{A_{L-1}...A_0}$. Call the overestimation as k.

Considering the second stage,

$$\frac{S_1 - S_2}{A^{L-2}} = \frac{V_{ref1} - V_{ref2}}{A^{L-2}} \tag{3.11}$$

We see that the overestimation in the second stage is due to the fact the weights of stage 0 are being used for calibration and the overestimation that will occur because of using the nominal value of the gain .i.e A. The overestimation due to stage 0 is $\frac{A}{A_0}$. The overestimation due to the difference from ideal weights is $\frac{A^{L-2}}{A_{L-1}...A_2A_1}$. Therefore the overestimation is k for all the stages. The result then is

$$\sum_i (S_1 - S_2)\left(\frac{1}{A^i}\right) = \sum_{some\ i(D=1)} (V_{ref1} - V_{ref2})\left(\frac{1}{A^i}\right) \tag{3.12}$$

$$= k(ideal\ result + constant) \tag{3.13}$$

$$= k(ideal\ result + offset) \tag{3.14}$$

Thus, the digital output of the ADC is

$$Digital\ Output = \sum_{all\ terms} (S_1 - S_2)\left(\frac{1}{A^i}\right) \tag{3.15}$$

3.5.3 An Intuitive Explanation

As can be seen in Table 3.2, at the major carry transition point the digital code changes by more than an lsb. If we look at the individual bits, then at each major carry transition point the MSB changes from '0' to '1'. The digital code for 125 at 8-bits is "01111101" and for 130 is "10000010". The 8-bit ADC is linear for the last seven stages. Hence the weight associated with each bit for those last seven stages should be $\pm V_{ref}$. But for the MSB or the 8th bit the weight has to be less than $-V_{ref}-$ so as to compensate for the jump in code. As a result the weight assigned to the 8th bit is less than $-V_{ref}-$. The weight that is assigned to $W[0]$ for bit '0' is obtained by measuring the value of $-V_{ref}$ using the pipeline and the weight assigned to $W[1]$ for bit '1' is obtained by measuring $+V_{ref}$ using the pipeline. Due to the gain < 2 stages the values of the of $\pm V_{ref}$ are going to come out less than $-V_{ref}-$, which is what it should be. As we continue this process for higher number of bits the weight assigned to each higher stage (i.e. upstream stage) is going to decrease. When all the stages are calibrated the updated weights can adjust the jumps in codes and the whole pipeline becomes linear.

3.6 Conclusion

In this chapter an introduction to digital self-calibration for non-binary radix architecture was given. An explanation of how the algorithm eliminates the missing codes was detailed. A new algorithm which uses both the concepts of digital self-calibration and accuracy bootstrapping called ABDiSC was proposed and was shown to achieve better accuracy.

4 EXPERIMENTAL RESULTS

The most important specifications of an ADC are the INL and DNL specifications. The offset and gain specifications are not critical as they as they can be removed, either by hardware or software. INL and DNL represent irreducible errors. The linearity of an A/D converter can be found by performing a linear sweep which gives its DNL and INL characteristics. The DNL is a measure of the ADCs resolution whereas the INL is a measure of the overall accuracy of the ADC. For communication applications, it is important to determine the performance of the ADC in frequency domain. The frequency domain specifications of the ADC can be obtained by single and two tone testing.

4.1 Results of the Linear Sweep

A linear sweep was performed on a 16 stage A/D converter having 10 stages of G = 1.93 and 6 stages of G = 2. The results were obtained for both digital self-calibration and ABDiSC. Figure 4.1 shows the transfer characteristic of the converter with 1% component error (described in section 1.3 in Chapter 1). Figure 4.2 shows the transfer characteristic after the converter was calibrated using digital self-calibration. Figure 4.3 and Figure 4.4 show the DNL and INL plots with digital self-calibration. The maximum value of the DNL is 0.52 LSB and the maximum value of INL is 1.21 LSB. Figure 4.5 shows the transfer characteristic of the converter after it was calibrated using ABDiSC. Figure 4.6 shows the DNL plot of the ABDiSC calibrated converter and Figure 4.7 shows the INL plot. The maximum value of DNL is -0.35 LSB and the maximum value of INL is 0.55 LSB. The tests were repeated for another 16 stage ADC with 2% component errors. Figure 4.8 shows the transfer characteristic of the converter calibrated with digital self-calibration, Figure 4.9 shows the DNL plot and Figure 4.10 shows the INL plot. The maximum value of DNL is 1.45 LSB and the maximum value of INL is 2 LSB. The transfer characteristic of the converter after it was calibrated using ABDiSC is shown in Figure 4.11. The corresponding DNL plot is shown in Figure 4.12 and the INL plot is shown in Figure 4.13. The maximum value of DNL is 0.35 LSB and the maximum value of INL is 1.05 LSB. These figures clearly demonstrate the fact that ABDiSC performs better than digital self-calibration.

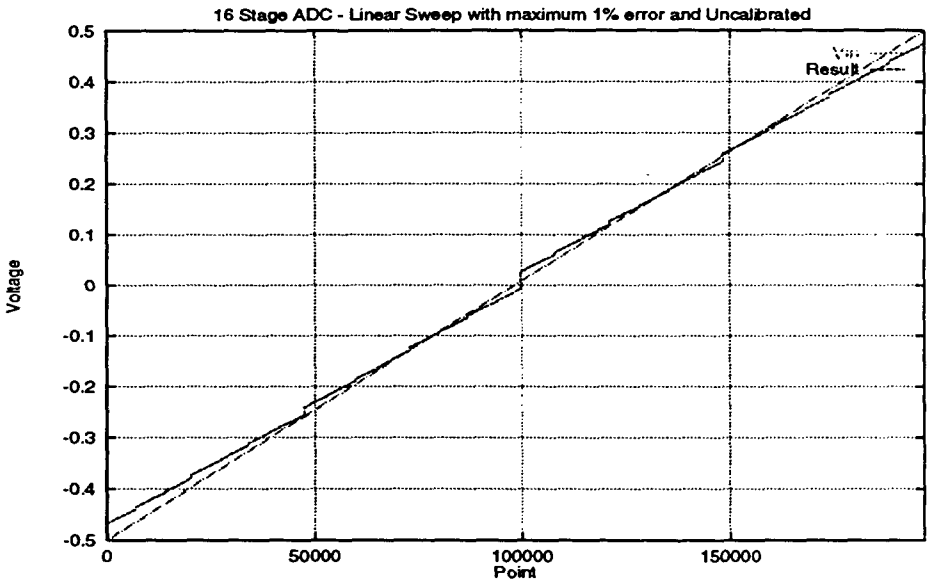

Figure 4.1 Linear sweep of an uncalibrated 16 stage ADC with 1% error

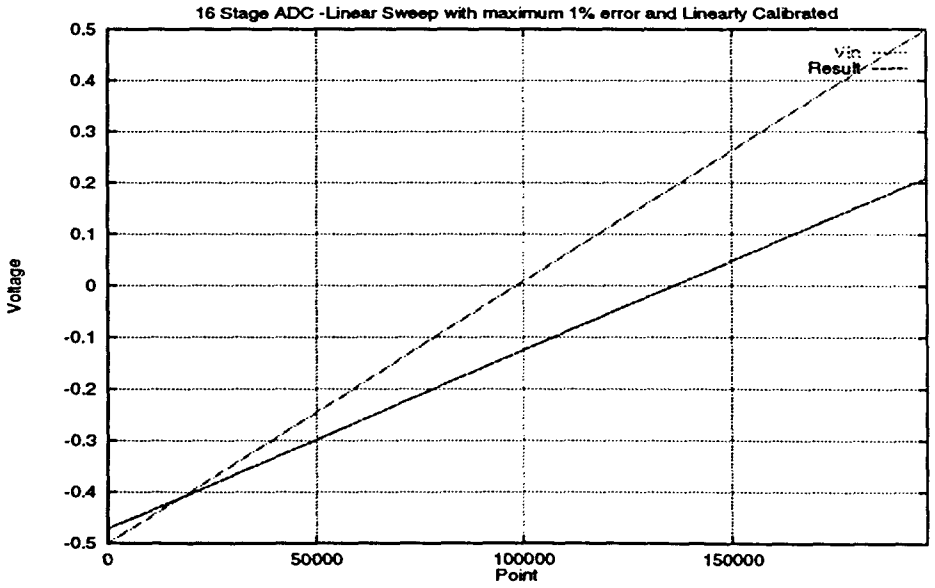

Figure 4.2 Linear sweep of a 16 stage ADC with 1% error and digitally self-calibrated

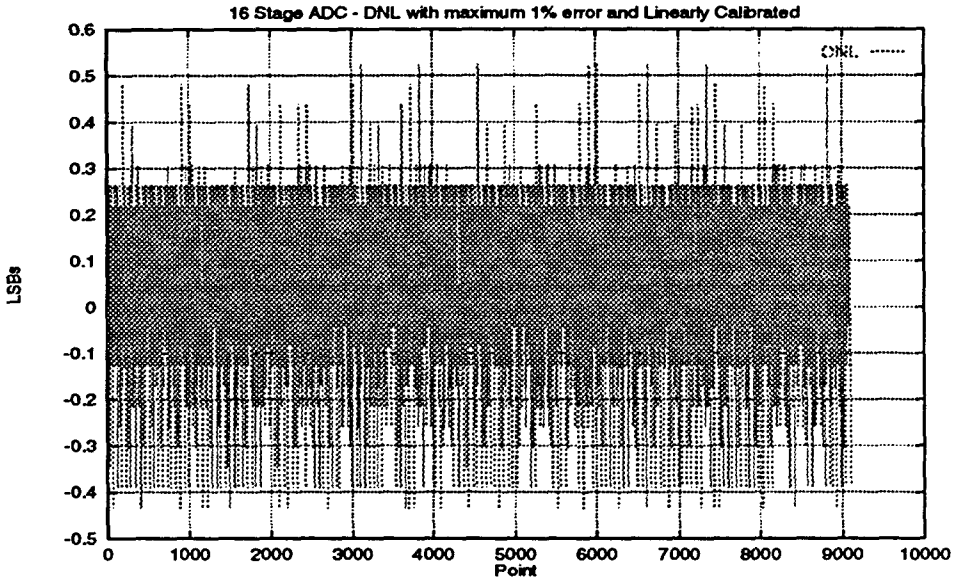

Figure 4.3 DNL of a 16 stage ADC with 1% error and digitally self-calibrated

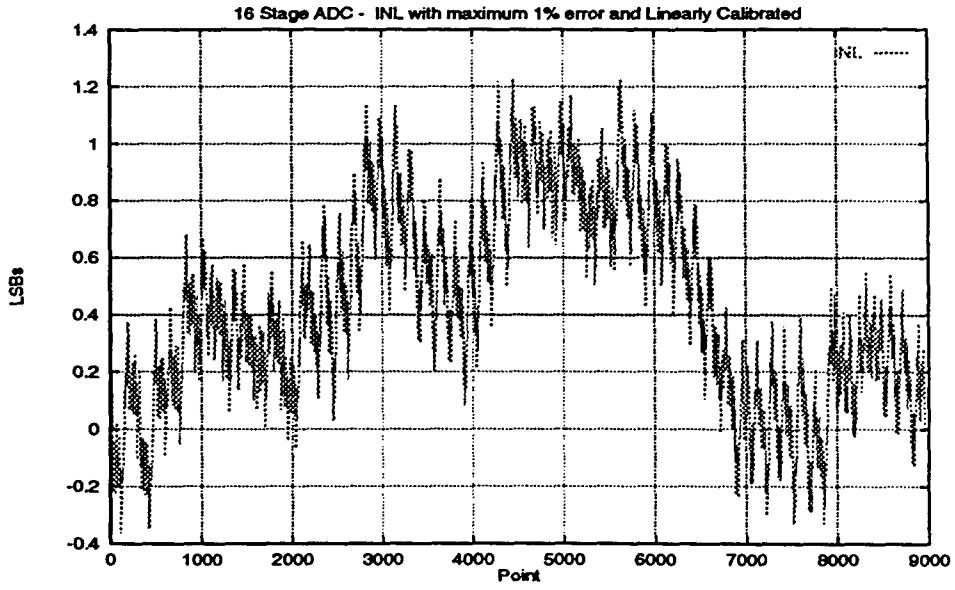

Figure 4.4 INL of a 16 stage ADC with 1% error and digitally-self-calibrated

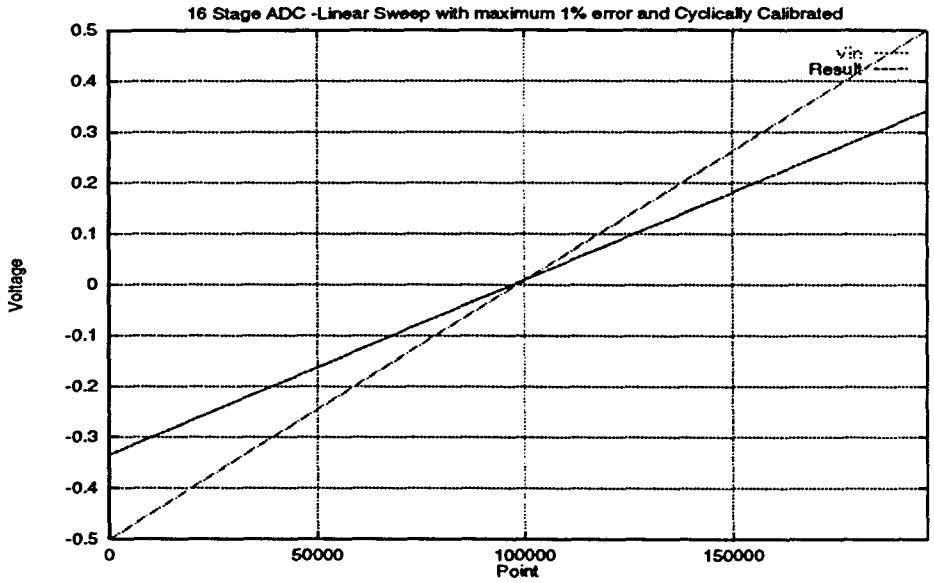

Figure 4.5 Linear sweep of a 16 stage ADC with 1% error and ABDiSC calibrated

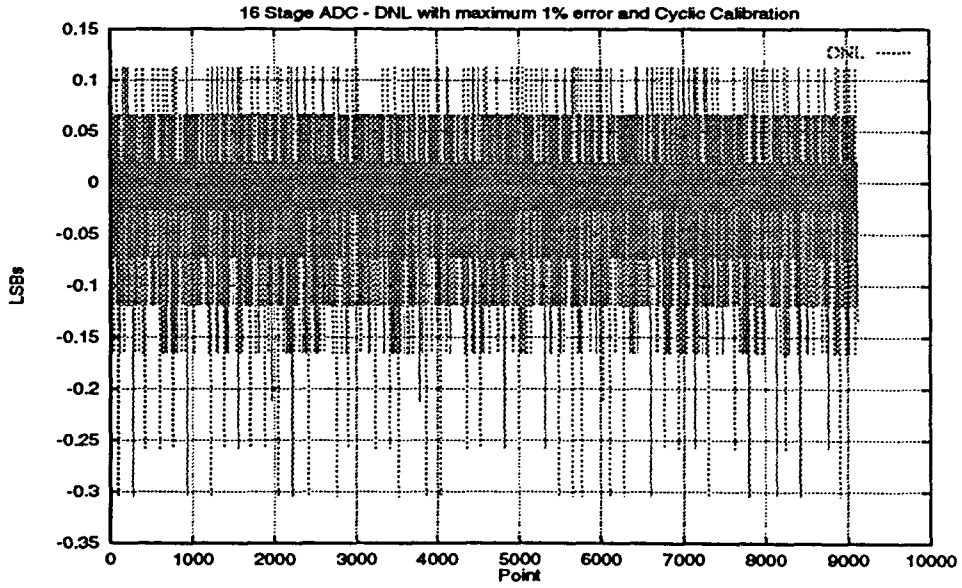

Figure 4.6 DNL of a 16 stage ADC with 1% error and ABDiSC calibrated

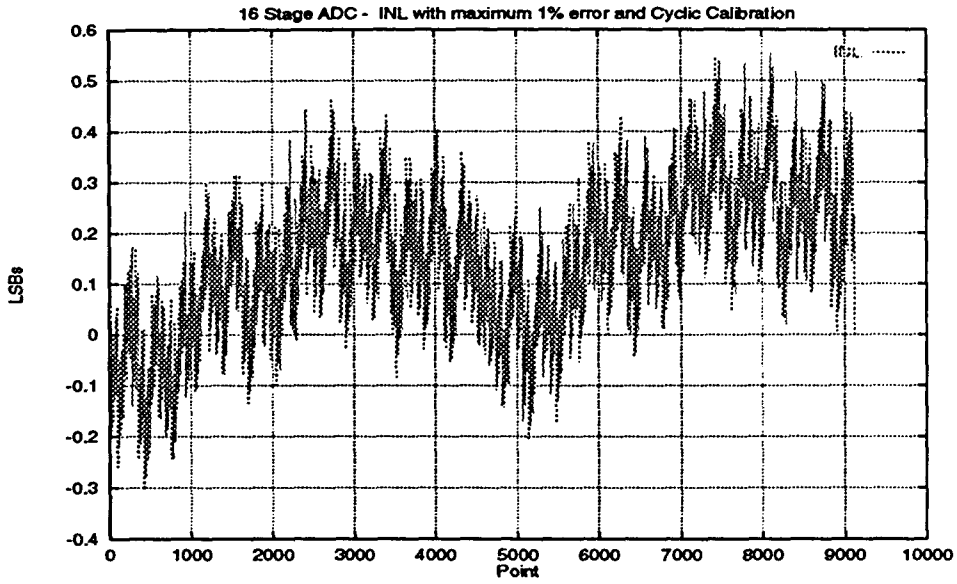

Figure 4.7 INL of a 16 stage ADC with 1% error and ABDiSC calibrated

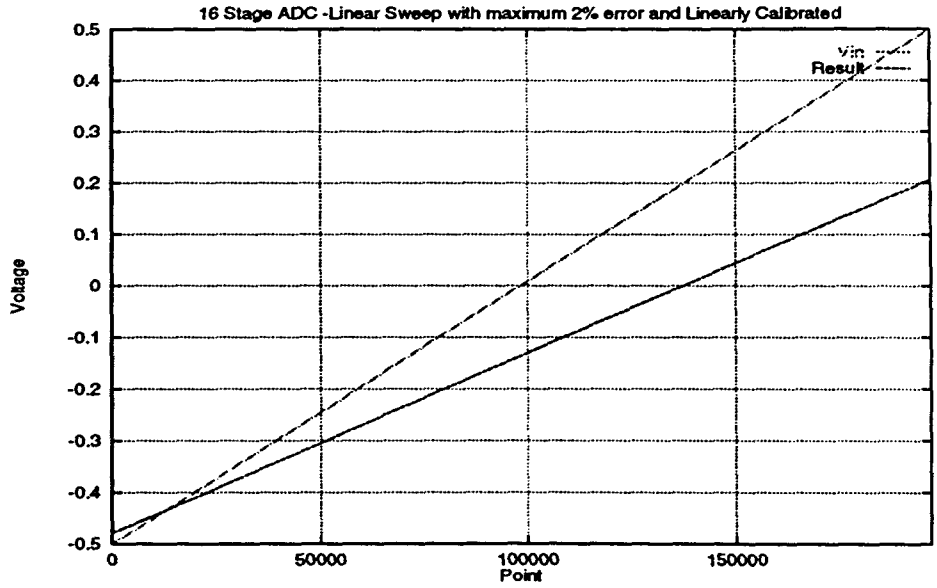

Figure 4.8 Linear sweep of a 16 stage ADC with 2% error and digitally self calibrated

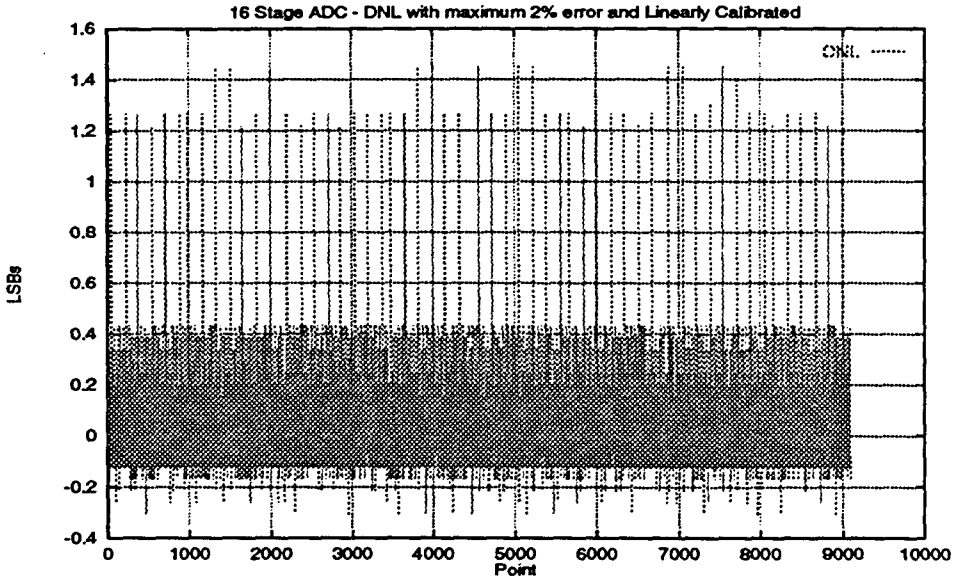

Figure 4.9 DNL of a 16 stage ADC with 2% error and digitally self-calibrated

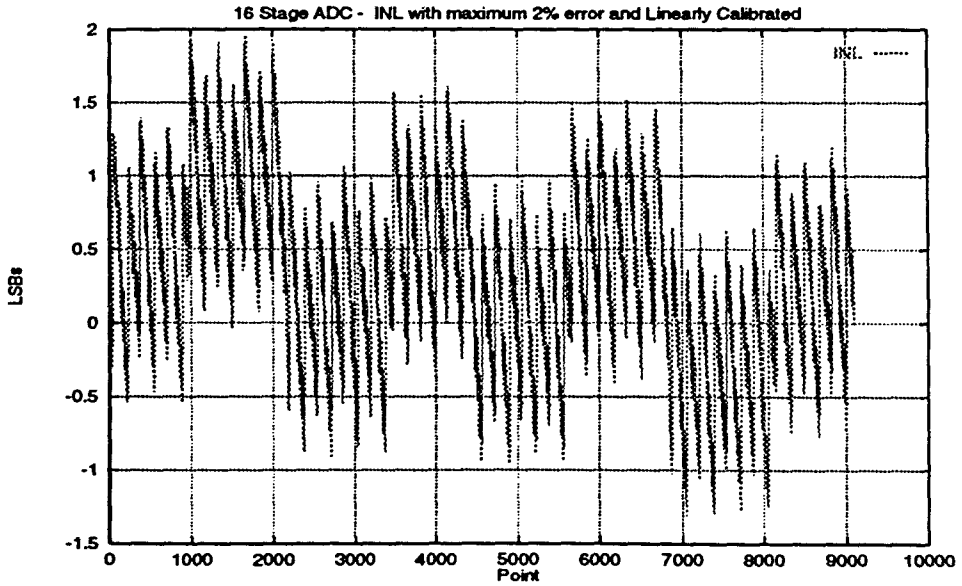

Figure 4.10 INL of a 16 stage ADC with 2% error and digitally self-calibrated

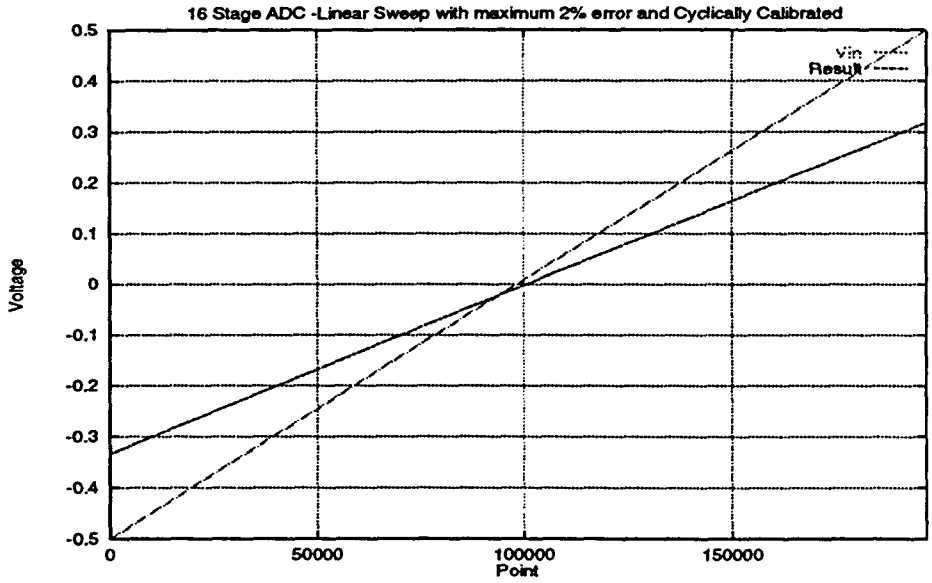

Figure 4.11 Linear sweep of a 16 stage ADC with 2% error and ABDiSC calibrated

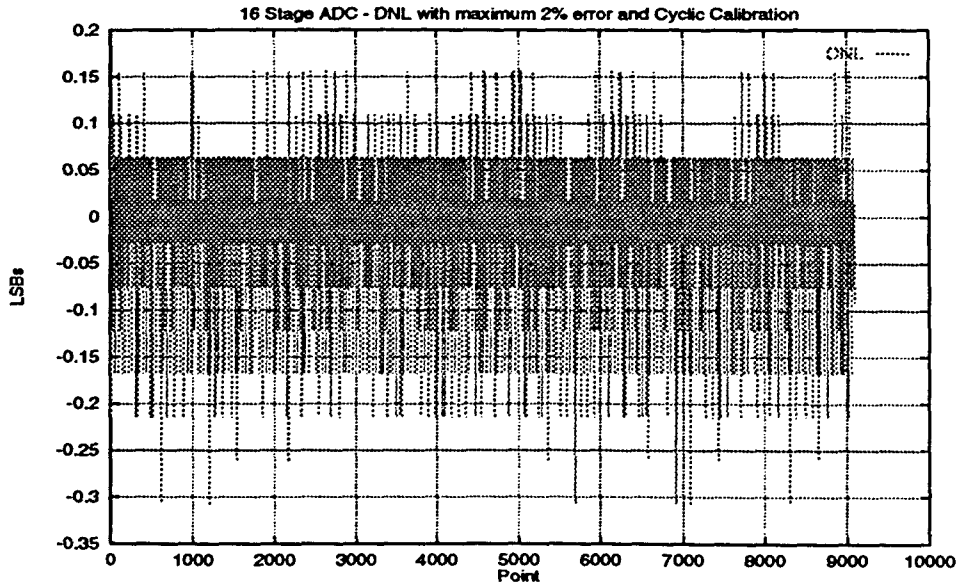

Figure 4.12 DNL of a 16 stage ADC with 2% error and ABDiSC calibrated

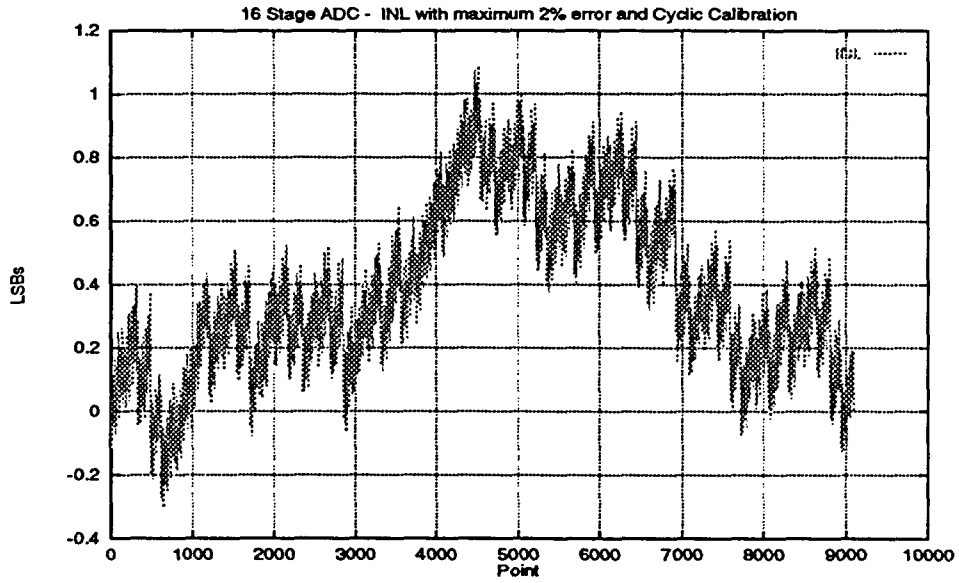

Figure 4.13 INL of a 16 stage ADC with 2% error and ABDiSC calibrated

4.2 Statistical Testing

A statistical experiment with 30 runs was also performed on a 12 stage converter with 6 stages of G = 1.93 and 6 stages of G = 2. The tests were performed with both methods of calculating the INL and DNL, the digital output reduction method (which has been explained in Chapter 3) and the end points method. For each run, an error of 1% randomly distributed on the components was generated. Then the converter was calibrated using ABDiSC and INL and DNL calculated. Then the converter was *de-calibrated* i.e. all the effects of calibration were removed so that the net result was a converter with only the original component errors. It was then calibrated using digital self-calibration, called DSC in the figures. The error generated in each run was unrelated to the error generated in the next run i.e. it was completely random and hence might be same as the previous one or not. Figure 4.14 shows the DNL plot and Figure 4.15 shows the INL plot of 30 runs obtained with the digital output reduction method. The y-axis denotes the absolute value of the maximum DNL and maximum INL, respectively, whereas the x-axis denotes the run count. Figure 4.16 shows the DNL plot and Figure 4.17 shows the INL plot calculated with end points method. In all the figures, ABDiSC calibrated converter has a smaller value of absolute INL and DNL values. The statistical results clearly show that the ABDiSC method gives better or comparable results to digital self-calibration.

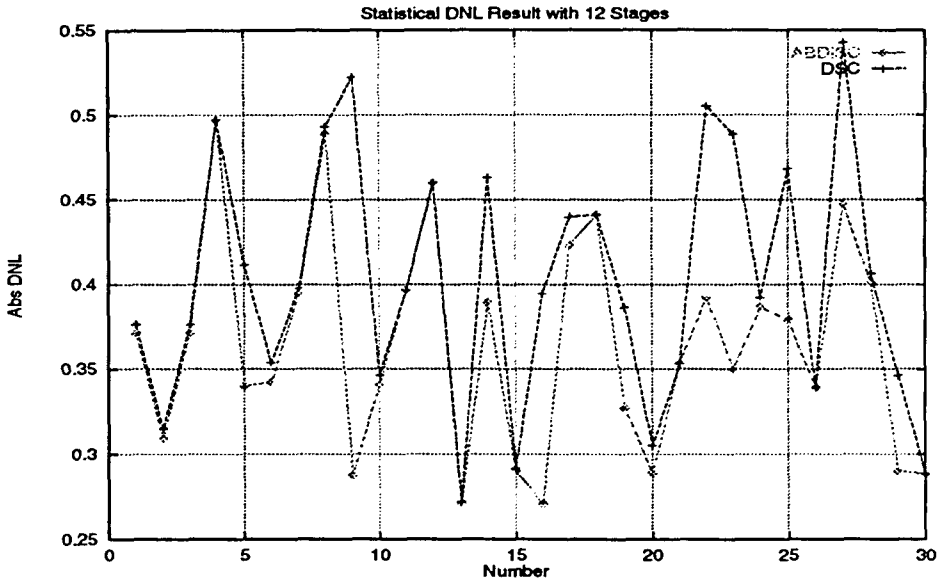

Figure 4.14 Statistical DNL plot of 30 runs with 12 stage converter and 1% component error obtained with digital output reduction method

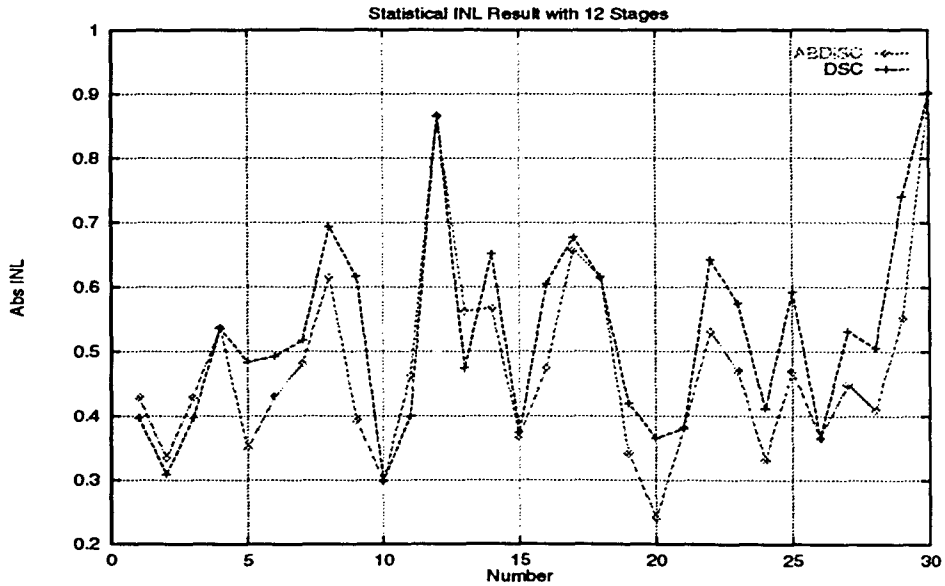

Figure 4.15 Statistical INL plot of 30 runs with 12 stage converter and 1% component error obtained with digital output reduction method

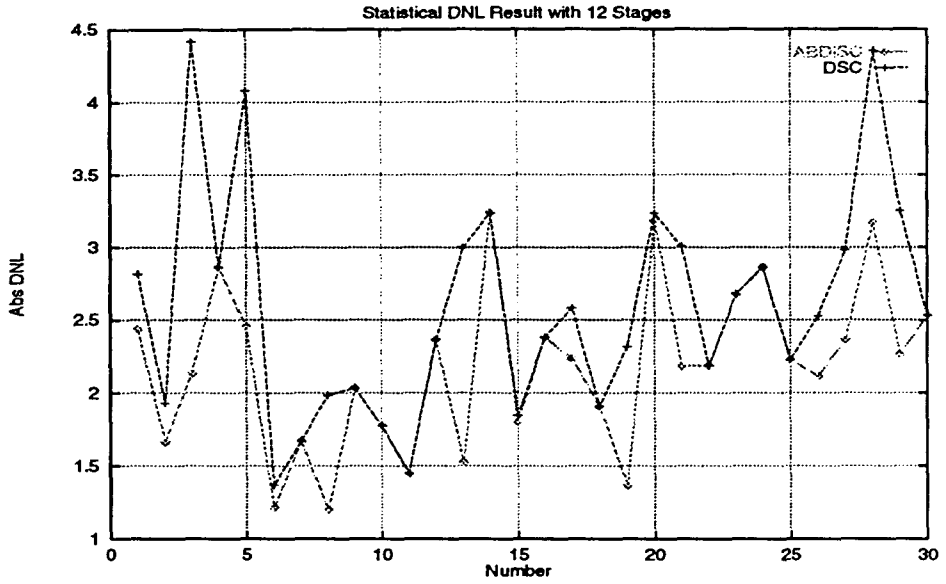

Figure 4.16 Statistical DNL plot of 30 runs with 12 stage converter and 1% component error obtained with end points method

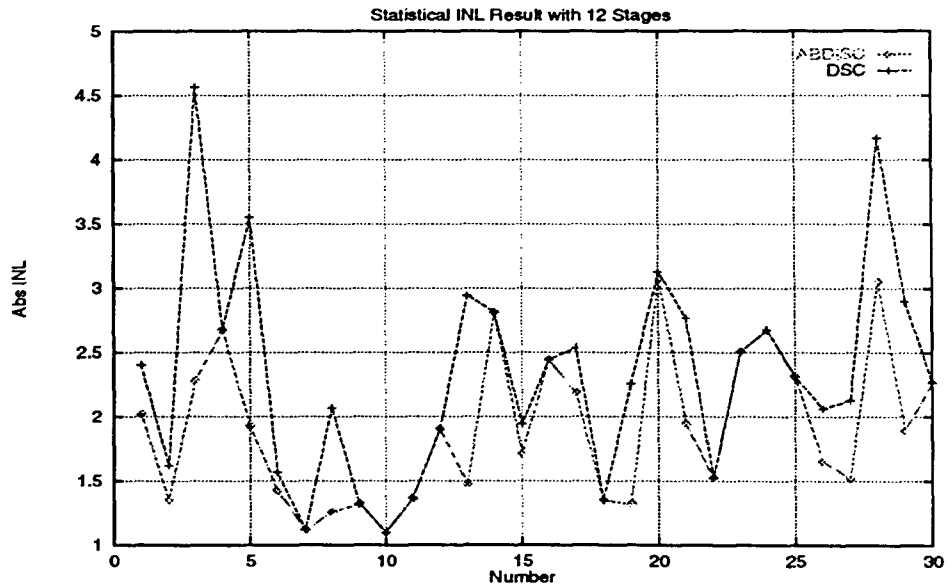

Figure 4.17 Statistical INL plot of 30 runs with 12 stage converter and 1% component error obtained with end points method

4.3 Results of the Single and Two Tone Testing

To check the spectral purity of the output of the converter calibrated by ABDiSC, single and two tone tests were performed. These results were then compared with digital self-calibration. Figure 4.18 shows the FFT of the single tone input to the converter having 0.5% component error and 0.001% opamp nonlinearity. The FFT of uncalibrated converter is shown in Figure 4.19, FFT of converter calibrated with digital self-calibration is shown in Figure 4.20 and FFT of converter calibrated with ABDiSC is shown in Figure 4.21. The SFDR for the digitally self-calibrated converter is approximately 55 dB and that for ABDiSC calibrated converter is 60 dB.

Figure 4.22 shows the FFT of the two tone input to the converter. The FFT of uncalibrated converter is shown in Figure 4.23, FFT of converter calibrated with digital self-calibration is shown in Figure 4.24 and FFT of converter calibrated with ABDiSC is shown in Figure 4.25. The SFDR for digitally self-calibrated converter is 50 dB and for ABDiSC calibrated converter 60 dB.

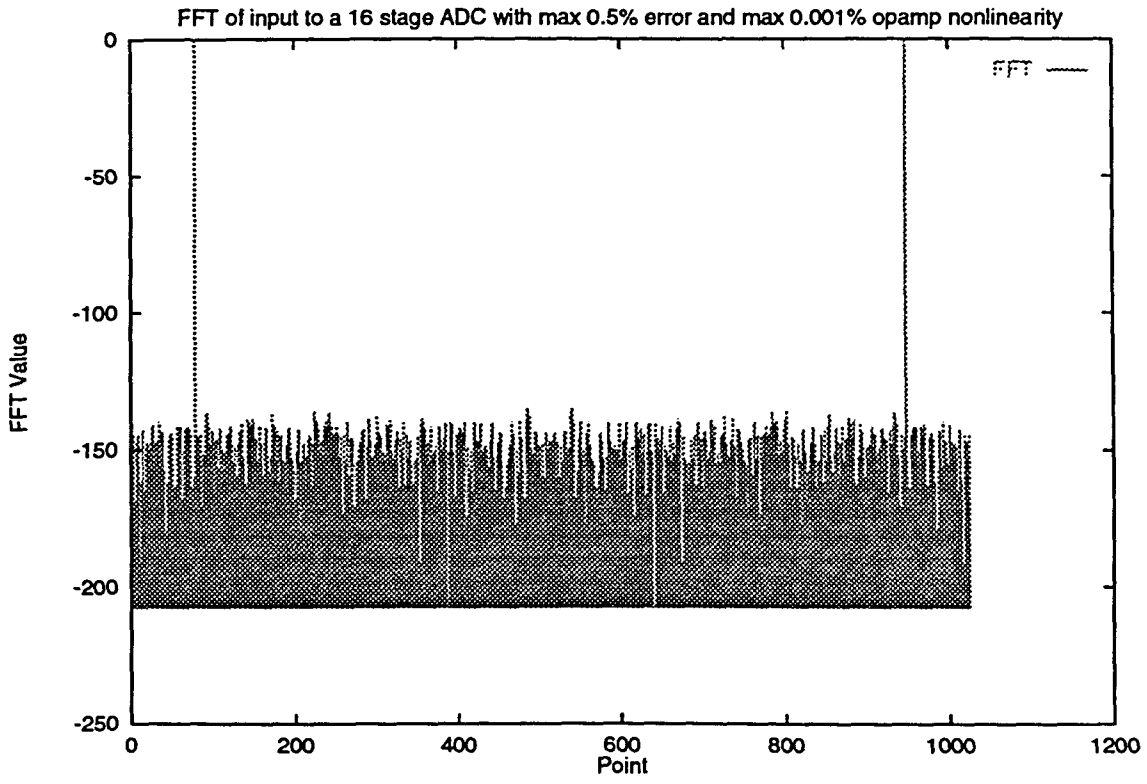

Figure 4.18 Single tone testing - FFT of input to a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity

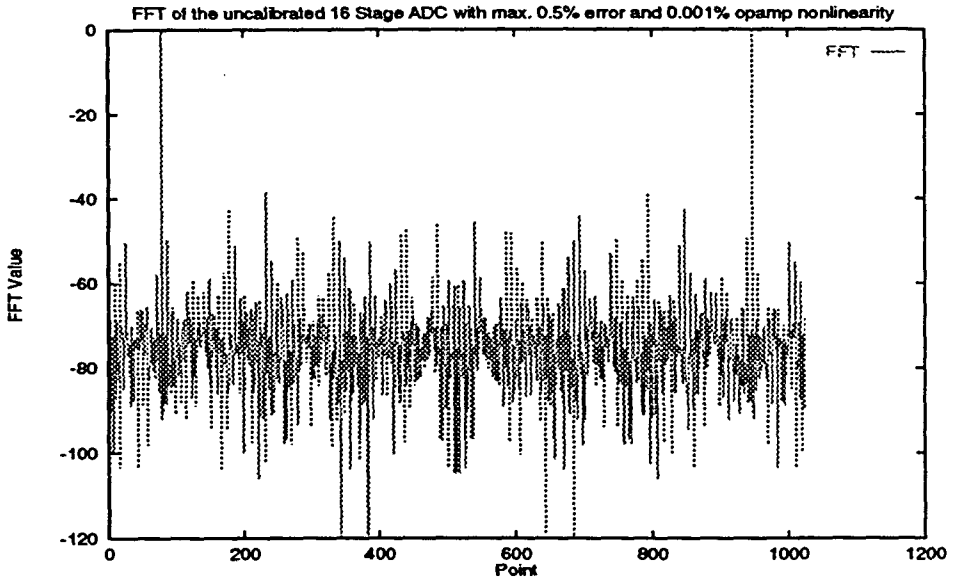

Figure 4.19 Single tone testing - FFT of an uncalibrated 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity

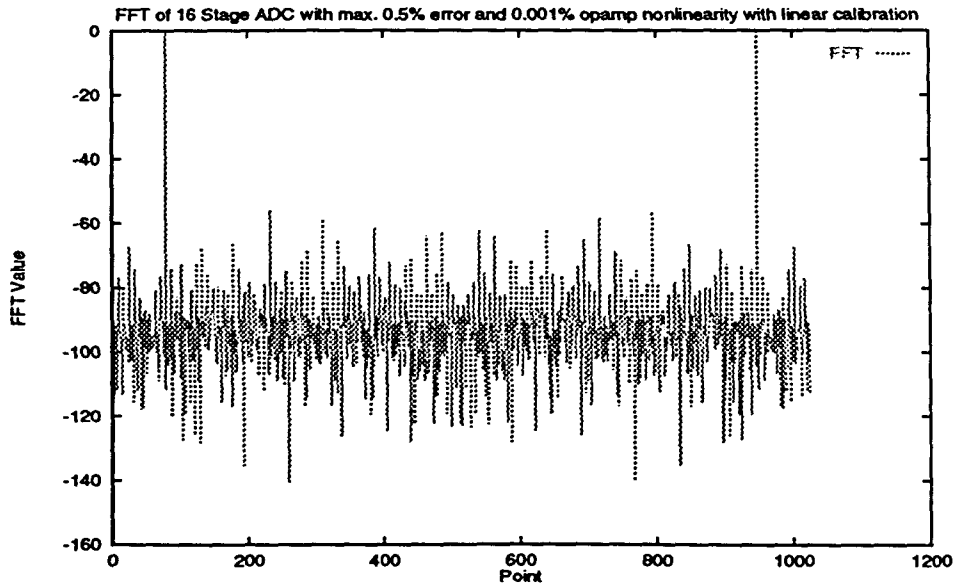

Figure 4.20 Single tone testing - FFT of a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity and digitally self-calibrated

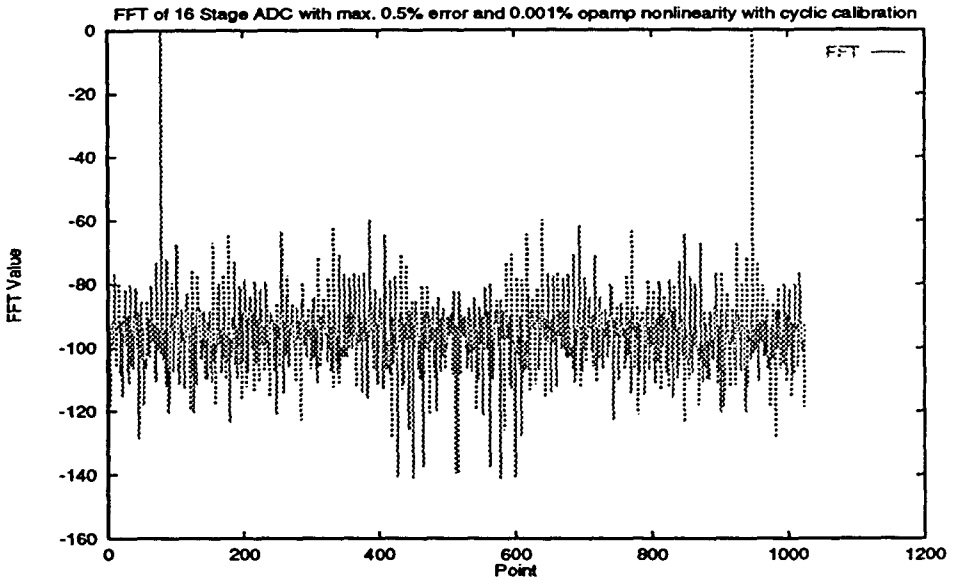

Figure 4.21 Single tone testing - FFT of a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity and ABDiSC calibrated

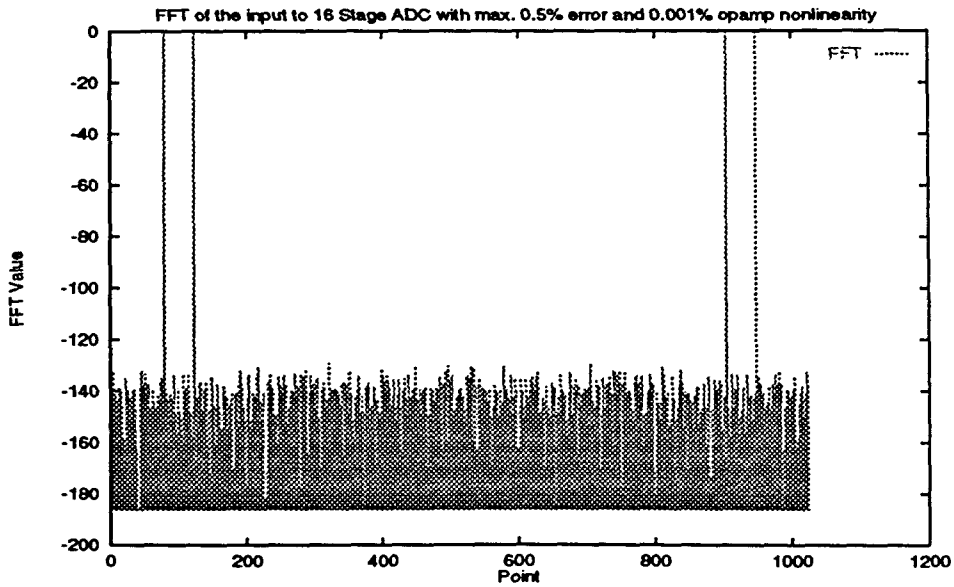

Figure 4.22 Two tone testing - FFT of input to a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity

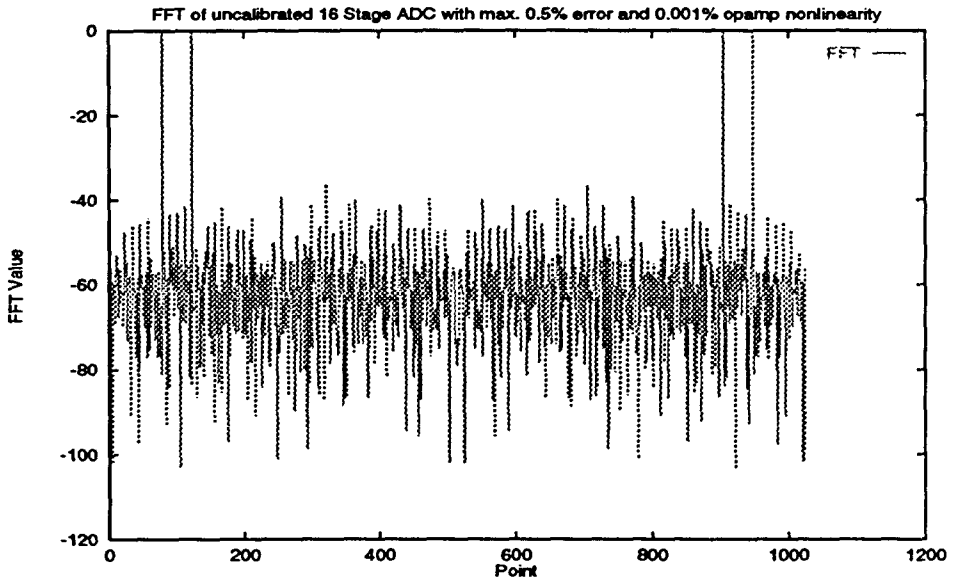

Figure 4.23 Two tone testing - FFT of an uncalibrated 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity

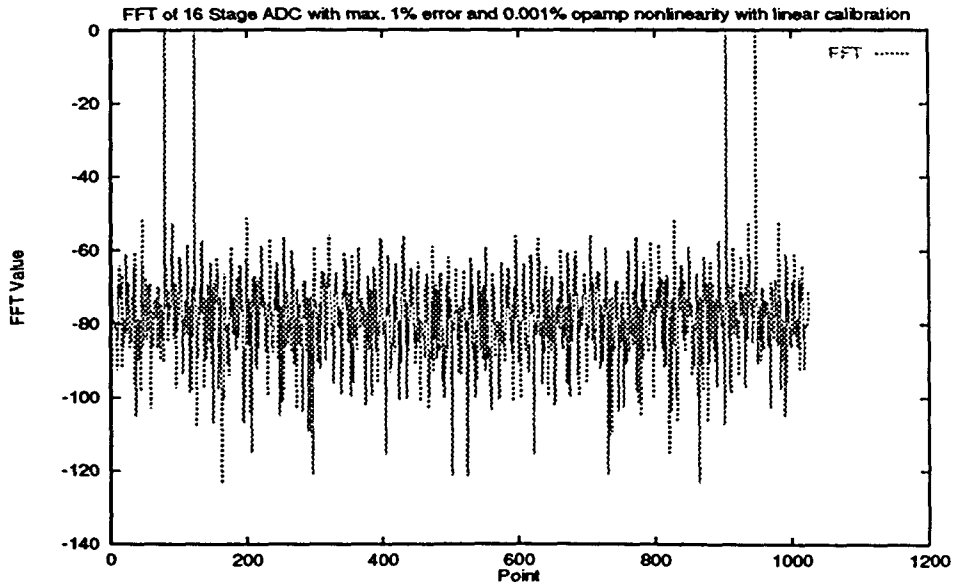

Figure 4.24 Two tone testing - FFT of a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity and digitally self-calibrated

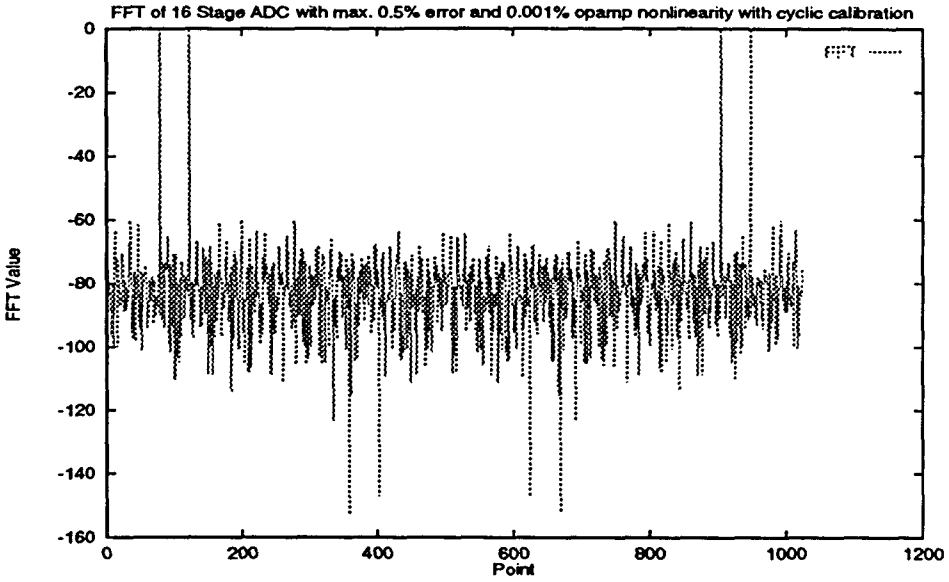

Figure 4.25 Two tone testing - FFT of a 16 stage ADC with 0.5% error and 0.001% opamp nonlinearity and ABDiSC calibrated

4.4 Conclusion

In this section results on the linearity of ADC calibrated with ABDiSC and digital self-calibration were presented and compared. It was shown that ABDiSC performed better than digital self-calibration in most cases and at worst it is comparable. This is a due to the the fact that more stages are used during calibration in ABDiSC. Their operation in frequency domain, however are comparable.

5 SIMPAD - SIMULATOR FOR MULTIPLE PIPELINE A/D CONVERTER

5.1 Introduction

As the demands for high speed/high resolution converters grow day by day, there is an increased interest in pipeline A/D converters especially after the development of calibration techniques without which the linearity of the A/D converters was limited to 8-10 bits. This has exposed a need for functional simulators which can model the pipeline converters at the functional level leaving the details of the analog domain to the designers and concentrating on development of algorithms. In the past, there have been some functional simulators for mixed signal and analog sampled data systems like MIDAS [31], but there has not been any available simulator specifically targeted towards pipelined A/D converters. SIMPAD [35] was originally developed with the intention of simulating only accuracy bootstrapped algorithm [3]. From there it has progressed to incorporating a few more algorithms and techniques for testing of ADCs. SIMPAD has the advantage that a new model or architecture can be easily incorporated into the program which can then be simulated. The amount of work required to incorporate the new model depends upon how different the architecture is from those already present in the program. Currently it supports the accuracy bootstrapping architecture, digital self-calibration algorithm, a new simplified cell architecture for accuracy bootstrapping and a pure non-binary radix architecture. All the architectures can be tested as a single pipeline or as parallel pipeline. Besides providing the standard tests for testing the linearity of an ADC another important specification for converters, especially the ones targeted for communication applications, Spurious Free Dynamic Range (SFDR), has also been incorporated. SIMPAD provides the capability to perform single tone and double tone testing for pipelines and to perform FFT's on the resulting data. The core of the accuracy bootstrapping was written by Eric Soenen while he was at Texas A & M university . SIMPAD was developed on this core. A manual for SIMPAD is attached as an appendix.

5.2 Model of a Single Stage of the Pipeline

A generalized model of a single stage of a pipeline converter is shown in Figure 5.1. By choosing one of the methods (architectures) mentioned in the introduction, the user selects the architecture which he/she wants to simulate. The gain of each stage and the architecture determines the bits per stage. Once the single stage has been initialized the program asks for number of such stages which form the pipeline. Three kinds of linear component errors are possible in each stage. These are errors in the Gain of the Sample/Hold amplifier, errors in the Flash ADC reference levels and errors in the reconstructing DAC. Depending on the user specification, the distribution of these errors is either uniform or extreme and is random in nature. The model can be simulated even without any errors.

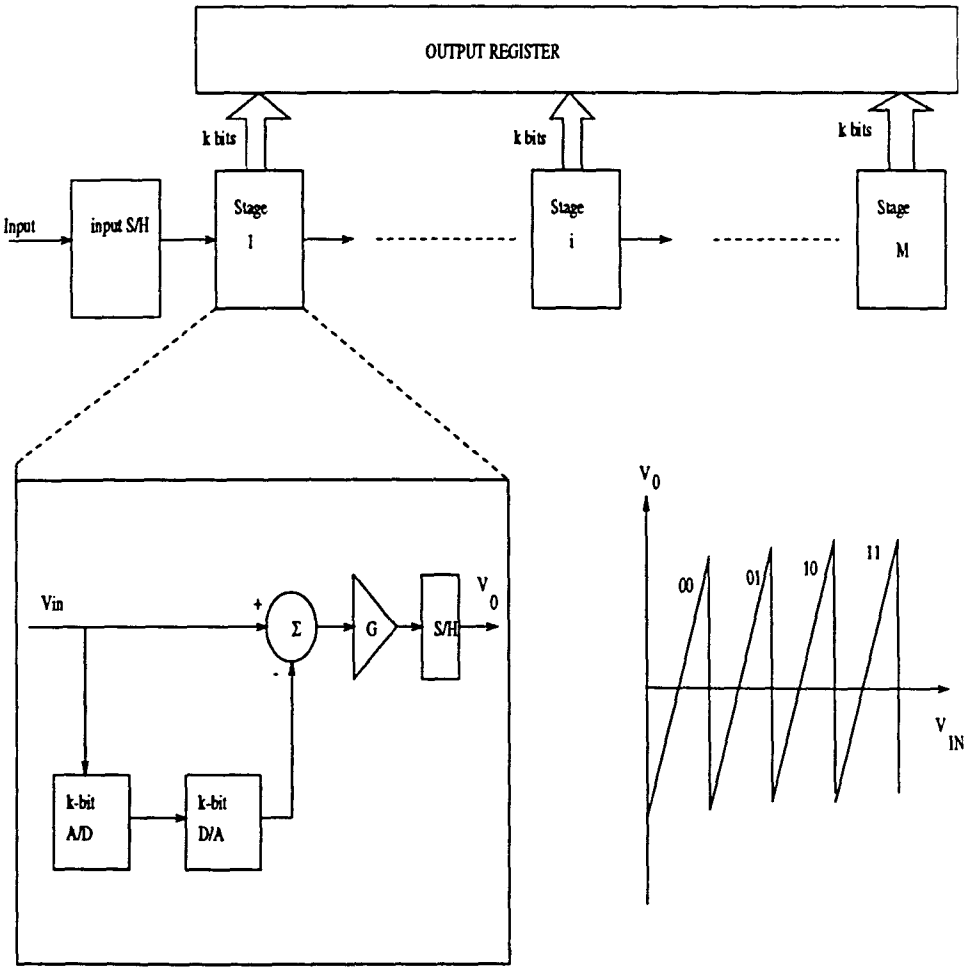

Figure 5.1 Model of the single stage

The gain (A), the DAC level and the ADC levels of each stage l are modeled by the following equations:

$$A[l] = A + \delta A_l \tag{5.1}$$

$$DAC[i][l] = DAC[i][l] + \delta DAC[i][l] \quad where \ \ i \ \ is \ \ the \ \ ith \ \ DAC \ \ level \tag{5.2}$$

$$ADC[i][l] = ADC[i][l] + \delta ADC[i][l] \quad where \ \ i \ \ is \ \ the \ \ ith \ \ ADC \ \ level \tag{5.3}$$

The errors in above equations are random in nature. Output referred noise and nonlinearity are also added at each stage. Nonlinearity is discussed in the next subsection. This is just added to the residue of each stage as

$$V_{res} = V_{res} + noise + nonlinearity \tag{5.4}$$

The digital output is obtained by adding the weight of the stage divided by the gain of the stage with the incoming residue to generate the output residue. The output of the flash A/D gives the bits of the present stage.

5.2.1 Opamp Model

Since the sample-and-hold is incorporated along with the opamp, the model of the opamp also allows for the simulation of maximum equivalent output nonlinearity (le) at each stage. The nonlinearity can be either modeled as single bow nonlinearity as shown in Figure 5.2 or as a double bow nonlinearity as shown in Figure 5.3. The nonlinearity for a single bow is modeled as

$$Single \ Bow \ Nonlinearity = 4 * le * V_{res} * (1 - V_{res}) \tag{5.5}$$

and for a double bow as

$$Double \ Bow \ Nonlinearity = 64/3 * le * V_{res} * (1 - V_{res}) * (V_{res} - 0.5) \tag{5.6}$$

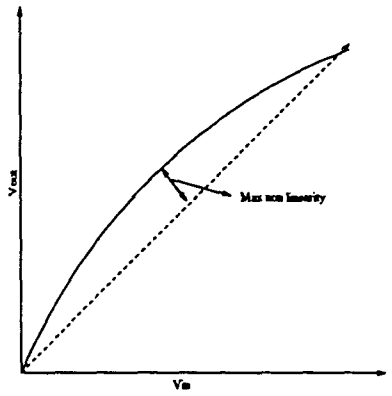

Figure 5.2 Single bow nonlinearity

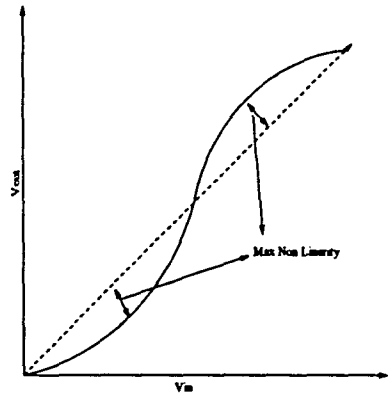

Figure 5.3 Double bow nonlinearity

5.3 Program Description

5.3.1 Input and Output Format

The parameters for simulating an algorithm and architecture can be either given at the command line or through an input file. The parameters initialized are the number of stages, the number of redundant stages, the gain per stage and the calibration algorithm used. Then, the gain error, the A/D reference errors i.e the errors in the flash converter, the DAC errors, the nonlinearity and the output referred noise are initialized as percentages of the full scale. The A/D converter can be simulated without these errors also. The errors are incorporated in a random fashion. The output of the program is the DNL and INL of the linear sweep performed on the ADC. The output is displayed on the screen by using Gnuplot. Sinusoidal sweep on the ADC can also be performed and its FFT analyzed for spectral purity.

5.3.2 Linearity Testing of the ADCs

The various routines used within SIMPAD for testing the linearity of the ADCs are:

1. Ramp or Sinusoidal Input: The A/D can be simulated with both sinusoidal or ramp inputs. The ramp sweeps over the entire range in user specified steps. The sinusoidal sweep is a single sine wave for single tone testing and a sum of two sine waves with frequencies in the ratio of an irrational number for two tone testing.

2. INL and DNL calculation [28]: Two of the most important ADC specifications are the INL and DNL specifications. INL (Integral Non Linearity) is the maximum deviation of the actual transition points in an ADC's transfer characteristic from the straight line drawn between the end points (first and last code transitions). DNL (Differential Non Linearity) is defined as the difference between the ideal bin width of the converter and the actual width of each bin. A bin is defined as the set of input voltages yielding the same output code.

3. Fast Fourier Transform (FFT) [29]: The Fast Fourier Transform can be performed on the output of the files created by the simulator. Plotting capabilities for the FFT are also provided.

4. Single and Two Tone Testing: In single tone testing a sinusoidal signal having a single sine component is applied to the ADC. Upto 95% of the full scale is used for simulation. The FFT of the single tone is a single pulse at N*(frequency of sine wave)/2π. In two tone testing, two sinusoids, whose periods are not integral multiples of each other are applied to the ADC. The FFT of the two tone testing output would be a pulse each at the applied frequencies.

5.4 Conclusions

In this chapter an overview of the SIMPAD was provided. The user interface to the program and the various routines for testing the ADC's linearity were also explained. The model of the single stage was also explained in detail. All the results presented in this thesis were obtained using SIMPAD. More work is in progress to extend the capabilities of the program and the models used.

6 CONCLUSION

6.1 Summary

This thesis discussed the two popular algorithms, namely accuracy bootstrapping and digital self-calibration, used in pipelined ADCs for digital calibration. A new simplified cell architecture for accuracy bootstrapping having less hardware was presented and experiments performed to compare it's performance to the earlier architecture. It was shown that the new simplified architecture performed equally well. A new algorithm for calibration of non-radix-2 converters called ABDiSC was also proposed and shown to perform better than digital self-calibration. The algorithm combines the ideas in accuracy bootstrapping and digital self-calibration to arrive at a better algorithm.

6.2 Contributions

The three main contributions of this work are:

1) A new architecture for accuracy bootstrapping algorithm was studied and results were presented. An insight into the problem of grounding one of the capacitors for obtaining the zero analog voltage was discussed and a method to correct the error was suggested.

2) An improved algorithm for the digital calibration of non-radix-2 pipeline called ABDiSC was proposed and was shown to be better than digital self-calibration. The ABDiSC algorithm uses the concept of accuracy bootstrapping to make the calibration process more accurate i.e. the digital self-calibration is combined with accuracy bootstrapping to obtain a better result with no extra hardware.

3) The simulation package was developed further to handle more models like the new architecture for accuracy bootstrapping and the non-radix-2 architecture.

6.3 Future Work

One of the possible directions for future work could be to incorporate analog calibration methods in the simulation package. There is a limit to what and how much digital calibration can achieve. So

some work must be done to explore the limits of analog calibration. Digital calibration is also being used in oversampling ADCs which can also be incorporated into the software.

APPENDIX THE MANUAL FOR SIMPAD

This manual was prepared as a reference material for the users of SIMPAD. It is not meant as a tutorial to understand the accuracy bootstrapping or digital self-calibration algorithms. The user is assumed to have a general familiarity with the pipelined A/D converters.

INTRODUCTION

The program is run at the shell prompt by typing the line

user name> accu

The program then asks you if you want the calibration to be done by the accuracy bootstrapping algorithm or by the digital self-calibration algorithm. See Figure A.1 for the flow chart.

Accuracy Bootstrapping (1) or Digital Self-Calibration (2) or TI Architecture (3) or Bipolar (4) ?

Choosing 1 results in the Default values being shown. Hitting the <RETURN> key at this stage results in the following menu being presented.

M: choose simulation Method (accu or dscp or TI or bipolar)

D: Define and initialize ADC

I: Input ADC configuration from a file (and pre-calculated

random errors if included in the file)

E: set Error limits

R: eRror cancellation algorithm

P: Perturb configuration (add random errors)

A: display ADC configuration

W: display Weights (digital coefficients)

N: calculate ADC Non-linearity

L: Linear sweep - INL / DNL

O: sinusOidal sweep

C: Calibrate configuration

B: run Batch of different configurations (Monte-Carlo)

T: scale weights using Two-point measurement

U: sUmmary

F: DAVID'S linearity test (may take a while)

G: DAVID'S lin sweep lsb/3 steps

H: DAVID'S random error calibration test

S: Save ADC configuration and errors to a file

Z: Define Multiple (Parallel) Paths

V: Perform the FFT

K: Calibrate the multiple pipes

Q: Quit

Description of the various options

The various options are described below. Note that the program is insensitive to the case of the letter typed in.

M: Choose simulation method (accu or dcsp or simplified cell or bipolar)

Choosing this option returns you to the stage where you are asked whether you would like accuracy bootstrapping to be used or digital self-calibration or simplified cell architecture or bipolar architecture. Once the choice is made, hitting the <RETURN> key results in the original menu returning.

D: Define and initialize ADC

This option is to initialize the initial configuration of the ADC. The various parameters to be initialized are:

No of stages

The first parameter to be initialized is the number of stages in the ADC. A stage is a sample and hold amplifier with a ADC, DAC, a Summer and a gain stage.

Delta

This corresponds to the extra stages in the pipeline. These extra stages are for redundancy in order to achieve accuracy. The resolution of the ADC is only up to the number of bits declared as the number

of stages. The delta is just for redundancy and error correction.

Nominal Interstage gain (A)

This corresponds to the gain of the amplifier which results in the difference being brought out as the residue. Note that the number of bits per stage is $|A| - 1$. Usually it is kept at 2.

At this stage, the initialization is completed and the nominal configuration along with the initial weights are calculated. Note though that the ADC is initialized with no errors. To incorporate errors, the P option is later used.

I: Input ADC configuration from a file (and pre-calculated random errors if included in the file)

As the name indicates, the initial ADC configuration along with the pre-calculated errors can be input as a file. This is valid even for multiple pipes.

E: set Error limits

As the title indicates, choosing this option allows us to set error limits on the various parameters responsible for ADC nonlinearity. These are as follows:

Maximum Interstage Gain error

This is the error in the interstage gain A. It is entered in percentage i.e if i enter 1 it means 1% gain error. Note that the default values are shown in the brackets and hitting the <RETURN> key without entering anything results in the default value being used.

Maximum ADC (flash) error

This is the error in the Flash ADC which converts the analog sampled input signal to its digital equivalent at that stage. This is also entered as a percentage.

Maximum DAC error

This is the error in the DAC which converts the Flash ADC output back to its analog equivalent in order to be subtracted from the original signal to generate the residue. This is the error which is mainly removed by the self-calibration algorithm. Note that this is also input as a percentage.

Error Distribution

This corresponds to the error distribution throughout the stages. Choosing u results in the error through all the stages being uniform. That is any error is generated by a random process. Choosing the e option on the other hand results in the errors being extreme in nature i.e. each error is equal in magnitude to either + or - the maximum as set by the error limit.

Simulate recycling ADC, stages identical (y/n)

The pipelined converter can also be simulated as a cyclic converter with the output of each stage being fed back to itself. It is the same as a non-cyclic converter except that the stages are separated in time rather than spatially.

Max. equivalent output noise of stage

This corresponds to the maximum output noise of the stage.

Number of averages during calibration

In order to reduce the effect of noise, a number of calibrations can be done successively. This option specifies the number of averages to be done. Usually for the accuracy bootstrapping algorithm it is found that 1 calibration is good enough and further calibrations do not result in any significant improvement.

Max. equiv. output non-linearity of stage

This is self explanatory.

Simulate identical non-linearity in each stage (y/n)

This option queries you on whether to simulate each stage with a similar nonlinearity or whether to simulate each stage with a different nonlinearity. The default is usually set at a similar nonlinearity.

P: Perturb configuration (add random errors)

Choosing this option results in the nominal configuration being "perturbed" from its initial condition. Random errors are generated for various ADC parameters and incorporated. The weights of the various stages are then computed.

A: display ADC configuration

This option allows us to view the various ADC parameters at this stage. It first asks you whether to dump the output into a file or onto the screen. The ADC levels of each stage, the incremental DAC levels of each stage, the nominal gain of each stage and the nonlinearity are all output.

W: display Weights (digital coefficients)

This option allows us to view the ideal and actual weights of each stage. As above the output can be output to a file or to the screen. The ideal weights (DAC levels) and the actual Weights (say after calibration) of each stage are output.

N: calculate ADC Nonlinearity

This option calculates the ADC nonlinearity using certain theoretical measures described in [2]. Refer to the paper for more details.

L: Linear Sweep -INL/DNL

This option inputs a ramp input to the ADC and sweeps across the entire range. Based on two point measurements, it calculates the ADC linearity in terms of the INL and DNL errors. Plotting options are also available.

O: sinusOidal sweep

This option inputs a sinusoidal signal to the ADC and measures the output. Plotting options are also available.

C: Calibrate configuration

Choosing this option results in the accuracy bootstrapping algorithm being implemented for the ADC under consideration. Once this is done, the ADC configuration, Weights, a linear or sinusoidal sweep can be done and the effect of the algorithm on the accuracy and nonlinearity of the ADC can be examined.

B: run Batch of different configurations (Monte-Carlo)

This option allows us to run a batch of different configurations allowing us to perform a statistical analysis.

F: DAVID'S linearity test (may take a while)

This option allows us to run a number of configurations allowing us to perform a statistical analysis.

S: Save ADC configuration and errors to a file

As the name indicates, choosing this option allows us to save the present ADC configuration and errors into a file for later use.

Z: Define Multiple (Parallel) Paths

This option allows us to define multiple paths of the ADC. The number of paths is first input. The paths are then perturbed depending on the users choice individually. Then the calibration of paths individually (accuracy bootstrapping) is done again depending on the user. Overall global normalization is done later by choosing K from the main menu.

V: Perform the FFT

This option allows us to perform an FFT. The input files to the FFT have to be first saved and named. It also allows plotting capabilities.

K: Calibrate the multiple pipes

This option allows us to globally normalize the ADC's multiple parallel pipes. It is done by two point measurements. Note that it has to be done after choosing the Z option and calibrating each pipe individually by accuracy bootstrapping.

Q: Quit

*** EXITING accu - GOOD BYE ***

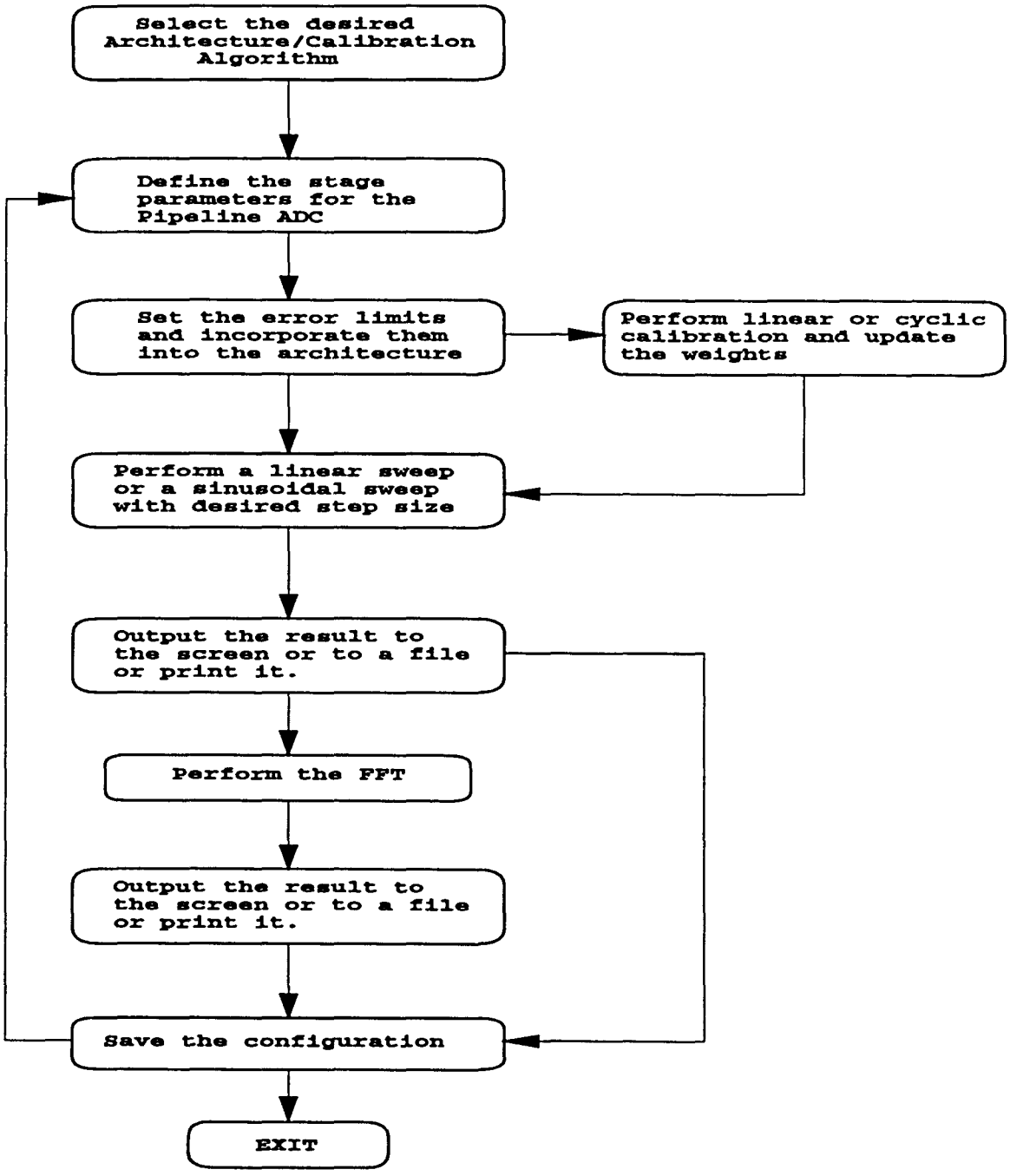

Figure A.1 Flow chart for SIMPAD

BIBLIOGRAPHY

[1] Andrew N. Karanicolas, Hae-Seung Lee and Kantilal L. Bacrania, "A 15-b 1-MSample/s Digitally Self-Calibrated Pipeline ADC," *IEEE Journal of Solid-State Ciruits*, vol. 28, No. 12, pp.1207-1215, Dec. 1993.

[2] Eric G. Soenen and Randall L. Geiger, "An Architecture and An Algorithm for Fully Digital Correction of Monolithic Pipelined ADC's," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 42, No. 3, pp. 143-153, Mar. 1995.

[3] Eric G. Soenen, "Error Modeling, Self Calibration and Design of Pipelined Analog to Digital Converters," Texas A & M University, College Station, Texas, 1992.

[4] Stephen H. Lewis, H. Scott Fetterman, George F. Gross,Jr., R.Ramchandran and T. R. Viswanathan, "A 10-b 20-MSample/s Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. 27, No. 3, pp. 351-357, Mar. 1992.

[5] Hae-Seung Lee, "A 12-b 600 ks/s Digitally Self Calibrated Pipelined Algorithmic ADC," *IEEE Journal of Solid-State Ciruits*, vol. 29, No. 4, pp. 509-515, Apr. 1993.

[6] Seung-Hoon Lee and Bang-Sup Song, "Digital-Domain Calibration of Multistep Analog-to-Digital Converers," *IEEE Journal of Solid-State Circuits*, vol. 27, No. 12, pp. 1679-1688, Dec. 1992.

[7] K. Nakamura, M. Hotta, L. R. Carley and David J. Allstot, "An 85mW, 10b, 40 MSample/s CMOS Parallel-Pipelined ADC," *IEEE Journal of Solid-State Ciruits*, vol. 30, No. 3, pp. 173-183, Mar. 95.

[8] Behzad Razavi and Bruce A. Wooley, "A 12-b 5-MSampl/s Two-Step CMOS A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 27, No. 12, pp. 1667-1677, Dec. 1992.

[9] Ping Wai Li, Michael J. Chin, Paul R. Gray and R. Castello, "A Ratio-Independent Algorithmic Analog-to-Digital Conversion Technique," *IEEE Journal of Solid-State Circuits*, vol. SC-19, No. 6, pp. 828-836, Dec. 1984.

[10] Hae-Seung Lee, David A. Hodges and Paul R. Gray, "A Self-Calibrating 15 Bit CMOS A/D Converter," *IEEE Journal Solid-State Circuits*, vol. SC-19, No. 6, pp. 813-819, Dec. 1984.

[11] Yuh-Min Lin, B. Kim and Paul R. Gray, "A 13-b 2.5-MHz Self-Calibrating Pipelined A/D Converter in 3-μ CMOS," *IEEE Journal Solid-State Circuits*, vol. 26, No. 4, pp. 628-635, Apr. 1991.

[12] H. Ohara, H. X. Ngo, M. J. Armstrong, C. F. Rahim and Paul R. Gray, "A CMOS Programmable Self-Calibrating 13-bit Eight-Channel Data Acquisition Peripheral," *IEEE Journal Solid-State Circuits*, vol. SC-22, No. 6, pp. 930-937, Dec. 1987.

[13] Seung-Hoon Lee and Bang-Sup Song , "A Direct Code Error Calibration Technique for Two-Step Flash A/D Converters," *IEEE Transactions on Circuits and Systems*, vol. 36, No. 6, pp. 919-922, Jun. 1989.

[14] Seung-Hoon Lee and Bang-Sup Song, "Interstage Gain Proration Technique for Digital-Domain Multi-Step ADC Calibration," *IEEE Transactions on Circuits and Systems - II*, vol. 41, No. 1, pp. 12-18, Jan. 1987.

[15] Tzi-Hsiung Shu, Bang-Sup Song and Kantilal Bacrania, "A 13-b 10-Msamples/s ADC Digitally Calibrated with Oversampling Delta-Sigma Converter," *IEEE Journal Solid-State Circuits*, vol. 30, No. 4, pp. 443-451, Apr. 1995.

[16] Stephen H. Lewis, "Optimizing the Stage Resolution in Pipelined, Multistage, Analog-to-Digital Converters for Video-Rate Applications," *IEEE Transactions on Circuits and Systems-II*, vol. 39, No. 8, pp. 516-523, Aug. 1992.

[17] Thomas B. Cho and Paul R. Gray, "A 10b, 20 Msample/s, 35mW Pipeline A/D Converter," *IEEE Journal Solid-State Circuits*, vol. 30, No. 3, pp. 166-172, Mar. 1995.

[18] M. Yotsuyanagi, T. Etoh and K. Hirata, "A 10-b 50-MHz Pipelined CMOS A/D Converter with S/H," *IEEE Journal Solid-State Circuits*, vol. 28, No. 3, pp. 292-300, Mar. 1993.

[19] Rudy Van De Plassche, *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[20] Christopher W. Mangelsdorf, "A 400 MHz Input Flash Converter with Error Correction," *Proceedings of the IEEE*, vol. 66, No. 1, pp. 51-83, Jan. 1978.

[21] Rudy Van De Plassche and Peter Baltus, "An 8b, 100 MHz A/D Converter with 500 MHz Resolution Bandwidth," *IEEE International Solid-State Circuit Conference, Digest of Technical Papres*, pp. 222-223, Feb. 1988.

[22] Behzad Razavi, *Principles of Data Conversion System Design*, IEEE Press, New York, 1995.

[23] Hae-Seung Lee, David A. Hodges and Paul R. Gray, "Self Calibrating ADC's: Accuracy Considerations," *IEEE Transactions on Circuits and Systems*, vol. CAS-32, pp. 590-597, Jun. 1985.

[24] Kh. Hadidi, Vincent S. Tso and Gabor C. Temes, "An 8b, 1.3 MHz Successive Approximation A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. 25, No. 3, pp. 880-885, June 1990.

[25] Venkata K. Navin, "An Analysis of a Digitally Self Calibrated Parallel Pipelined Analog-to-Digital Converter," Iowa State University, Master's Thesis, July 1996.

[26] D. R. Martin and D. J. Secor, "High Speed Analog-To-Digital Converters in Communication Systems: Terminology, Architecture, Theory and Performance," *TRW Technology Report*, Nov. 1981.

[27] B. S. Song, M. F. Tompsett and K. R. Lakshmikumar, "A 12-bit 1-Msample/s Capacitor-Averaging Pipelined A/D Converter," *IEEE Journal of Solid-State Circuits*, vol. SC-23, pp. 1324-1333, Dec. 1988.

[28] Michael J. Demler, *High Speed Analog-to-Digital Conversion*, Academic Press Inc., Harcourt Brace Jovanovich Publishers, San Diego, California, 1991.

[29] Alan V. Oppenheim and Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall of India, New Delhi, 1994.

[30] G. M. Yin, F. O. Eynde and W. Sansen, "A High-Speed CMOS Comparator with 8-b Resolution," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 208-211, Feb. 1992.

[31] Louis A. Williams, B ernhard E. Boser, Edward W. Y. Liu and Bruce A. Wooley, *MIDAS-UCB User Manual*, Version 2.1-UCB, Stanford University and UC Berkeley, California, June 1994.

[32] Cormac S. G. Conroy, "A High-Speed Parallel Pipeline A/D Converter Technique in CMOS," University of California, Berkeley, Ph.D Thesis, 1994.

[33] Andrew Karanicolas, "Digital Self-Calibration Techniques for High-Accuracy, High-Speed Analog-to-Digital Converters," MIT, Ph.D Thesis, Mar 1993.

[34] Walt Kester, *High Speed Design Seminar*, Analog Devices, Greensboro, North Carolina, 1990.

[35] V. Navin, T. Ray, M. Hassoun, W. Black, E. Lee, E. Soenen and R. Geiger, "A Simulation Environment for Pipelined Analog-to-Digital Converters," *IEEE ISCAS*, Vol. 3, pp. 1620-1623, Jun 1997.