Calibration of multi-bit per stage pipelined ADC using statistical properties of capacitor arrays

Ray, Sourja

2008

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Calibration of Multi-Bit per Stage Pipelined ADC Using Statistical Properties of Capacitor Arrays

A dissertation submitted in partial satisfaction of the

requirements for the degree Doctor of Philosophy

in

Electrical and Computer Engineering

(Electronic Circuits and Systems)

by

Sourja Ray

Committee in charge:

Professor Bang-Sup Song, Chair
Professor Peter Asbeck
Professor Gert Cauwenberghs
Professor Chung-Kuan Cheng
Professor Bhaskar Rao

2008

The dissertation of Sourja Ray is approved, and is acceptable in quality and form for publication on microfilm:

Chair

University of California, San Diego

2008

DEDICATION

I dedicate this dissertation to Saraswati, the
Hindu Goddess of knowledge and learning.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge and thank my thesis advisor Prof. Bang-Sup Song for his extreme patience and understanding towards my work. There have been several moments during the course of my stay in his lab when I faltered and I surely would not have reached this stage without his continued support and encouragement. I have learnt a lot from him not only in the field of circuit design and calibration techniques but also about the attitude required to succeed in my life and career. I would also like to thank the members of my Ph.D. committee Prof. Peter Asbeck, Prof. Gert Cauwenberghs, Prof. Chung-Kuan Cheng and Prof. Bhaskar Rao for their valuable time.

Sections of chapter 3 and 4 are published in Sourja Ray, Bang-Sup Song, "A 13-b Linear, 40-MS/s Pipelined ADC with Self-Configured Capacitor Matching," *IEEE J. Solid-State Circuits*, vol.42, no.3, pp.463-474, March 2007[1] and S. Ray, Bang-Sup Song, "A 13b linear 40MS/s pipelined ADC with self-configured capacitor matching," *ISSCC Dig. Tech. Papers*, pp. 852-861, Feb. 2006[2]. Relevant sections have been reprinted with permission of the IEEE.[3]

I am forever indebted to my parents Dr. Biswajit Ray and Gopa Ray who encouraged me to follow my dreams. I must acknowledge my sister Bipasha Ray who constantly amazes me by her ability to achieve her goals. I am extremely fortunate to have my supportive wife, Trupti Dhavle, by my side at all times. It is through her unconditional love and understanding that I am able to complete this dissertation. She was a super-mom during the past year and took care of our home and our daughter in the evenings even after working a full day at her job. I thank the Creator for our sweet daughter, Shreya Ray. Her smile lightens my heart and gives me the strength to move forward with my life.

[1] © 2006 IEEE.

[2] © 2007 IEEE.

[3] This work was supported by Intersil and CORE under contract 21-6771 and 02-10110.

I want to acknowledge the thought provoking discussions with the people with whom I have shared office space: Seung-Tak Ryu, Yun-Shiang Shu, Supisa Lerstaveesin, Jin-Ho Choi, Yiping Han, Nikhil Vaidya, Rahul Kodkani, Kok-Lim Chan, Hidenobu Takeshita, Yoshiaki Konno, Akio Maruo, Takuya Ohi, Katsumi Ozawa, Kazuki Egawa, Mohammed Farazian, Shadi Dayeh, Moon-Jung Kyung and Manoj Gupta. I am extremely fortunate to interact with graduate students working with other circuit design groups at UCSD – Ashok Swaminathan, Jaspreet Bhatia, Eric Siragusa, Kevin J. Wang, Yang Sun, Sudhakar Pamarti and Andrea Panigada. I acknowledge the contribution of my friends who made my stay at San Diego memorable – Adityakiran Jagannathan, Ali Rangwala, Anuj Mishra, Saurabh Panjwani, Silpa Suthram, Niranjana Nagarajan, Krishna Sekhar, Debashis Panigrahi, Pavankumar Patibanda, Pavan Nugehalli, Sandeep Kanumuri, Mainak Biswas, Shoubhik Mukhopadhyay, Kanishka Lahiri, Saumya Chandra, Anuj Grover, Narayana Prasad Santhanam, Vibha Nahata and Yogen Deshpande. There are several names that I miss here but their impact on my life is treasured.

Several people and organizations have played key parts during the course of my research at UCSD. I would like to thank Cadence, Synopsys, Mentor Graphics and Mathworks for providing the necessary software at reduced costs and MOSIS fabrication services for providing discounted access to the semiconductor foundry. I acknowledge the part played by Kanti Bacrania and his team at Conexant Inc. for supporting our lab and our research. I have had a memorable stay at Palm Bay, Florida during my internship at Conexant in 2004. I would like to thank Shola Agbelemose and Kos Chitle of Flexible Manufacturing Inc. for fabricating our test system and also for assembling the last crucial test board at no additional charge. I would like to thank Prof. M. B. Rao of the Univ. of Cincinnati for discussing the statistical implications of the self-configuration scheme that forms the heart of this dissertation.

During the course of working on my dissertation, I have had my up and down moments. I would like to thank my running buddies from Asha for Education who ran

26.2 miles with me – this was my first marathon medal. I realized how similar the Ph.D. program is to running a marathon - the last few miles are conquered by will power alone. I would also like to thank Pt. Kartik Seshadri for teaching me how to appreciate music and play the sitar. I would like to thank my colleagues from Texas Instruments: Sumeet Mathur, Preetam Charan Anand Tadeparthy, Jomy Joy, Subhashish Mukherjee, Srinivasan Venkatraman, Ganapathy Subramanium, Vivek Pawar among others who have taught me circuit design and from Agilent Laboratories: Ken Poulton, John Corcoran, Brian Setterberg, Dan Huber, John Keane, Ken Nishimura, Don Pettengill, Jackie Liu, Bob Jewett, Claudio San Roman, Jeff Galloway, Jake Wegman, Gunter Steinbach, Robert Neff for being such excellent colleagues.

Last but not the least; I would like to thank my extended family – the Palkars, the Mitras, the Chatterjees, the Dhavles and the DePasquales for quality personal time they have spent with me.

VITA

1995	Higher Secondary School Certificate, Fr. Agnel Jr. College, Vashi, Navi Mumbai, INDIA
1999	Bachelor of Engineering (Electronics Engineering) VESIT, University of Mumbai, Mumbai, INDIA
1999 - 2000	Systems Engineer Wipro Technologies Chennai, INDIA
2000 – 2002	I.C. Design Engineer Texas Instruments (India) Ltd. Bangalore, INDIA
2002 – 2005	Graduate Student Researcher Department of Electrical and Computer Engineering University of California, San Diego La Jolla, CA, U.S.A.
2004	Graduate Student Intern Conexant Inc. Palm Bay, FL, USA
2005	Master of Science Degree (Electrical Engineering) University of California, San Diego La Jolla, CA, U.S.A.
2005 – 2008	Hardware R & D Engineer Agilent Technologies Santa Clara, CA, USA
2008	Doctor of Philosophy Degree (Electrical Engineering) University of California, San Diego La Jolla, CA, U.S.A.

PUBLICATIONS

Sourja Ray, Bang-Sup Song, “A 13-b Linear, 40-MS/s Pipelined ADC With Self-Configured Capacitor Matching,” *IEEE J. Solid-State Circuits*, vol.42, no.3, pp.463-474, March 2007

S. Ray, Bang-Sup Song, “A 13b linear 40MS/s pipelined ADC with self-configured capacitor matching,” *ISSCC Dig. Tech. Papers*, pp. 852-861, Feb. 2006

Seung-Tak Ryu, S. Ray, Bang-Sup Song, Gyu-Hyeong Cho, K. Bacrania, “A 14-b linear capacitor self-trimming pipelined ADC,” *IEEE J. Solid-State Circuits*, vol.39, no.11, pp. 2046-2051, Nov. 2004

Seung-Tak Ryu, S. Ray, Bang-Sup Song, Gyu-Hyeong Cho; K. Bacrania, “A 14 b-linear capacitor self-trimming pipelined ADC,” *ISSCC Dig. Tech. Papers*, pp. 464-540 Vol.1, Feb. 2004

ABSTRACT OF THE DISSERTATION

Calibration of Multi-Bit per Stage Pipelined ADC

Using Statistical Properties of Capacitor Arrays

by

Sourja Ray

Doctor of Philosophy in Electrical Engineering

University of California, San Diego, 2008

Professor Bang-Sup Song, Chair

With the rapid growth of powerful digital algorithms in communications and control technology, it is now considered advantageous to partition systems such that more traditionally analog signal conditioning tasks like filtering, equalization, frequency translation are now being performed using digital circuits. In such systems, it is necessary burden the data converter to perform the conversion at higher speeds and higher resolutions to maximize the amount of information flowing in to and out of the digital domain. The analog-to-digital converter that links the “real” analog world with the computationally convenient digital domain is often a performance bottleneck in such systems. This dissertation is a study of analog techniques applied to multi-bit

DACs that are required to overcome the analog limitations that makes it difficult to achieve high accuracy analog-to-digital and digital-to-analog converters.

The improvement of accuracy is achieved in the analog domain using a novel technique called self-configuration. Using statistical matching properties of capacitor arrays, a pipelined ADC self-configures the MDAC capacitor array for best matching from many trial combinations of smaller capacitive sub-elements. These sub-elements having opposite error magnitudes are grouped together to form matched elements thus permitting an accurate multi-bit MDAC to be created without using an explicit trimming network. A random search algorithm enables the self-configuration process by quickly regrouping the sub-elements to reduce the spread between the reconstructed elements. The proposed state machine based permutation algorithm allows near unique permutations of the sub-elements and achieves a near unity state repetition ratio with a simple hardware implementation. An ADC system is designed with the self-configuration algorithm contained in the same die, and improvement in capacitor matching is demonstrated after the self-configuration process. A 0.18μm CMOS prototype achieves 13b linearity and over 80dB SFDR at 43MS/s. The chip consumes 268mW at 1.8V and occupies 3.6mm^2.

A short study of an existing alternate analog calibration scheme is presented as a comparison. This scheme helps improve the matching of elements in the feedback DAC of a multi-bit delta-sigma ADC. Opportunities to improve the scheme for high-speed operation are identified and proposed solutions are verified using simulations.

Chapter 1:

Introduction

A review of the basic concepts necessary to understand data converters and the important figures of merit used to characterize data converters is presented followed by a short description of common ADC architectures relevant to this dissertation.

1.1: INTRODUCTION

Man, since the beginning of civilized times, has attempted to control his surroundings with increased precision [1]. The industrial era heralded an age of control where mechanical, hydraulic and pneumatic systems improved quality of life. The electronic age improved the precision of control through analog computers and analog control systems. The conversion of mechanical stimulus to electronic quantities like charge, current and voltage followed by processing using various circuits like amplifiers, integrators and differentiators and converting back to mechanical response resulted in a higher precision of control than by using mechanical systems alone. The world shrank with the advent of telegraphy and telephony. The digital revolution brought a further increase in the complexity of control and communication systems. The combination of complex algorithms and the repeatability and ease of manufacturing digital circuits and digital memory made it economical to process all real world stimuli in the digital domain and convert the response of the system from the digital domain to the real world. It is counter-intuitive to note that as systems improve in performance, the actual control happens in a layer that is further away from the environment being controlled (Fig. 1.1).

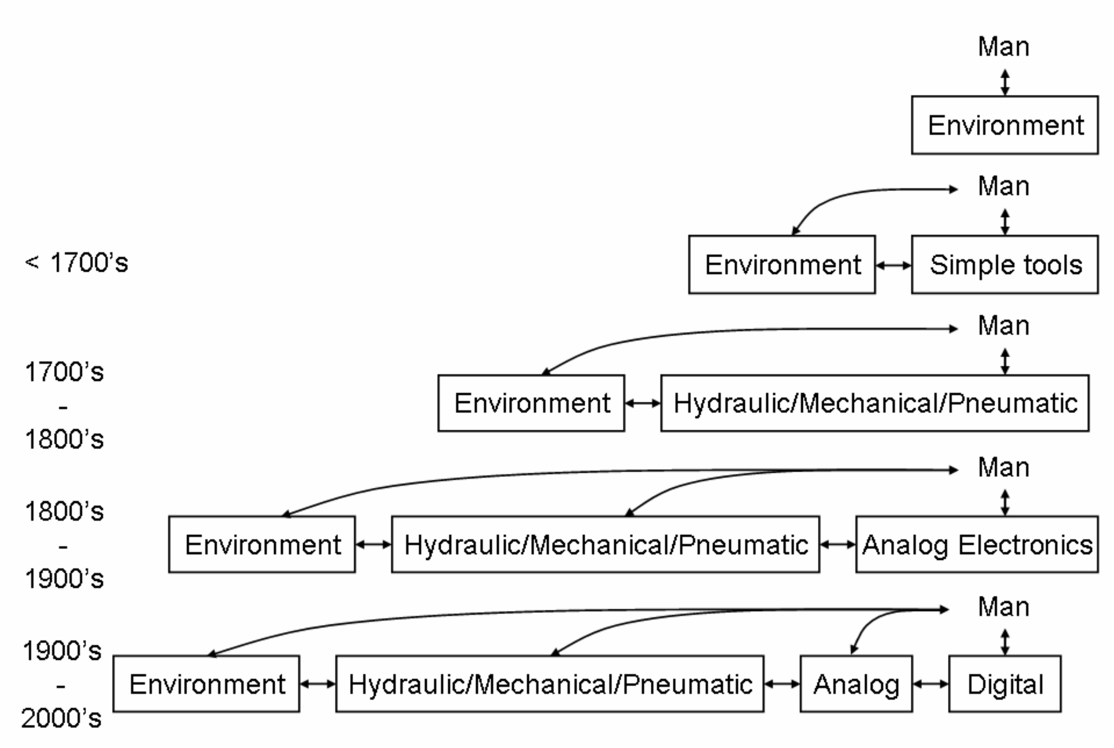


Figure 1.1: Evolution of technology.

The discrete-valued and discrete-time nature of the digital domain is an abstraction of the continuous-valued and continuous-time nature of the real analog world. Sensors convert the real world quantities of interest to analog signals. Analog signals exert influence back to the real world using actuators. Data converters transform information from the analog domain to the digital domain and back using analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). The performance of the overall system not only depends on the algorithm exerting the control but also on the quality of each conversion step. The ADC is often the bottleneck in performance in several modern information-processing systems.

1.2: DISSERTATION ORGANIZATION

This chapter describes various ADC architectures and the limitations of these architectures. The second chapter describes calibration systems that alleviate these problems. A novel technique proposed in the third chapter reduces the mismatch in nominally alike unit elements used in DACs. A pipelined ADC chip that uses this mismatch reduction technique to perform analog domain calibration of unit elements in a multi-bit MDAC is further discussed in the fourth chapter and forms the heart of this dissertation. The fifth chapter describes an alternate analog calibration scheme applied towards reducing the mismatches of unit elements in a multi-bit feedback DAC in a continuous-time delta-sigma ADC.

1.3: BUILDING BLOCKS

Comparators and digital-to-analog converters (DAC) are two fundamental building blocks that can be identified in all ADC architectures. The comparator is a non-linear block that compares the input analog variable (voltage, charge or current) with an applied reference value (voltage, charge or current) to generate a decision indicating if the applied signal was greater than or lesser than the reference value (Fig. 1.2). This decision-making capability of this building block allows the signal range to be partitioned into distinct regions and each region is associated with a code value. Several decisions are made using either one or several comparators during the analog-to-digital conversion process and the resulting decisions are combined in an architecture specific manner to generate the final digital output code word. The DAC provides a way to map the digital code space back to the input analog quantity.

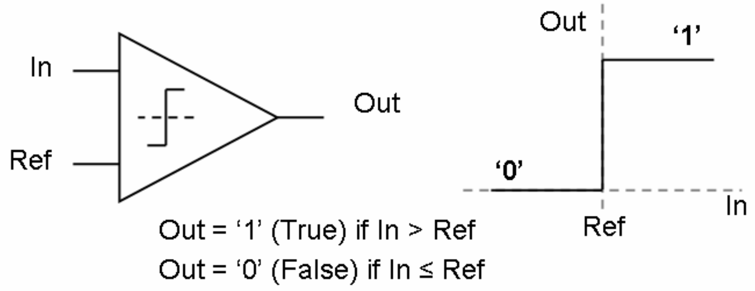


Figure 1.2: Comparator as a decision-making element.

In the strictest sense, a digital system operates on discrete quantities. For ease of implementation, most digital systems are also implemented as discrete-time systems i.e. digital systems accept or deliver data at discrete time steps. This fits very well with the analog sampling concept and a sampling system like a sample-and-hold or a track-and-hold network is often a part of an ADC system. Sampling freezes the signal at a chosen time instant. This allows a relationship to be set between the analog signal and the digital codes versus time. Uniform sampling is preferred for most data converters in which the samples are uniformly spaced in time. The rate of sampling determines the maximum bandwidth of signal that can be reconstructed as specified by the Nyquist sampling theorem [2]. A signal band-limited to B Hz to be reconstructed without artifacts has to be sampled at 2B Hz, called the Nyquist sampling rate, as shown in Fig. 1.3. Signals at frequencies greater than B Hz, when sampled at 2B Hz, fold back or alias into the signal band. An anti-alias filter (AAF) is commonly used at the input of an ADC to prevent aliasing. ADCs that can potentially operate at signal frequencies until half the sampling rate are called Nyquist rate ADCs.

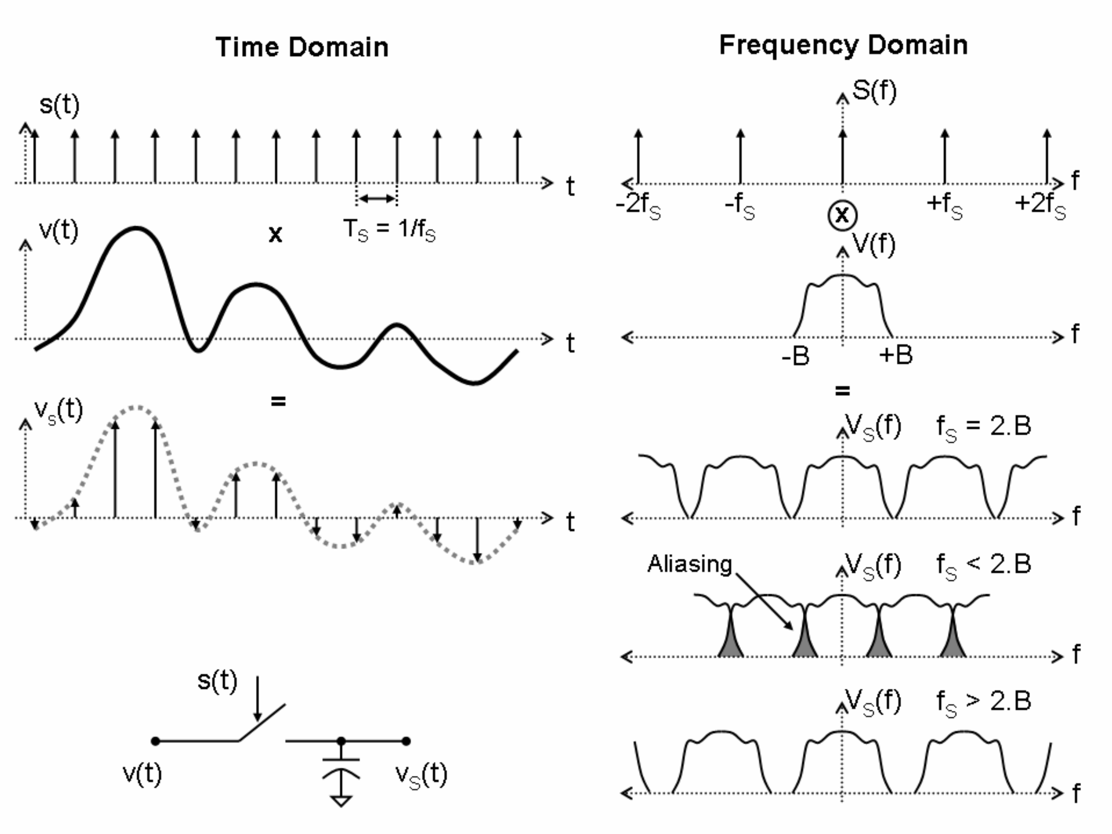


Figure 1.3: Sampling.

A finite number of bits in a word are used in a realizable digital system to represent the continuous analog signal. The signal is converted to one of a finite number of possible codes in the digital code space or in other words is quantized (Fig. 1.4). ADCs therefore are also called quantizers. The code is usually expressed in terms of these bits forming a binary number. The binary representation is common in practical ADCs because this is a leads to simple relationship between the digital code and the analog input value. The fewest number of bits that can be used to express uniquely each allowable code is called the resolution of the ADC. The resolution of an ideal ADC also represents a measure of the smallest change in the input that can be detected i.e. the minimum resolvable input. The smallest change results in a change in the least significant bit (LSB) in the binary word. LSBs are frequently used as units to measure the signal range and size of errors in a data converter.

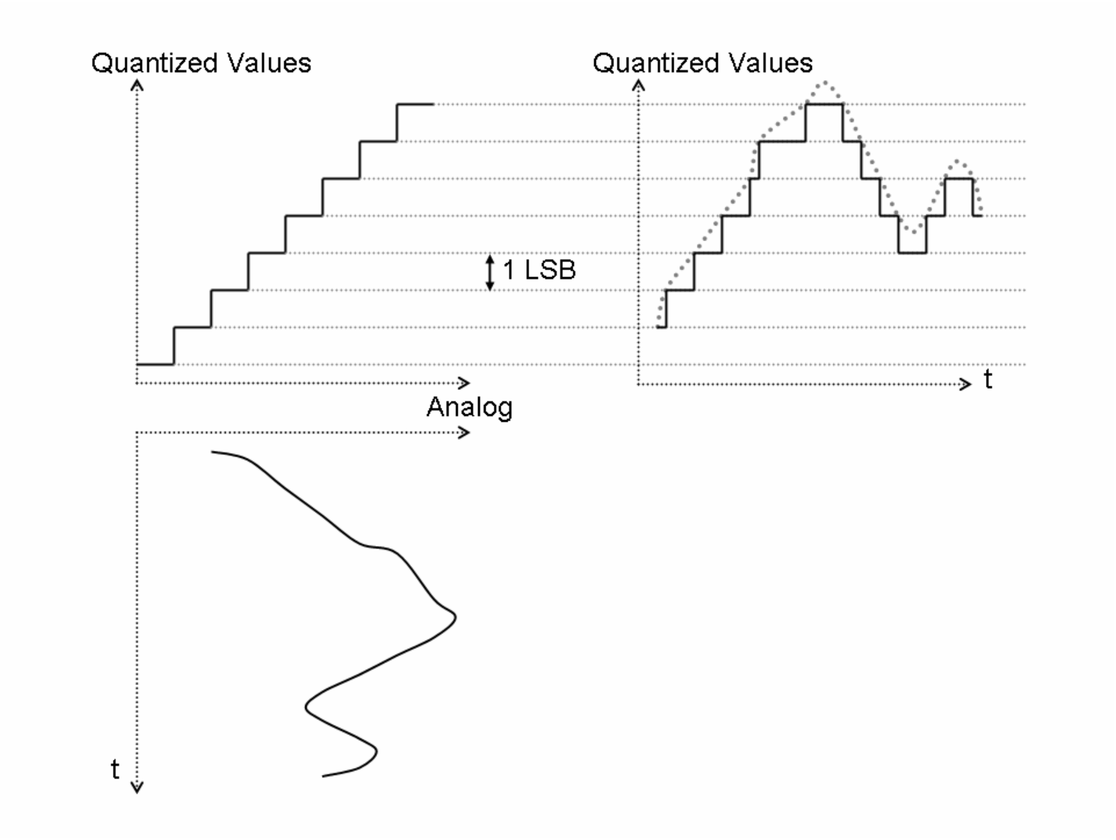


Figure 1.4: Quantization.

ADCs consist of other building blocks such as amplifiers and reference sources in addition to the comparator, the DAC and the sampling network. Impairments in each of these building blocks limit the performance of a data converter. Few commonly encountered errors in these building blocks are described in the following section.

1.3.1: Comparators

A comparator typically consists of a pre-amplifier followed by a regenerative latch shown in Fig. 1.5.

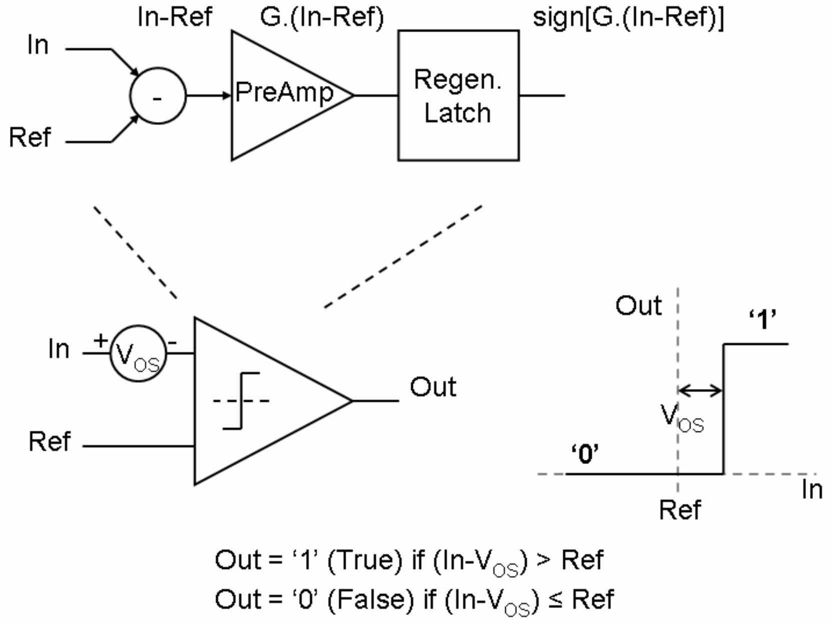


Figure 1.5: Offset in comparator.

Due to mismatches in the transistors [3], the comparator circuit is not perfectly balanced and because of this, the comparator does not trigger exactly at the point where the input crosses the reference value. This is modeled using an equivalent input referred offset. Such an offset clearly has an impact on the quantization step size and this can lead to errors in the final converted value. Offsets can be reduced by using large sized devices at the cost of increased area and power. Alternatively, offset compensation schemes have been suggested [4] at the expense of added circuit complexity.

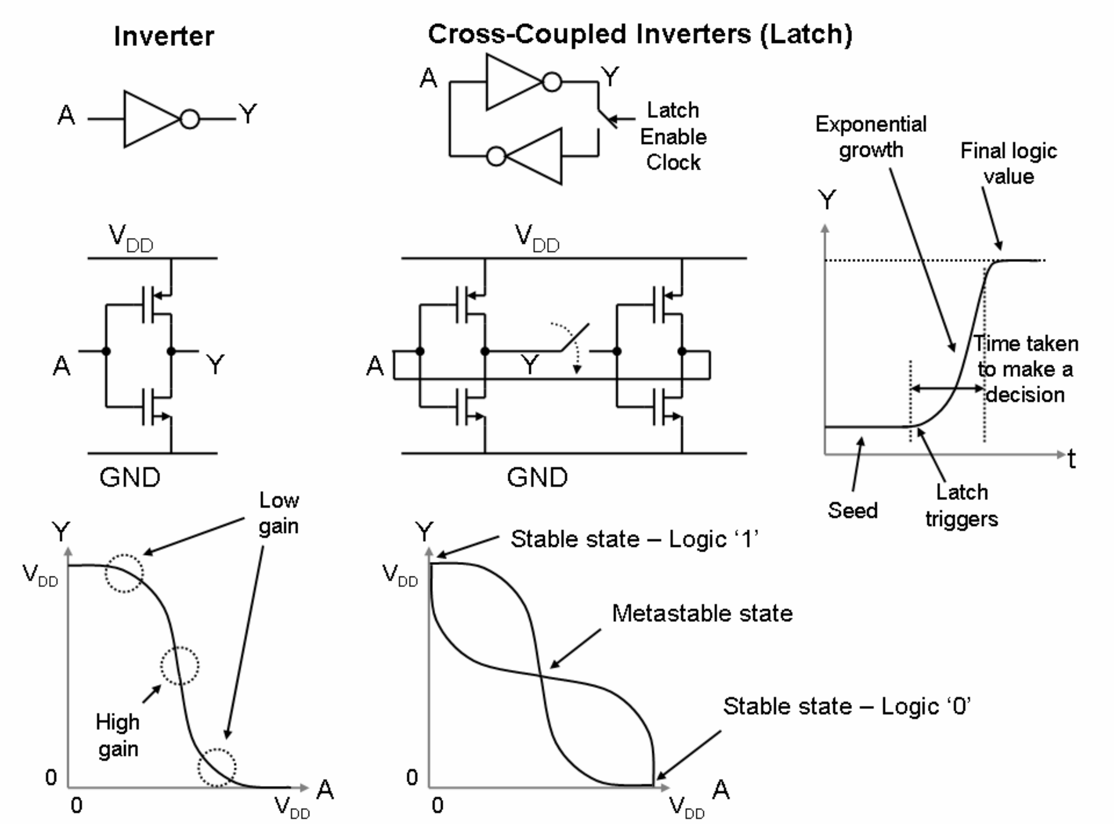


Figure 1.6: Regeneration in comparator.

The regenerative latch relies on a circuit with positive feedback to make a decision quickly. All latch designs can be equivalently drawn as cross-coupled inverters acting on a seed i.e. the initial voltage across the latch just before it is triggered. This is shown in Fig. 1.6. As the latch voltage grows, the gain of the devices that form the latch reduces and the amount of positive feedback reduces thus reaching a stable decision point. The regeneration time constant of a latch is determined by the architecture of the latch and the transition frequency of the transistors used in the latch. The transition frequency of a transistor is the frequency at which the magnitude of the transistor intrinsic gain falls to unity due to parasitic capacitances and is a characteristic of the semiconductor process technology. A large regeneration time constant implies that the latch takes longer to regenerate if the seed voltage V_{SEED} is small and it takes longer for the outputs of the latch to grow to voltages that correspond to valid logic levels. This contributes to a signal dependant delay in the output decision of the latch and this sets the maximum possible sampling rate for a chosen ADC architecture for the chosen semiconductor process technology.

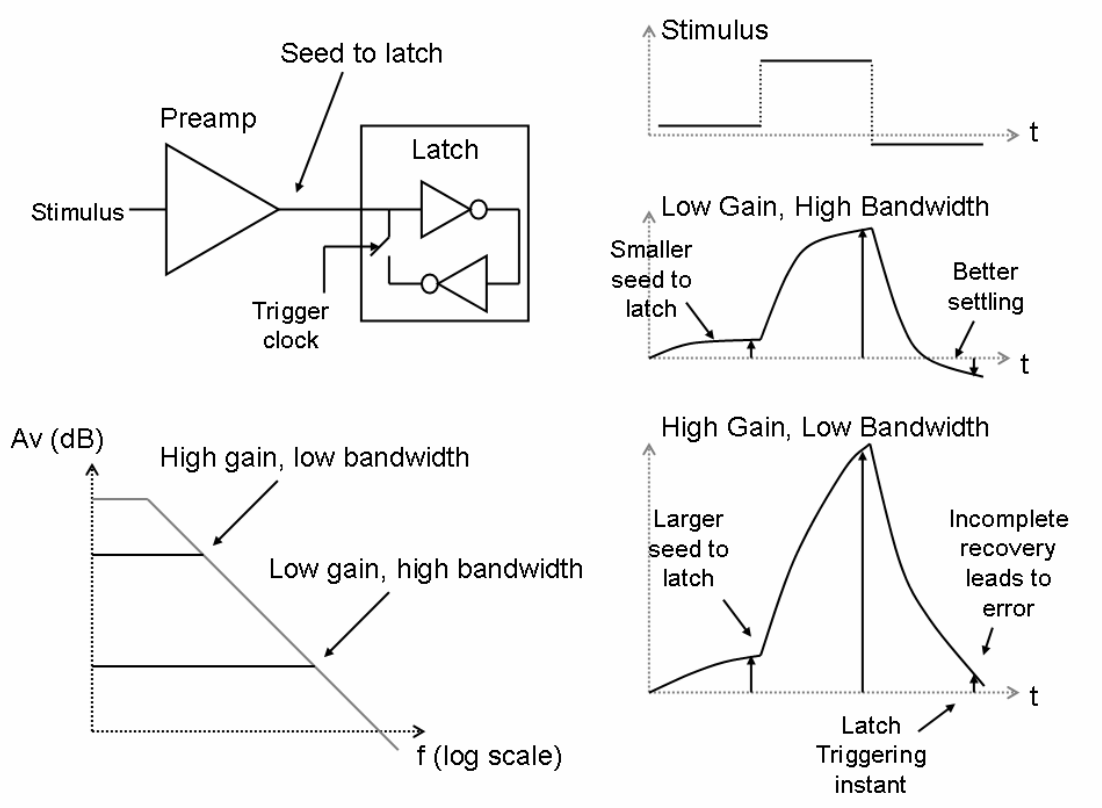


Figure 1.7: Bandwidth and gain tradeoff in comparator.

The pre-amplifier driving the comparator provides gain to minimize the effect of offset from the latch and helps reduce the probability of a metastable event by increasing the gain in the signal path. The choice of a power dissipation target sets the gain bandwidth product of the pre-amplifier. It is advantageous to configure the pre-amplifier to obtain more gain at the expense of losing bandwidth as shown in Fig. 1.7

and this is preferred for small input signals. However, reducing the bandwidth too much affects the recovery of the comparator from large input values. This is observed as a signal history dependent hysteresis in the comparator-switching threshold and is avoided by proper design choices.

1.3.2: Sampling network

The sampling network in an ADC freezes the value of signal at a chosen time instant. One of the edges of a sampling clock governs the sampling instant. Error in a sampling network is primarily attributed to clock jitter [5].

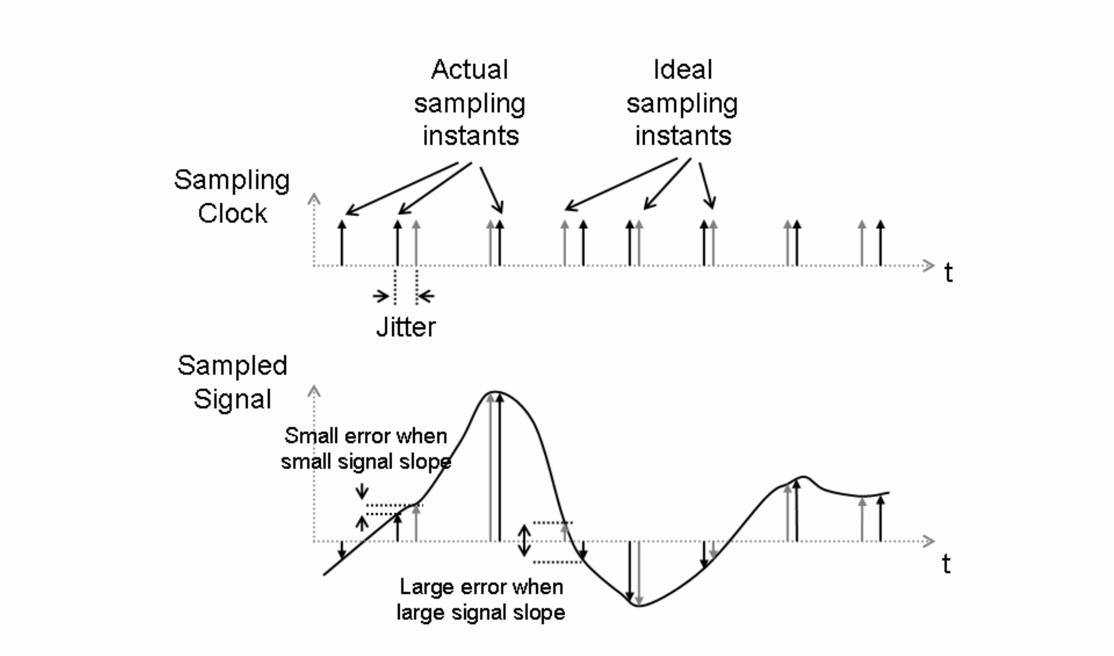


Figure 1.8: Jitter in sampling network.

Fig. 1.8 shows that clock jitter causes the signal to be sampled at an incorrect time. The amount of error is a function of the signal amplitude and the rate of change i.e. the frequency of the signal. The sampling function is frequently implemented using a sample and hold switch as shown in Fig 1.9. Charge feed-through from the switch overlap capacitances and the channel charge injection during the switch turn-off transition also corrupt the charge stored. Non-linear effects in the switch affect the value of the sampled signal. Various techniques [6], [7] have been proposed to reduce the effect of these errors and should be considered during the design of the sampling system.

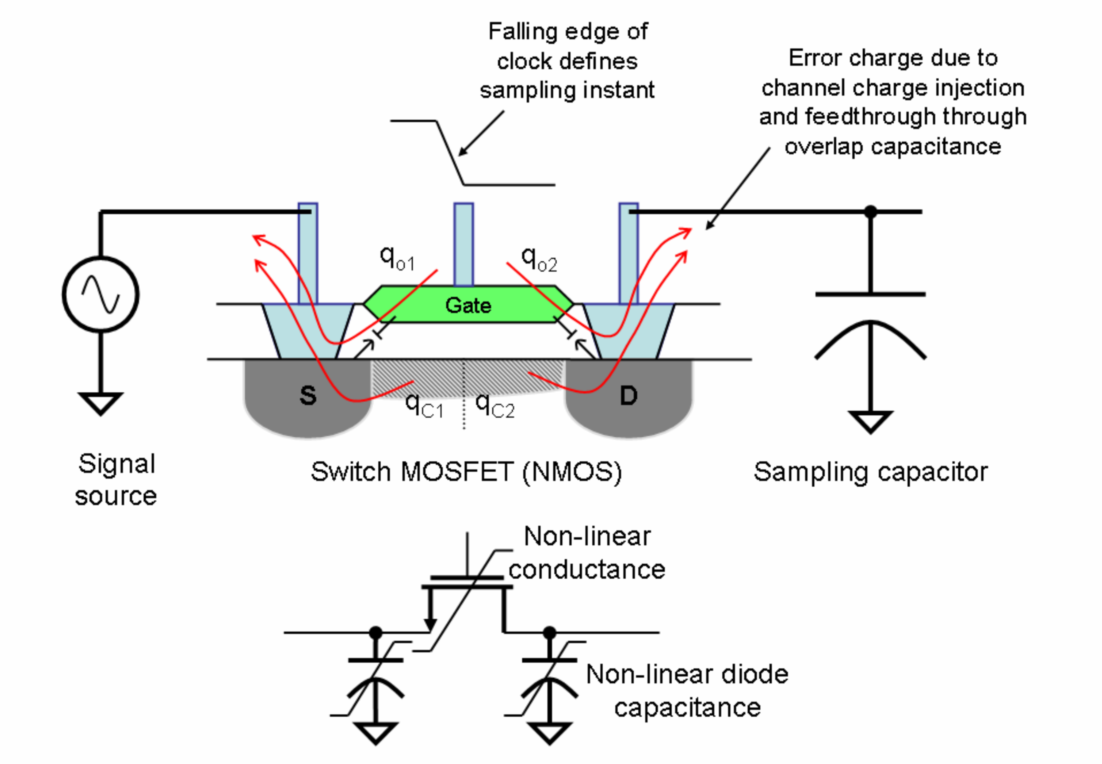

Falling edge of clock defines sampling instant
Error charge due to channel charge injection and feedthrough through overlap capacitance
q_{o1}
q_{o2}
Gate
S
D
q_{C1}
q_{C2}
Signal source
Switch MOSFET (NMOS)
Sampling capacitor
Non-linear conductance
Non-linear diode capacitance

Figure 1.9: Errors due to sampling switch.

1.3.3: DAC

The DAC in an ADC converts a digital code to an analog quantity proportional to the value of the code in a linear time invariant fashion. In the simplest case this is achieved by using an array of elements, each activated by a bit of the digital code.

DACs are typically constructed either using binary weighted devices or using matched or equal weighted devices as shown in Fig. 1.10. In a DAC based on a binary weighted array, the nth binary bit activates an element that is sized *n* times the unit element. In a DAC based on an equal weighted array of matched elements, the N-bit binary word is first converted to a thermometric code with 2^N-1 bits and each of these bits activate a unit element. Inaccuracies in the elements of a DAC lead to static errors in the transfer function. An advantage of a DAC based on a thermometric array is that the transfer function is strictly monotonic.

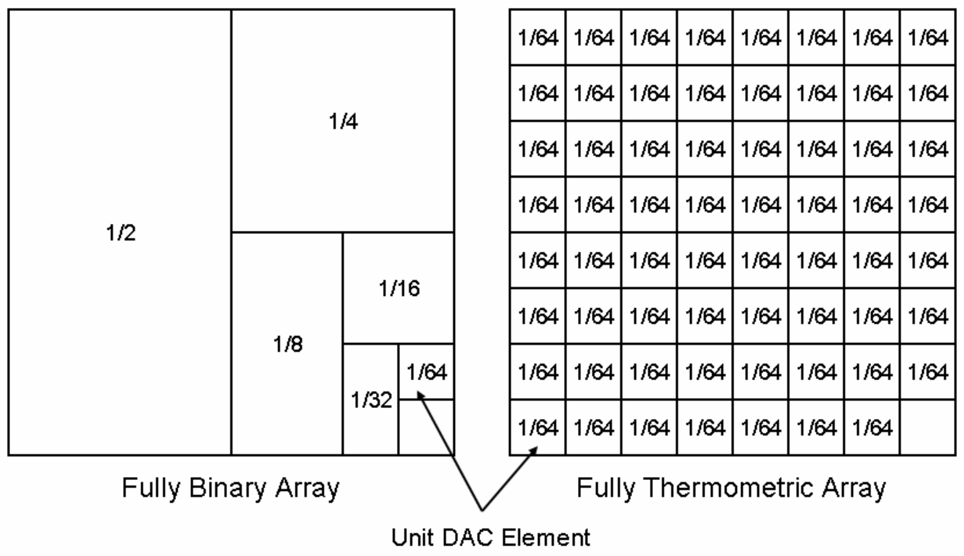


Figure 1.10: 6-Bit DAC array in binary and thermometric arrangements.

The other cause of errors in a DAC is due to dynamic effects. In a DAC based on a resistor ladder driving a flash ADC, due to disturbances at the different nodes and due to finite resistance, the output value shows a position dependent error vs. time. In a current steering DAC, due to unequal rise and fall transients, the output shows a signal dependent error [8].

1.3.4: Amplifiers

Certain ADC architectures like pipelined ADCs (described later in this chapter) use amplifiers to amplify the size of signals between sub-conversions by a precise amount and these sub-converted decisions are combined together using digital gains that match the analog path gains to lead to a higher resolution code as shown in Fig. 1.11.

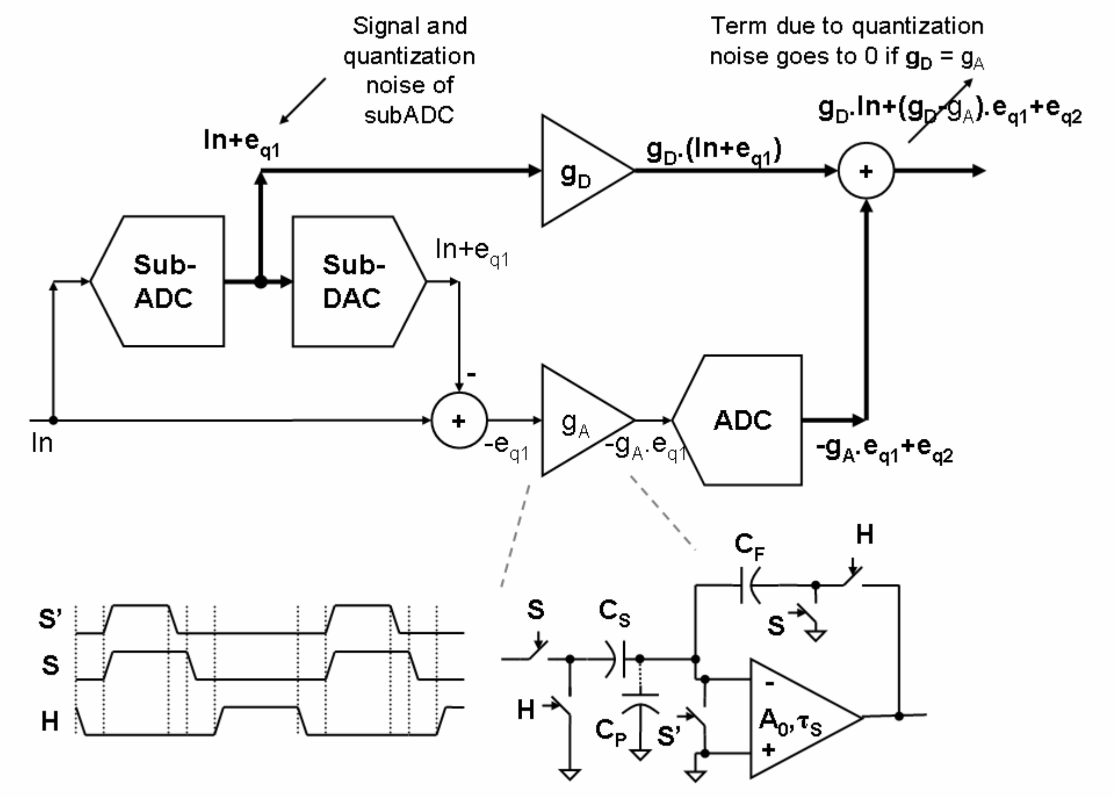


Figure 1.11: Use of precise gains in ADCs.

Inaccuracies in the amplifier gain leads to an error in the final reconstructed digital code. Consider the case of a switched capacitor amplifier providing an accurate gain of g_A set by the ratio of C_S and C_F that are the sampling and the feedback capacitors. Eqn. 1.1 shows that the gain is corrupted by the parasitic capacitance to ground from the summing node C_P and the operational amplifier open loop gain A_0 and these should be minimized in order to obtain accurate analog path gain.

$$V_{oIDEAL} = V_i \cdot \left(\frac{C_S}{C_F} \right)$$

$$V_{oACTUAL} = V_i \cdot \left(\frac{C_S}{C_F} \right) \frac{1}{\left(1 + \frac{(C_S + C_F + C_P)}{A_0 C_F} \right)}$$

$$V_{NormalizedERROR} = \frac{V_{oIDEAL} - V_{oACTUAL}}{V_{oIDEAL}} = 1 - \frac{1}{\left(1 + \frac{(C_S + C_F + C_P)}{A_0 C_F} \right)}. \qquad \ldots \text{Eqn. 1.1}$$

The rate of change of the output value of an amplifier is limited by either the slew rate or the finite bandwidth i.e. the settling time constant τ_S of the amplifier. The finite settling time at the output of an amplifier, assuming a single pole model, leads to an error in the output at a chosen sampling instant T_S expressed by Eqn. 1.2. The bandwidth is chosen such that the settling error is contained within limits allowed by the system error budget.

$$V_o(t) = V_o(\infty) \cdot \left(1 - e^{-\frac{t}{\tau_S}} \right)$$

$$\frac{V_{ERROR}(t)}{V_o(\infty)} = \frac{V_o(\infty) - V_o(t)}{V_o(\infty)}$$

$$V_{NormalizedERROR}(T_S) = e^{-\frac{T_S}{\tau_S}}. \qquad \ldots \text{Eqn. 1.2}$$

1.4: FIGURES OF MERIT

It is important for the system designer to specify the desired quality of the conversion when a data converter is used as a part of a larger system. The various figures of merit capture the essential behavior of a data converter for a chosen application space and are used to quantify the impact of adding a data converter to the system being designed.

1.4.1: Signal to noise ratio (SNR)

The signal to noise ratio or the SNR is a commonly specified figure of merit. An upper bound of the achievable SNR is determined by the effects of quantization and is often called the signal to quantization noise ratio (SQNR). The process of quantization partitions the signal space to finite number of discrete regions each of which map to a code in the output code space. The simplest case is that of a uniform quantizer with an N-bit output code that divides the signal space in 2^N equal regions. The difference between the input signal applied to the quantizer and the quantized output values is the error in the conversion or the quantization noise. An assumption made during this analysis is that the input signal is a sinusoid that occupies the full signal range. If the signal range is normalized to unity, the sinusoid has amplitude of 0.5 and power of $0.5^2/2.0$. For a large number of quantization steps, the quantization noise is assumed to have uniform distribution with spread of $\pm 2^{-N}/2.0$ around a mean of zero. The power in the quantization noise is the variance of this distribution and for a uniformly distributed quantization noise can be calculated to be $2^{-2N}/12.0$ [9].

$$SNR = 10 \cdot \log_{10}\left(\frac{P_{Signal}}{P_{QuantizationNoise}}\right)$$

$$SNR = 10 \cdot \log_{10}\left(\frac{0.5^2 / 2.0}{2^{-2N} / 12.0}\right)$$

$$SNR = 10 \cdot \log_{10}\left(1.5 \cdot 2^{2N}\right)$$

$$SNR = 6.02 \cdot N + 1.76 \qquad \text{… Eqn. 1.3}$$

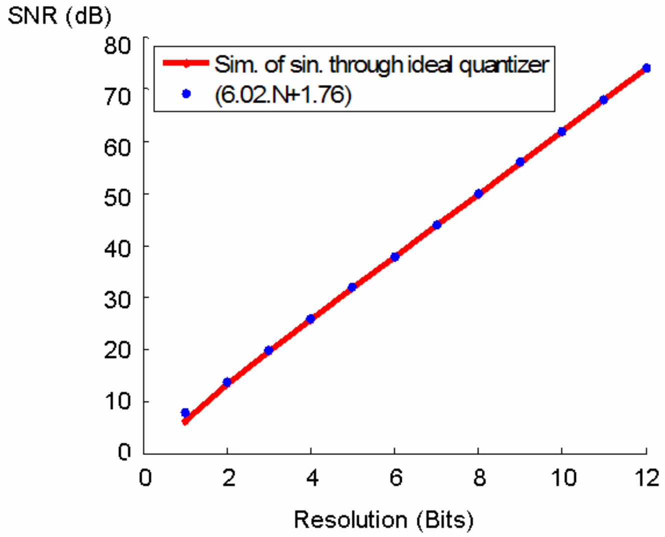


Figure 1.12: Quantization noise vs. resolution in an ideal ADC.

Fig. 1.12 shows the SNR from simulation of a quantizer vs. the number of bits in the ADC when excited using a sinusoid that occupies the full available input range. The uniform distribution approximation of the quantization noise gets better as the number of quantization levels increase. The SNR (in dB) due to the process of quantization by an N-bit ideal ADC is predicted using Eqn. 1.3 and is a very good approximation even for fewer number of bits in the ADC. In order to improve the

SNR of a quantized signal, the resolution of the quantizer or the number of bits of the quantization must be increased. The amount of quantization noise is halved for every additional bit of resolution as a result the noise power is reduced by a factor of four thus increasing the SNR by 6.02 dB. This result is commonly stated as *"every added bit of resolution improves the SNR by 6 dB"*.

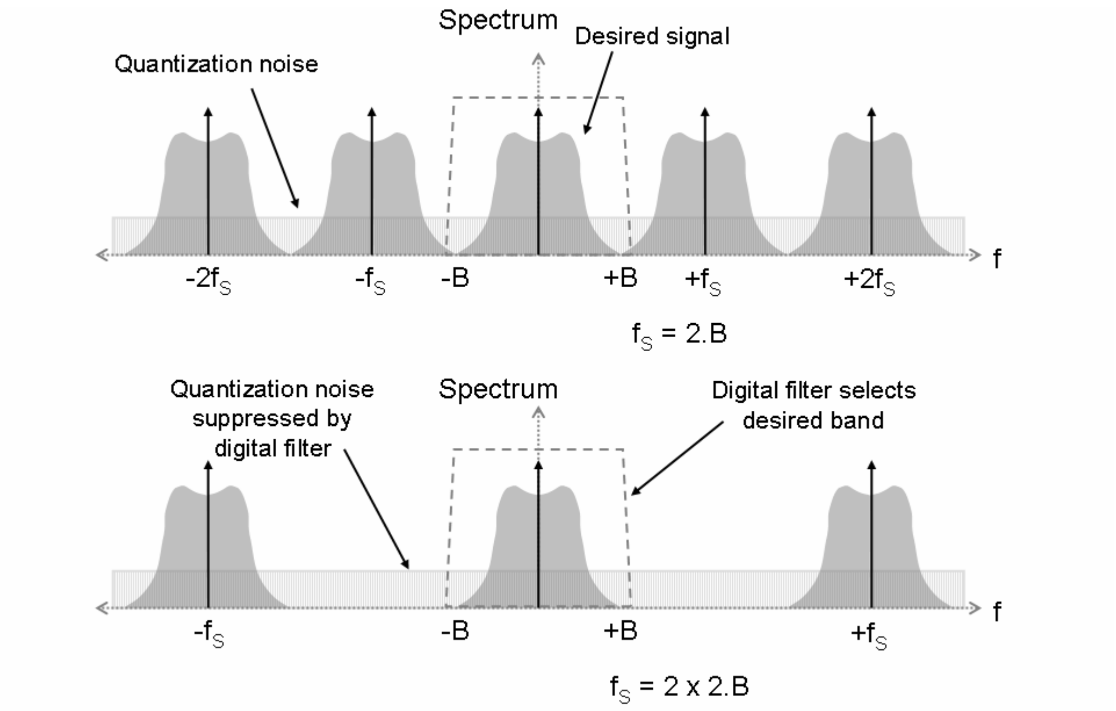


Figure 1.13: Oversampling.

The process of sampling is mathematically equivalent to multiplying the continuous-time signal by a series of unit impulses spaced at discrete time intervals. A spectrum of a signal in the frequency domain repeats at the sampling rate when sampled. The noise due to the process of quantization spreads out in the frequency domain and this is no longer limited by the original band containing the signal and must fold back. The total noise within the Nyquist band however still remains the same. As a result, Eqn. 1.3 is true even for sampled systems when taking into account the noise in the entire Nyquist band. However if the signal of interest only occupies a chosen sub-band, as an example only the lower half of the available Nyquist band, the output of the quantizer can be filtered using a digital filter to only extract information in this band as shown in Fig. 1.13. This is called oversampling i.e. sampling at a higher rate than necessary. Only half the quantization noise power is present at the output of this filter since the quantization noise is uniformly spread over the Nyquist band but the entire signal power is recovered. This improves the SNR by $10.\log_{10}(2.0)$ or by 3.01 dB. This result is commonly stated as *"oversampling by every factor of two improves the SNR by 3dB"*.

1.4.2: Integral and differential non-linearity (INL and DNL)

The integral and differential non-linearity (INL and DNL) are fundamental figures of merit useful for characterizing the transfer function of ADCs as well as DACs. These are indicators of the linearity of a data converter vs. the output code in the case of an ADC are very useful diagnostic tools that are useful to determine the sources of impairments that cause errors in the conversion.

The DNL is a measure of deviation of the actual step size for an output code from the ideal or the average step size measured in units of LSBs and plotted versus the output code. The DNL is useful to characterize the effects of errors when a small input signal is quantized since this predicts the noise-like errors in the output. The INL is the integral of the DNL and represents the deviation of the actual transfer function from the ideal or average transfer function measured in units of LSBs and plotted versus the output code. The DNL is conversely the derivative of the INL. The INL is useful to characterize the effects of errors when a large input signal is quantized since this predicts the harmonic distortion-like errors in the output. The INL and DNL are used to identify features like missing codes, non-monotonic transfer functions and systematic and random errors that point to limitations of the circuits. The smallest step that can occur in a transfer function is no step at all and this sets the minimum possible value of DNL to -1 representing missing codes.

The INL and DNL can be measured by applying a well-known uniform signal like a ramp to an ADC and plotting the actual code vs. the expected code value. Practically, however, this technique is only limited to characterizing ADCs having accuracy of less than 10 bits since creating a linear and accurate ramp stimulus at the operating speed of high speed ADCs is difficult. The sine wave histogram method [10], [11] is used in a high-resolution high-speed system. In this, a sinusoid having a relatively prime frequency with respect to the sampling frequency is chosen as the stimulus and a histogram of output codes is created. The measured code histogram is then modified by the inverse density function of a sinusoid to obtain a histogram as if a uniformly distributed input was used to excite the ADC. Pure sinusoids at various frequencies are easily generated using precise signal sources followed by narrow band pass filters and provide a stimulus with typically much higher accuracy than a ramp stimulus. This stimulus can also be used to characterize the performance of the ADCs at various frequencies. This method also is used to compute the INL and DNL plots for the ADC described in Chapter 4 in this dissertation (Fig. 4.17 and 4.18). The disadvantage of the code density test is that non-monotonic transfer functions cannot be accurately detected and any noise in the system causes an averaging effect as a result of which sharp discontinuities in the INL are smeared out thus causing a misleading and optimistic DNL result.

1.4.3: Impact of circuit noise on SNR

Every ADC is designed using real analog components that exhibit noise inherent to the operation of the device. Circuit noise is primarily due to the fact that charge is a discrete quantity that depends on the number of electrons and current flow is not continuous but the consequence of electrons (or other charge carriers) crossing a cross section area of a conductor in a unit time. The simplest noise model is explained by the Brownian motion of charge carriers in a conductor under equilibrium conditions. For a resistor *R*, the root mean square noise voltage examined over a bandwidth *B* is expressed by Eqn. 1.4, which forms the basis of noise modeling in all practical circuits of importance for an analog/RF designer. Here *k* is the Boltzmann's constant and *T* is the absolute temperature of the system. Such a noise, called white noise, has a Gaussian probability density function and a power spectral density that is flat over the bandwidth of interest.

$$\overline{v_n(R,B)} = \sqrt{4 \cdot k \cdot T \cdot R \cdot B} \qquad \text{… Eqn. 1.4}$$

This equation is used as a starting point for the equation of noise in the channel of a MOSFET [12]. Another noise equation of interest to circuit designers is the shot noise due to charge carriers crossing a P-N junction carrying an average current *I*, which is expressed by Eqn. 1.5 where *q* is the charge carried by an electron. The shot noise equation is used to model the noise current in a BJT.

$$\overline{i_n(I,B)} = \sqrt{2 \cdot q \cdot I \cdot B} \qquad \text{… Eqn. 1.5}$$

An important conclusion of interest in switched capacitor designs is Eqn. 1.6a, which is the case of the noise from a resistor integrated across a capacitor. This is a very useful expression since this is used as a starting point to budget noise in switched capacitor systems where the driving circuit and the switch appears as a resistance to the capacitor. This equation is multiplied by a factor *γ* (Eqn. 1.6b) to account for additional sources of noise added by active circuits in amplifiers and other noise mechanisms.

$$P_n(C) = \frac{k \cdot T}{C}. \qquad \text{… Eqn. 1.6a}$$

$$P_n(C) = \gamma \frac{k \cdot T}{C}. \qquad \text{… Eqn. 1.6b}$$

Circuit noise adds an additional amount of error during the quantization process not accounted for by the quantization noise. This degradation in the quality of the quantization process is expressed by the modifying the equation of the SNR as shown by Eqn. 1.7.

$$SNR = 10 \cdot \log_{10}\left(\frac{P_{Signal}}{P_{QuantizationNoise} + P_{CircuitNoise}} \right) \qquad \text{… Eqn. 1.7}$$

1.4.4: Impact of unequal quantization steps and distortion on ADC performance

Fundamentally, the quantization noise in a data converter can be attributed to very high order harmonics of a signal [13]. Unequal quantization step sizes also add significant distortion terms resulting in spurs in the spectrum of the converted samples.

Real circuits made using BJTs and MOSFETs are not linear and in the static case exhibit compression near the ends of the signal range due to limitations of these transistors. Static distortion components can be predicted by using a polynomial based approach to model such compression. Distortion may also be due to dynamic reasons i.e. due to incomplete settling in the analog signal path or due to non-linear capacitances in the circuit. These memory dependent non-linear effects are more difficult to model and analyze. The Volterra series approach [14] is often used to model these effects. These errors also result in spurs in the spectrum of the converted samples.

It is useful to quantify the effect of these distortion terms through the use of frequency domain related figures of merit namely Spurious Free Dynamic Range (SFDR), Total Harmonic Distortion (THD), Dynamic Range (DR) and Signal to Noise and Distortion Ratio (SNDR). These are measured using a single frequency sinusoidal stimulus at the input of the ADC and further characterized for sinusoids at different frequencies of interest. *Spurious Free Dynamic Range* is a measure of the smallest desirable signal that can be detected when a large sinusoid drives the ADC and in other words relates to the power in the largest undesired spur in the spectrum with respect to the power in the large desired sinusoid expressed in dB. *Total Harmonic Distortion* is defined as the sum of the power of harmonic distortion components in an ADC. Since quantization noise is due to high order harmonic distortion components, this definition is modified for use in ADCs [11] to be the sum of the second to tenth terms of harmonic distortion. The *Dynamic Range* is a measure of the smallest detectable signal in an ADC and is defined as the ratio of the maximum signal power to the power of the signal for which the signal power is equal to the noise power i.e. 0 dB SNR. Conversion errors can be grouped into one figure of merit called the *Signal to Noise and Distortion Ratio* (SNDR, sometimes called SINAD) as in Eqn. 1.8.

$$SNDR = 10 \cdot \log_{10}\left(\frac{P_{Signal}}{P_{QuantizationNoise} + P_{CircuitNoise} + P_{Distortion}} \right). \qquad \text{… Eqn. 1.8}$$

The *Effective Number of Bits* (ENOB) is a figure of merit derived from the SNDR as in Eqn. 1.9 and represents the resolution of an ideal ADC equivalent to the ADC being characterized.

$$ENOB = \frac{(SNDR - 1.76)}{6.02}. \qquad \text{… Eqn. 1.9}$$

The SNDR is often plotted vs. the input frequency and this is used to compute the *Effective Resolution Bandwidth* (ERBW) of an ADC defined as the frequency at which the SNDR drops by 3 dB from the specified value. This is used to define the *Figure of Merit of Energy Consumption* (E_{FOM}) of a data converter as defined in Eqn. 1.10 measured in Joule-per-conversion step. This is often quoted in competitive literature where the power efficiency is considered the crucial factor in the implementation of a system.

$$E_{FOM} = \frac{P_{CHIP}}{2^{ENOB}} \cdot (2 \cdot ERBW) \text{ .} \qquad \text{… Eqn. 1.10}$$

1.5: COMMON ADC ARCHITECTURES

1.5.1: Flash ADC

The flash architecture is conceptually the simplest ADC architecture. A flash ADC makes 2^N-1 decisions simultaneously in order to achieve N bits of resolution. This is achieved using a parallel search approach in which a DAC, commonly implemented using a resistor divider called an R-DAC, divides the signal range to 2^N regions as shown in Fig. 1.14. Each region corresponds to one of the codes in the output digital code space. The R-DAC generates the reference quantities against which the input signal is compared. One or more pre-amplifier stages subtract the reference quantity from the input quantity and the difference is amplified and is further quantized using a regenerative comparator/latch. The signal is quantized is using (2^N-1) comparators. The output of this comparator array is a thermometer code that is converted to a binary number using a thermometer to binary decoder.

The flash ADC is a brute force approach to signal quantization. The input referred offset of the comparators and inaccuracies in the R-DAC output values are the prime sources of error in the quantization process. Offsets are due to inaccuracies in nominally matched devices used in the circuits. The flash has the lowest latency of all

ADCs since all the decisions are made in one clock cycle. The highest clock speed of operation is determined by the regeneration time constant of the latch. A major limiting factor in the design of flash ADCs is that the hardware grows exponentially with the number of bits of resolution. Practical flash ADCs typically have 6 - 8 bits of resolution. It is possible to improve the resolution without dramatically increasing the size of the devices by adding trimming and offset auto-zeroing circuits. These circuits add extra analog complexity and are employed in the most demanding applications where other architectures are not feasible [15].

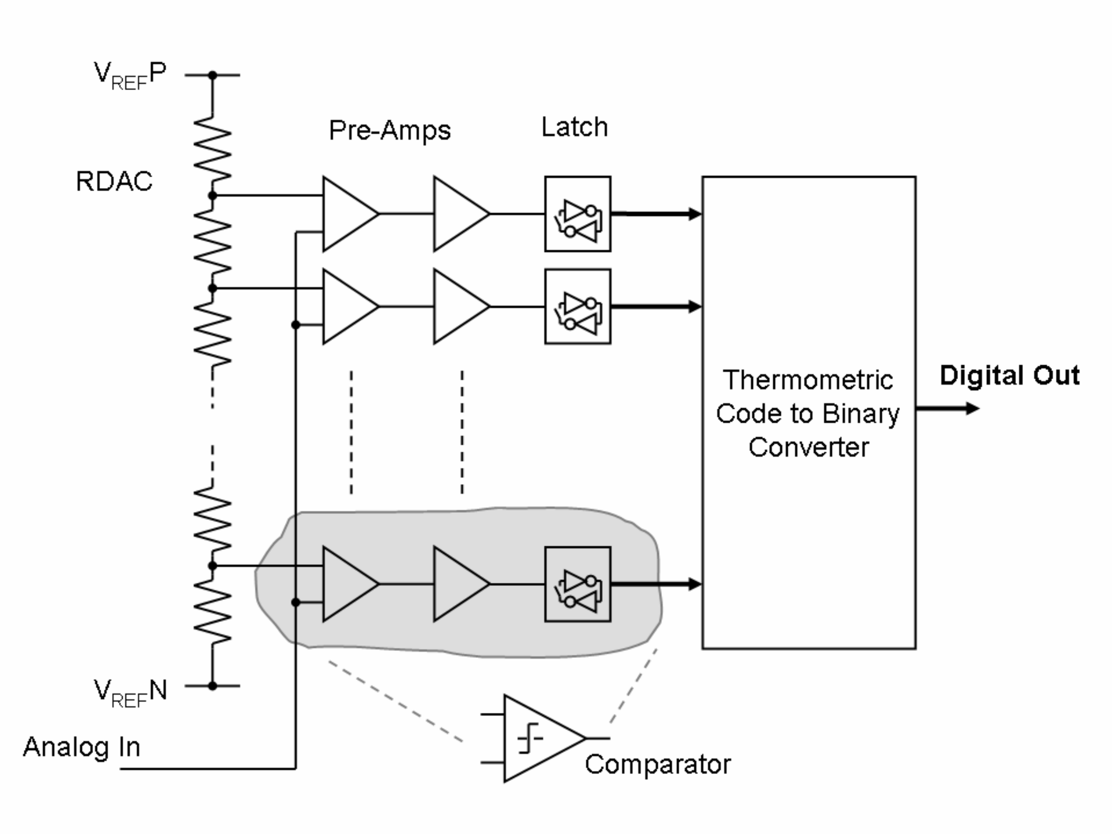


Figure 1.14: Flash ADC architecture.

1.5.2: Two/Multi-step flash ADC and pipelining

In a two-step flash ADC, a coarse quantizer is used to select the sub-range in which the signal of interest is present. The sub-DAC converts the quantized portion back to the analog domain and this value is subtracted from the signal to generate the residue or the un-quantized signal. The residue is amplified to occupy the full-scale range of the signal, which is then quantized by a second quantizer.

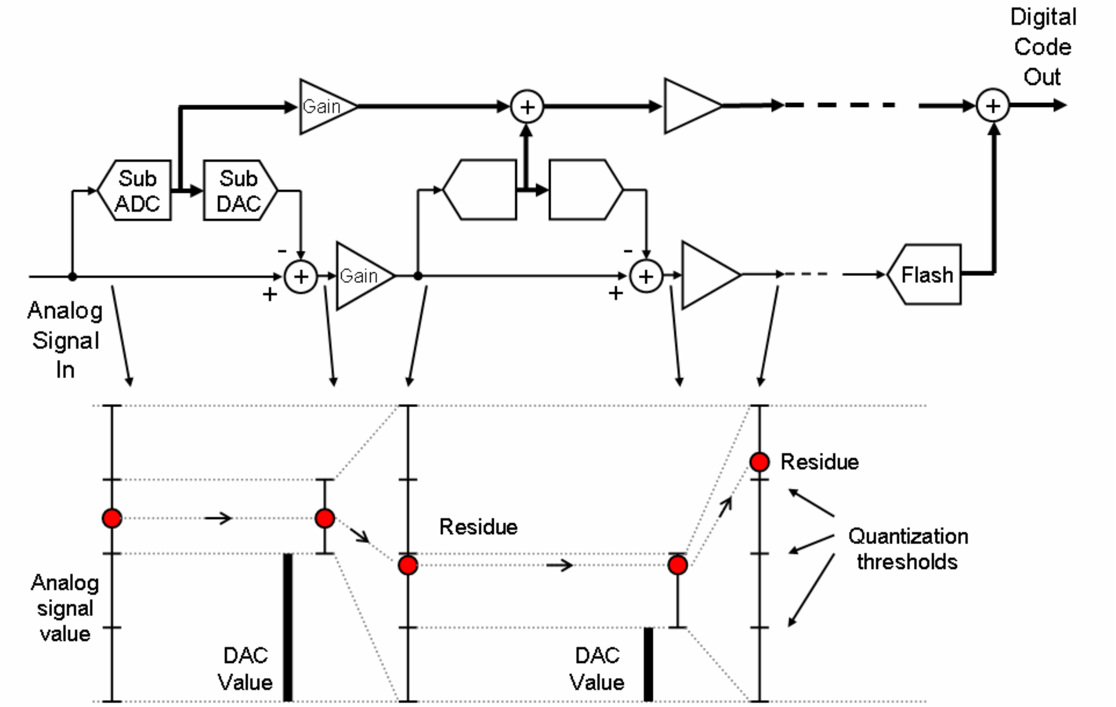


Figure 1.15: Multi-step flash.

This architecture saves on the number of comparators required though does not reduce the accuracy requirements of the comparators in the coarse quantizer with respect to that in an N-bit flash. The accuracy of the second quantizer is relaxed by the gain of the residue amplifier. An N-bit ADC with M bits resolved in the coarse quantizer requires ($2^M+2^{N-M}-2$) comparators. Mismatch errors in the sub-DAC and inaccurate gain in the residue amplification limit the performance of this ADC architecture. This concept can be extended to more steps of quantization to form a multi-step flash as shown in Fig. 1.15.

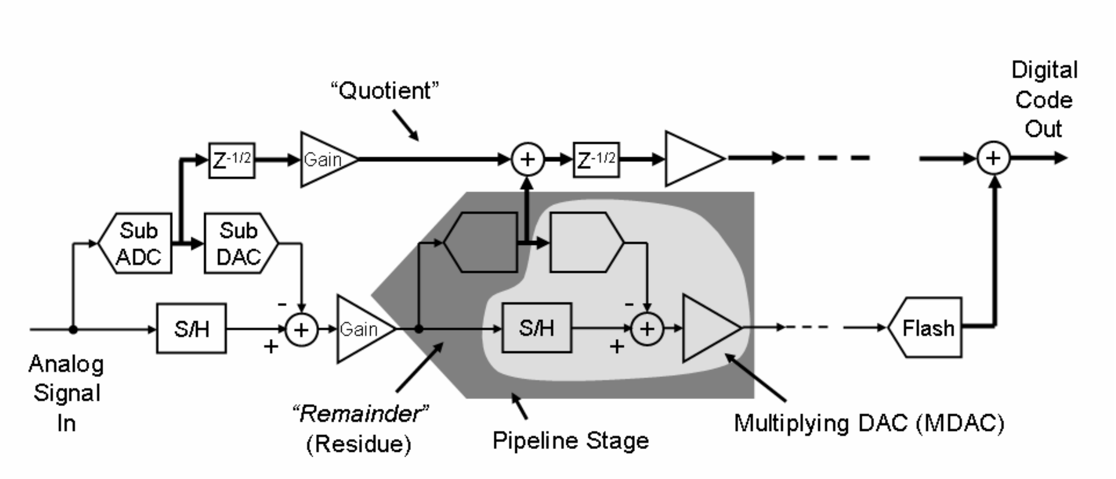


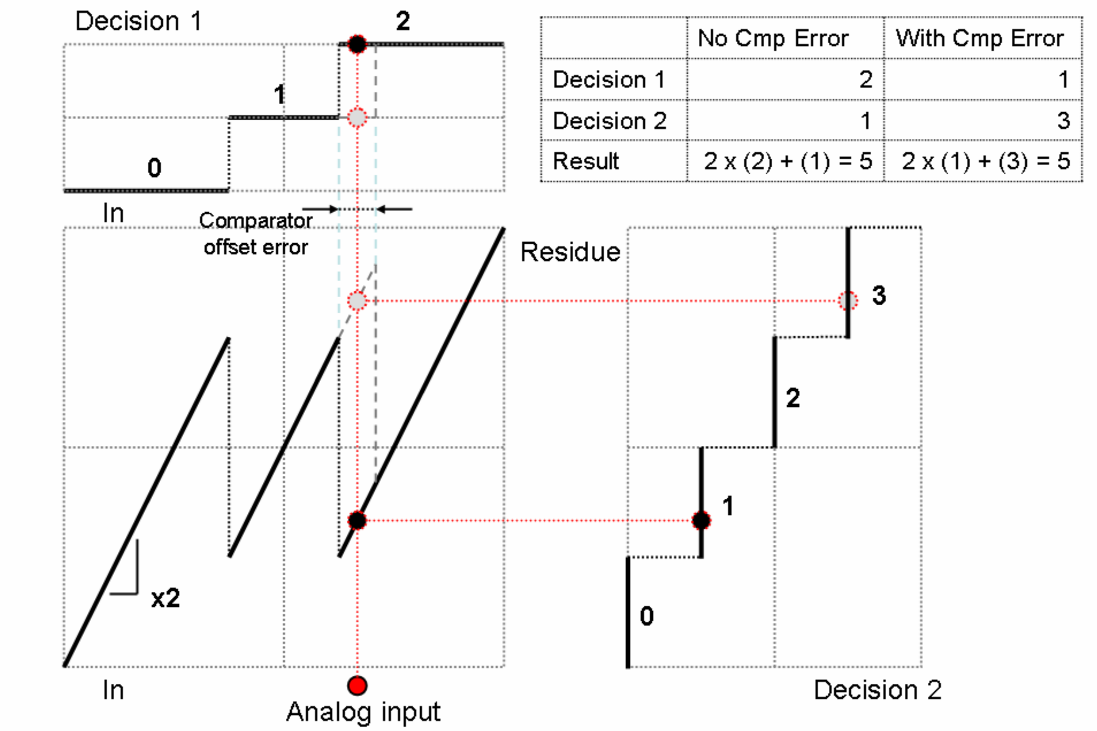
Figure 1.16: Multi-step ADC with pipelining.

The time available for the second quantizer is lesser than the clock period by the time required by the coarse quantizer to make its decision, the sub-DAC to switch and the output of the residue amplifier to settle to the desired value. The introduction of a sample and hold between the input signal and the residue amplifier, called pipelining as shown in Fig. 1.16, permits subsequent conversions to utilize a full half clock cycle. Pipelining is a general technique that has been applied to multi-step flash ADCs and even multi-step folding ADCs [16].

The quantization procedure in a pipelined flash ADC is similar to the process of long division. Each sub-ADC produces a part of the final result similar to the quotient and the un-quantized portion or the residue is quantized by the next step of the pipeline similar to the remainder. The sub-DAC and the residue amplifier accuracy limit the accuracy of the overall conversion. A switched capacitor circuit called the multiplying DAC (MDAC) [17] is used to perform the sample and hold, the DAC, the subtraction and the gain multiplication functions. The throughput of this ADC is one output sample per clock cycle however the pipelining introduces a fixed latency that may or may not be important in the system being designed. As an example, the latency due to the ADC is important in control systems with the ADC in the control loop since this increases the complexity of the control while on the other hand the latency due to the ADC is not important for most communication systems.

1.5.3: Pipelined ADC with redundancy

Adding redundancy to each sub-conversion step allows each sub-ADC to make larger conversion errors that, to the extent of the amount of redundant information, are corrected by subsequent conversion cycles. This is similar to the residual signed division (RSD) algorithm [18]. In such an ADC system, each sub-ADC resolves more bits than necessary. Consider the case of a pipelined stage with a gain of two that would have otherwise resolved one bit. Let the sub-ADC resolve more than one bit – say using two comparators in this case instead of one as shown in Fig. 1.17. The additional bit resolved allows each of the comparators to make errors, which are corrected as shown in the example. Such a sub-ADC is called a 1.5-bit sub-ADC [19]. The MDAC structure commonly used in pipelined ADCs is shown in Fig. 1.18. An ADC with pipelined stages and redundancy in the sub-ADC is commonly called a pipelined ADC.



	No Cmp Error	With Cmp Error
Decision 1	2	1
Decision 2	1	3
Result	2 x (2) + (1) = 5	2 x (1) + (3) = 5

Figure 1.17: Redundancy in a pipelined ADC.

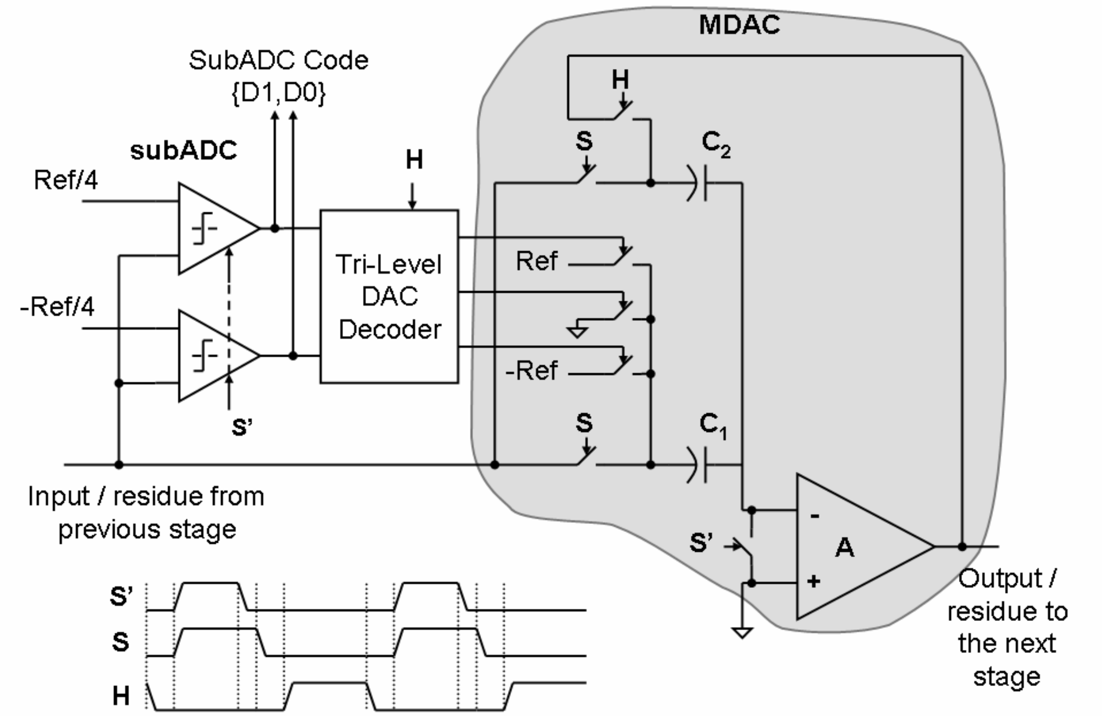


Figure 1.18: MDAC of a 1.5 bit per stage pipelined ADC.

1.5.4: Delta-sigma ADC

Delta-sigma ADCs employ oversampling and noise shaping techniques to quantize signals with a very high dynamic range and linearity (Fig. 1.19). This ADC architecture is very popular for high-resolution conversion at a low effective conversion bandwidth. The frequency band of interest is selected at the baseband or around a center-frequency of choice and this can be quantized using the low pass and the band pass architectures respectively. The feedback loop shapes the quantization noise due to a coarse quantizer outside the band of interest through the choice of an appropriate quantization noise shaping transfer function determined by a loop filter. The coarse quantizer operates at a much higher clock rate than the effective Nyquist sampling rate. The ratio of the actual sampling frequency to the effective Nyquist sampling frequency is the oversampling ratio (OSR).

The loop filter driving the quantizer tries to predict the incoming sample value based on the values of the past input samples and the error feedback loop serves as a corrector. This prediction-correction process modifies the spectral shape of the quantization noise. Digital filters and decimators at the output of the quantizer help reduce the noise and constrain the digital samples to the band of choice. The sampling rate of the comparators for a chosen signal-to-noise ratio is limited by the regeneration time constant, which is set by the choice of the semiconductor process technology. In other words, this also limits the choice of the oversampling ratio. Eqn. 1.13 shows the relationship of the output $Y(s)$, with respect to the input $X(s)$ in terms of Laplace transforms of the output and input signals. The loop filter transfer function $H(s)$ is transformed by the feedback to form the signal and the noise transfer functions (STF and NTF) that act on the input signal and the quantization noise due to the coarse quantizer $E_Q(s)$. Fig. 1.19 shows how the loop filter transfer function H(s) is modified to a low pass STF and a high pass NTF with a deep notch in the NTF to provide additional noise attenuation. Note that the error injected by the DAC $E_D(s)$ possibly due to dynamic effects or due to unit DAC element mismatches do not get noise shaped and is indistinguishable from the desired signal.

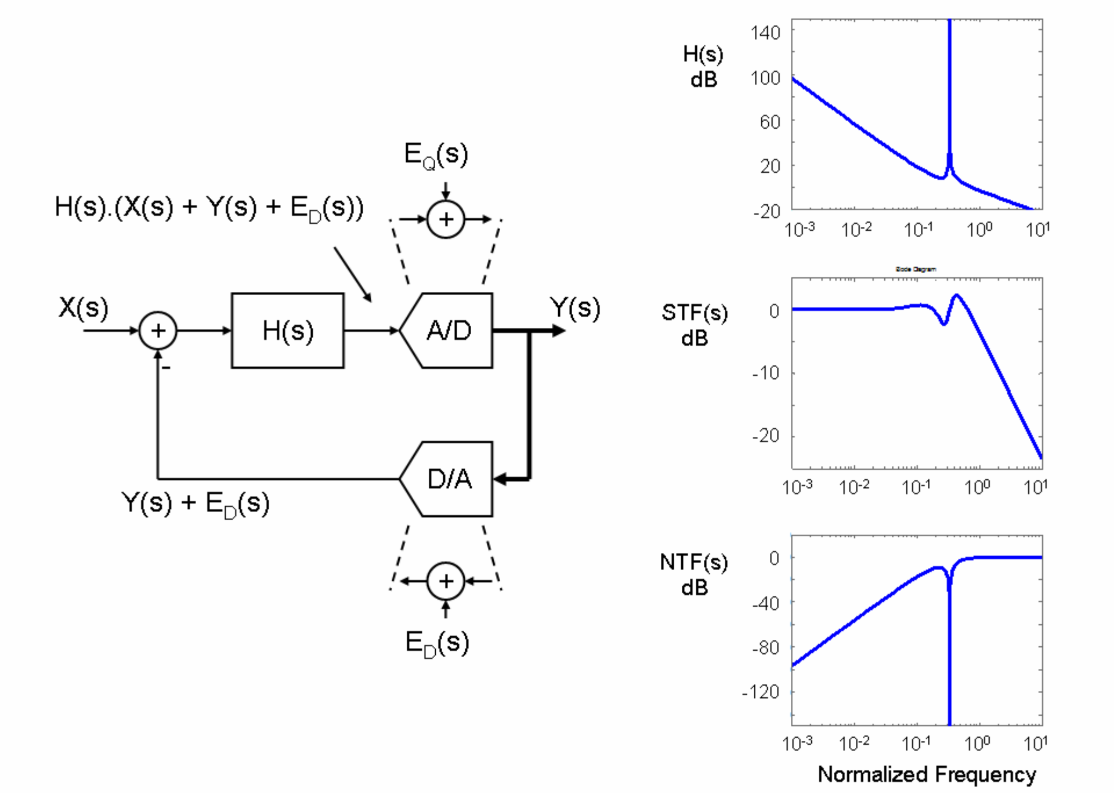


Figure 1.19: Delta-sigma ADC.

$$Y(s) = \frac{H(s)}{1+H(s)} X(s) + \frac{H(s)}{1+H(s)} E_D(s) + \frac{1}{1+H(s)} E_Q(s) \,. \qquad \ldots \text{ Eqn. 1.13}$$

The delta-sigma ADC architecture with a single-bit quantizer is extremely popular since the one element feedback DAC is inherently linear. This architecture is difficult to stabilize with tones present at the output during idle periods, which are the consequence of limit cycles and chaotic behavior [20]. The design has to be carefully evaluated in order to insure that the quantizer is not overloaded since the gain in a single-bit quantizer is not well-defined [21]. The theory supporting the single-bit quantizer architecture is not rigorous with several approximations made and these are typically designed with very conservative design margins. The choice of the oversampling ratio from technology constraints limits the choice of achievable dynamic range. Increasing the order of the loop filter can improve the SNR by an additional 6 dB but higher order loops become progressively difficult to design and stabilize [22]. The other option that is steadily gaining popularity is the use of a multi-bit quantizer because the SNR improves by 6 dB for every bit added to the coarse quantizer. The key advantage in using a multi-bit quantizer is that the quantizer gain is well defined thus easing the stability analysis of such systems. Delta-sigma ADCs based on multi-bit quantizers show more immunity to limit-cycles and idle tones. Errors due to inaccurate comparator decision thresholds in the multi-bit ADC is reduced because these are contained within the feedback loop. These architectures also allow a higher dynamic range of the analog circuits in the loop filter as compared to the single-bit quantizer architecture because the quantizer has more immunity to overloading [23]. The most significant limitation is that this design requires a multi-bit DAC in the feedback path. The mismatch accuracy of the DAC elements set the limit on the accuracy of the conversion since these errors are not suppressed by the feedback factor and these effects have to be mitigated in such architectures.

Chapter 2:

Techniques used to improve the performance of ADCs

A review of the techniques used to improve the accuracy of data converters is presented with emphasis on the error sources and correction / compensation methods in pipelined ADCs.

2.1: DISTINCTION BETWEEN ACCURACY AND RESOLUTION OF AN ADC

The process of quantization in an ADC poses an inherent limit to the accuracy of representing an analog quantity with a digital word. An ADC with N bits of resolution divides the finite signal range to 2^N regions and this sets the size of the quantization error to 2^{-N} with respect to the input range. Increasing the resolution of the conversion can reduce the quantization error to values acceptable to the system being designed by dividing the signal range to more regions. Errors in the quantization step size of a data converter can be attributed to lithographic inaccuracies (mismatches) and the physics of currently available deep-submicron processes (low intrinsic gain, device non-linearity). These errors affect the conversion process and often limit to the achievable accuracy of a data converter.

There is a fundamental difference between the resolution and the accuracy of a data converter though these terms are often used interchangeably. The resolution of a data converter relates to the number of states that a data converter can have or in other words the number of bits entering a DAC or exiting an ADC. The accuracy, on the other hand, refers to the precision with which each of these states map to the corresponding analog ranges measured in units of LSBs or in terms of bits. A 6-bit ADC has a resolution of 6 bits and an increase in the code in an ideal 6-bit ADC by an LSB corresponds to an increment of 1/64th of the analog signal range. If the increment is 1/50th of the signal range because of an error, this would correspond to an LSB of an ADC with 50 ranges which would correspond to $\log_2(50) = 5.644$ bits. This lack of precision is quantified by saying the ADC has 6 bits of resolution but accuracy of an LSB of a 5.644-bit ADC or in other words accuracy of 5.644 bits. It is trivial to show that an ADC or a DAC can have less accuracy than the resolution due to the presence of such errors. However, an ADC or a DAC can also have significantly higher accuracy than the resolution. Consider the case of a 2-bit DAC similar to that used in a two-step ADC implemented using four matched unit elements. If the relative mismatch error between any two elements is lesser than an LSB of a 16-bit data converter, the DAC can be considered 16 bit accurate. This terminology is frequently used in data converter architectures like the stage sub-DAC of a pipelined ADC where the typical resolution of the sub-DAC may be 1 to 3 bits but the accuracy may be equal to the accuracy of the overall pipelined ADC typically 8 to 16 bits.

Several systems and applications require the use of high-resolution data converters. Digital communication receivers are traditionally designed using 6 to 10-bit ADCs with adequate front-end analog signal conditioning [24], [25], [26]. Trends in development of semiconductor process technology has allowed the design of sophisticated digital filtering and signal processing algorithms to be implemented more economically than the designs using analog signal conditioning. As a result, it is often preferable to move the ADC closer to the physical transmission medium interface [27] and perform more signal conditioning tasks in the digital domain. These architectures require ADCs with high resolution (10 to 16 bit) and high sampling rates (20 to 100 MHz) [28]. It is interesting to note that continuous-time delta-sigma ADCs are gaining popularity in the 10 – 20MHz bandwidth at the 12 – 14 bit resolution level [29], [30]. ADCs with 8 to 10 bits of accuracy usually do not need additional correction since the circuit design techniques are mature enough to handle these accuracy levels. Though it is now an acceptable practice to use simple and inaccurate but power efficient high-speed analog circuits with associated calibration and compensation to undo the effects of the associated inaccuracies in some high speed ADCs [31], [32]. In order to achieve accuracies of the order of 12 to 16 bits, traditional circuit design techniques are often complemented with some form of compensation or calibration system in order to make the design feasible from a power and area perspective.

2.2: METHODS USED TO IMPROVE THE ACCURACY OF A DATA CONVERTER

Various techniques have been proposed in literature to improve the overall accuracy of a data converter. Static techniques make no prior assumptions regarding the frequency and time behavior of the signal during normal operation mode. The effect of the errors is measured and this information is used to update the state of the calibration in a way to reduce or compensate for these effects. The calibration state is held steady (or modified very slowly) during acquisition cycles. Dynamic techniques, on the other hand, transform the system in a time or frequency dependent manner often on a per acquisition cycle level in a way such that the effects of the errors do not affect the system output of interest [33], [34]. This is especially advantageous in systems where the information per-sample is not as important as the aggregate information observed over several samples e.g. oversampled ADCs. Each technique has their characteristic advantages and disadvantages and the use of different techniques strongly depend on the architecture of the data converter. This chapter presents a short summary of existing calibration techniques and specifically focuses on the calibration of pipelined ADCs. The pipelined ADC architecture is suitable for high-resolution conversion at high speeds and is popular in both industry and academia, hence is used as an example system to understand the implementation and issues in various calibration techniques.

2.2.1: Errors in pipelined ADCs

The first step in improving the accuracy of a data converter is to know that an error has occurred and to measure it with sufficient precision. These errors usually have a random part that depends on the mismatches between unit elements and a systematic part that depends on errors transformed by the architecture of the data converter. It is, therefore, necessary to understand the nature of these errors in the converted digital output and devise ways to correct them. The correction techniques mitigate the effects of the errors by mathematically manipulating the converted digital word or by correcting the source of the error in the analog domain.

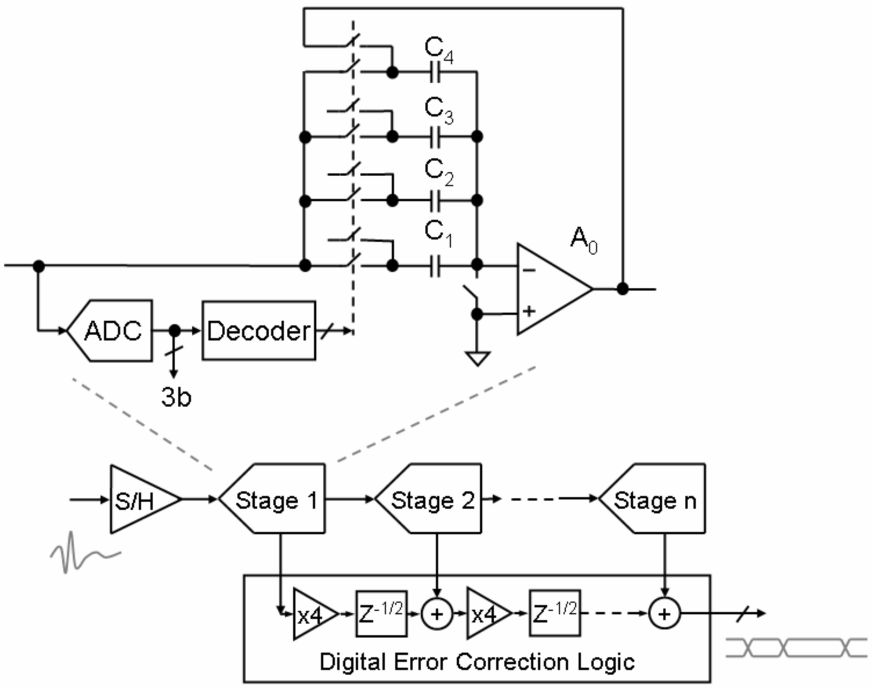


Figure 2.1: Pipelined ADC MDAC with two effective bits per stage.

It is known that errors in each sub-conversion step in a pipelined ADC can be easily corrected by adding redundancy to the conversion process [19]. The errors in the MDAC usually limit the performance of a pipelined ADC. Fig. 2.1 shows the architecture of a pipelined ADC with an MDAC stage resolving two effective bits with redundancy using six comparators in the sub-DAC (2.8 bits per stage). The sub-DAC, that converts the output of the sub-ADC back to analog levels, has to be as precise as the desired accuracy of the conversion. The step size of the sub-DAC is determined by an array of matched elements and a mismatch in the array is often the dominant source of error in pipelined ADCs.

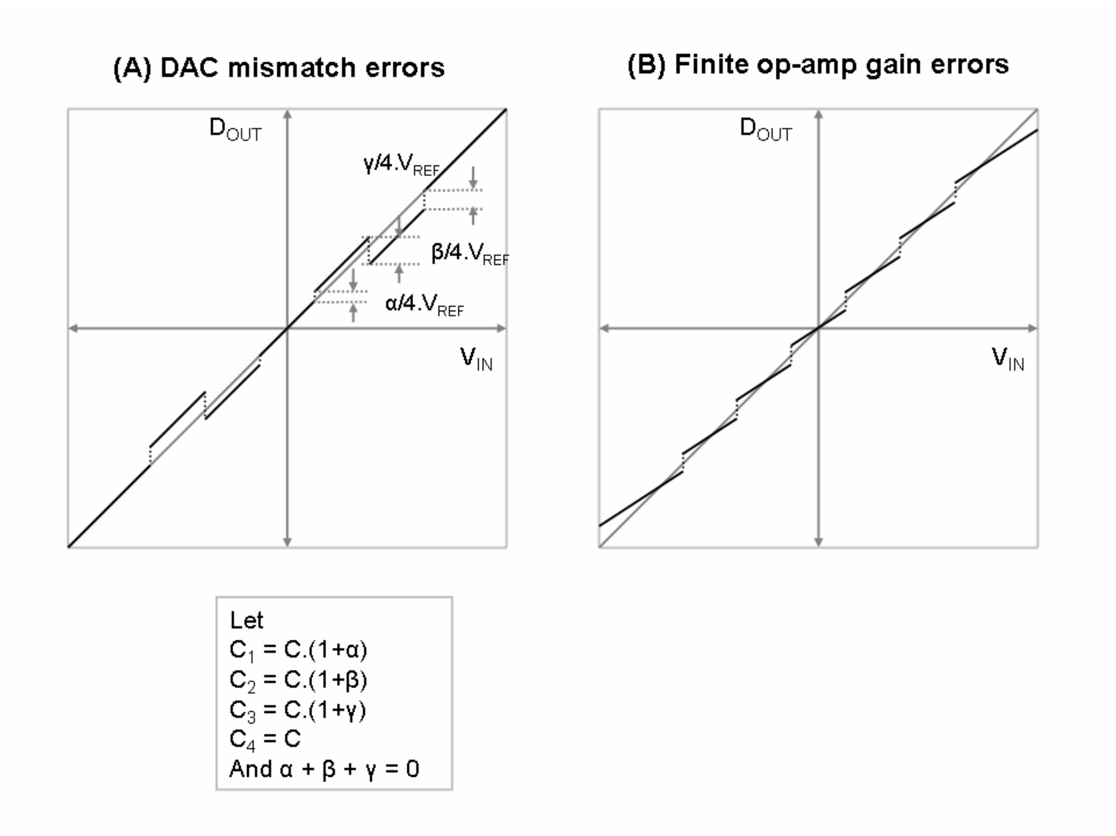


Figure 2.2: Effects of unit element mismatch and finite op-amp gain error.

This mismatch error affects the linearity of an ADC as seen in Fig. 2.2a. The mismatch error in the DAC elements is observed as the displacement of the segments in the transfer characteristic by an amount equal to the capacitor mismatch divided by the inter-stage gain. Capacitors are usually the most accurate components that can be fabricated in a standard CMOS process and are used to implement a switched-capacitor multiplying DAC (MDAC) [17]. Modern CMOS processes offer M-I-M (Metal-Insulator-Metal) capacitors that match to about 0.1%, for 1pF of capacitance, which is a typical choice from a thermal noise requirement. This limits the achievable accuracy of Nyquist-rate pipelined ADCs to 10 to 12b. The matching inaccuracy is usually due to limitations of lithography and subsequent processing steps [35]. The operational amplifier (op-amp) used in switched capacitor MDACs has to provide a precise analog gain that matches with the digital gain used during the reconstruction process as described in Fig. 1.11. Eqn. 2.1 is used to compute the open loop gain A_0 required from the operational amplifier in order to design an MDAC that resolves n bits in an N bit accurate ADC to achieve a worst case static gain error of ¼th LSB at the output. Here C_P is the parasitic capacitance at the input of the operational amplifier and C_{Stg} is the total sampling capacitor of the MDAC stage.

$$A_0 = 2^n\left(1+\frac{C_P}{C_{Stg}}\right)\cdot\left(2^{N-n+2}-1\right)\approx 2^{N+2}\left(1+\frac{C_P}{C_{Stg}}\right). \qquad \text{... Eqn.2.1}$$

This is interpreted by Fig. 2.2b which shows that the segments in the transfer function are rotated by an amount that depends on the finite operational amplifier gain. The discontinuities in the segments become significant to cause code errors when the amplifier gain is not large enough thus causing missing codes. Other errors in pipelined ADCs stem from the fact that the inter-stage gain is not perfectly linear and this signal dependant gain can cause distortion in the transfer function. The finite operational amplifier bandwidth causes an error in the settling of the residue amplifier. These errors are currently mitigated by designing circuits that meet the linearity and bandwidth specifications though there have been recent publications that demonstrate that the effect of these errors can be reduced through calibration [36], [37], [38].

Most published calibration techniques correct for static errors i.e. the errors are assumed constant for different frequencies and the corrective actions are independent of the input frequency. While this is a good assumption for most ADCs, it is possible to save significant design effort and power and area if the circuits are permitted to have frequency dependent errors. The work by Hummels et. al. [39] attempts to correct dynamic errors in an 8-bit ADC using a "phase-plane" compensation technique by characterizing the errors of the ADC with respect to the input signal and the derivative of the signal [40]. Such techniques use significant computations to perform the necessary corrections and as of this time are not worth the effort and are not efficient to implement in hardware. However, it must be noted that such algorithms may be practically implemented on chip with advances in process technology where the digital circuits exhibit superior performance in a dramatically smaller area. Such process advances are usually accompanied by poor analog performance of the transistors and a study of these methods might be essential for high-performance analog design in future process technology nodes.

2.2.2: Factory calibration – laser trimming

Laser trimming is a post-silicon correction technique that has been in use for a long time in the design of high-precision data converters [41], [42]. The device being calibrated is stimulated by using a well-known input signal like an accurate ramp or a sinusoid and the response is measured and fitted to a model of the system in order to determine the source and the magnitude of the various error terms. Circuits that are prone to errors, such as matched unit elements, are laid out using special patterns to form a trimming array that can be selectively burnt using a laser beam such that the matching accuracy is improved. This step requires specialized equipment and requires that the die be not fully packaged and on a test setup. It is possible that the chip characteristics could change after completing the packaging steps. The laser could heat and possibly damage circuits close the trim array. Though this technique is not common for modern data converters, it has been used recently for some experimental chips [43], [44]. Laser trimming is a destructive technique and cannot be reversed or re-done after the first time calibration. This is not very convenient and has since been replaced by several self-calibration techniques.

2.2.3: Architecture specific compensation techniques

There are several techniques reported in literature that use architecture specific features in a way to reduce or minimize the effects of errors. In [45], [46], the authors describe a capacitor error averaging technique to reduce the errors due to capacitor mismatch in a 1.5 bit per stage pipelined ADC. This uses a modified clocking scheme with three phases with the added phase used to average the mismatch error in the two capacitors. Yu [47] describes a commutative feedback capacitor switching technique for a 1.5 bit per stage pipelined ADC in which the DAC and the feedback capacitor are swapped based on the sub-ADC decision. This technique does not require a modified clocking scheme and improves the DNL; however, the INL is not improved as in the capacitor error-averaging scheme. Both these techniques work for 1.5 bit per stage pipelined ADCs. These are volatile techniques i.e. the errors are corrected during the operation of the data converter unlike non-volatile self-calibration techniques in which the calibration state is stored in a memory and the correction is applied in either the analog or the digital domain.

2.2.4: Self-calibration techniques

In self-calibration of an ADC, the errors are detected and subsequently corrected or compensated by the ADC system itself without special additional equipment to perform the correction. Often, during prototyping or for ease of debug during test, the data is captured and processed by an FPGA or a computer program to perform the correction. Though assisted by external equipment, it is feasible to implement these algorithms in the ADC hardware itself so these methods are also called self-calibration. Calibration techniques either measure the deviation of the ADC transfer function from the ideal transfer function i.e. the INL or the relative size of quantization steps i.e. the DNL. The correction can be implemented either by correcting the INL by storing information about the code step or by correcting the DNL by storing information about the relative values of the codes or ratio of components in the calibration state.

Digital calibration techniques have been successfully demonstrated for MDAC-based pipelined ADCs initially with foreground error measurements [48], [49], and recently embedding error measurement cycles in background [31], [36], [50]-[54]. These techniques are becoming popular due to the scaling advantages in modern fine geometry CMOS processes, which allow sophisticated digital algorithms to be incorporated in the same die as the ADC. The cost of implementation reduces with advances in semiconductor process technology because digital circuits scale better than analog circuits. However, these algorithms are limited by finite word length truncation errors and the accuracy of measuring the errors themselves as a result require additional bits resolved in the data converter in order to minimize these measurement errors. Analog techniques, on the other hand, usually add circuit complexity but have the unique advantage of applying corrections right at the source of the error rather than compensating for them later [45], [55]-[58]. This may be an attractive feature in many systems. These techniques usually employ resistor or capacitor trimming networks to correct the mismatch between nominally like elements.

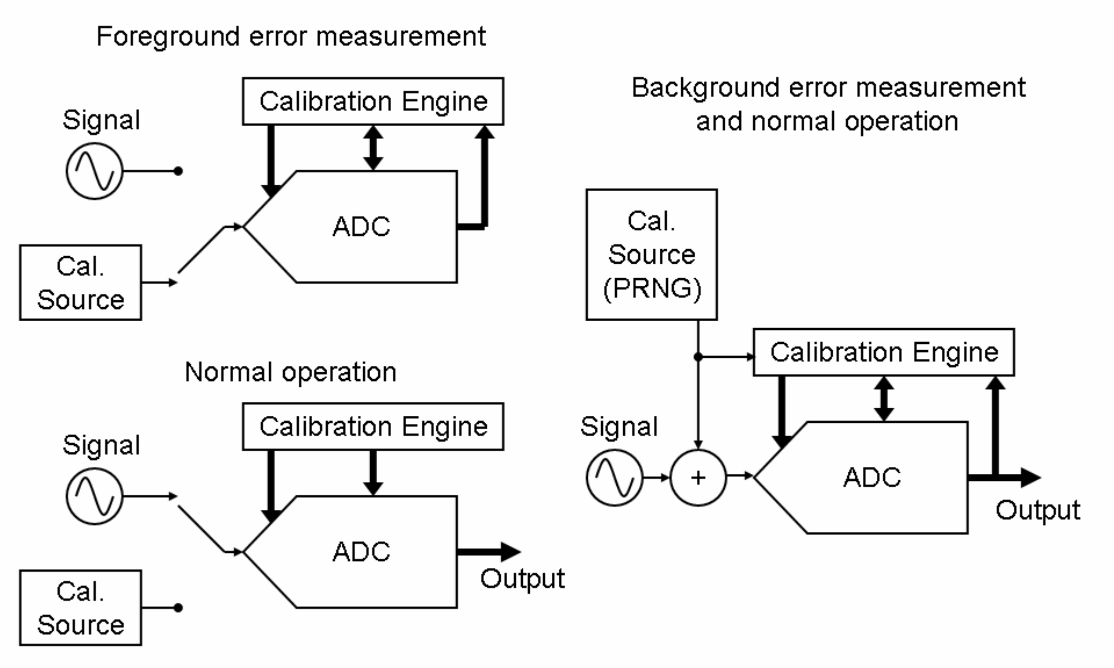


Figure 2.3: Foreground and background calibration.

Foreground or offline calibration is conceptually the simplest calibration technique. In this technique, the ADC is configured in a calibration mode in which the signal of interest is not connected to the input of the ADC i.e. the ADC is taken “offline” as shown in Fig. 2.3. The ADC is then connected either to an accurate calibration signal source like a linear ramp or a pure sinusoid or various circuits of the ADC are configured differently and connected to stable reference values. The response of the circuit to these known stimuli is measured and compared against expected ideal values to obtain an indication of the error. Further corrective steps can be taken to correct these errors and often the calibration procedure is repeated to measure residual errors. Once calibration is performed, the ADC is reverted to the “online” mode in which the signal is connected to the input and normal conversion is resumed, now with the errors corrected or compensated. These schemes are usually simple to implement but calibration needs to be repeated if the errors change due to temperature changes or due to aging. Foreground calibration, thus, is more suitable when errors due to ratio of like components are calibrated since these do not change much over environment changes e.g. capacitor mismatches in a switched capacitor MDAC. Since there is no metric available to insure that the calibration is still good, the ADC needs to be periodically taken offline and characterized/calibrated. This might not be acceptable in several systems. Foreground calibration has the advantage that only signals derived from the calibration sources are present in the circuit (along with thermal noise) and the error and the detection system can quickly determine the amount of error.

Background calibration is a technique that allows an ADC to be calibrated while remaining “online” i.e. while converting signals of interest by embedding error measurement cycles with normal conversion cycles. Measurement and update cycles are applied iteratively until the system converges to a state with the errors minimized. Several schemes have been proposed but the common theme is that this technique allows the accuracy of the ADC to be monitored continuously and the correction or the compensation tracks with the changes in errors due to changes in the environment. During the initial start-up or during tracking, the performance of the ADC is lower than the specifications but improves and stabilizes to a steady state value as calibration is performed. Background calibration schemes are broadly classified as either deterministic background calibration or stochastic/randomized background calibration. Deterministic background calibration algorithms hide the error measurement cycles or the error information in a deterministic fashion. The limitation of this technique is that during the calibration process, the residual errors may result in visible patterns in time or frequency domain (spurs). Stochastic or randomized background calibration algorithms hide the error measurement cycles or the error information in a random fashion. During the calibration process, the residual errors appear similar to noise and degrade the SNR but rarely exhibit visible patterns in time or frequency domain (spurs). The randomization is performed using pseudo-random sequences. Background calibration schemes are typically more complicated to implement than foreground calibration schemes. Since the error measurement cycles are multiplexed with the normal conversion cycles, the initial convergence of the calibration system to the correct values is often slow.

Based on the method used for measuring the errors (foreground or background) and implementing the corrective action (tweaking an analog component to correct the error or adjusting a digital coefficient to compensate the effect of the error), most calibration techniques can be classified as either “Analog Foreground Calibration”, “Digital Foreground Calibration”, “Analog Background Calibration” or “Digital Background Calibration”. A few key research publications that have made a significant impact are discussed in the following sections.

2.2.5: Analog foreground calibration

One of the earliest published works on analog calibration was that by Maio et. al. [59] in which a 14-bit DAC was calibrated offline using a very accurate ramp signal and a comparator. The correction terms were applied during the normal conversion mode using a trimming DAC with the amount of correction determined by correction terms computed and stored in the system memory. Generating the accurate ramp requires specialized test equipment as a result this solution is more suitable for factory calibration and is not practically implemented as a part of a general-purpose digitizer system.

A more practical self-calibration solution was introduced by the work due to Lee et. al. [60], [61], [62] and this forms the basis of several modern calibration schemes. The authors describe foreground measurement of errors in a SAR type ADC with capacitive charge redistribution architecture. The binary weighted main capacitor DAC array in calibration mode is excited in various configurations using the same fixed reference voltages that are used during the normal operation mode. The error information for each bit transition is stored in memory and this drives a resistor string based calibration DAC to compensate for mismatch errors during normal conversion cycles. A similar work was described by Goes et. al. [63] applied to the MDAC capacitor mismatches. This work also demonstrated the calibration of inter-stage gain errors.

The work by Manoli [64] used a dual slope method to measure the errors in the DAC used in a SAR ADC. This work stands out because this uses a high-accuracy but slow conversion procedure to calibrate an ADC that will ultimately operate at a higher conversion speed. The dual slope digitizing process uses time as the reference quantity and the accuracy and the resolution is limited by the clocks and can be implemented without precision components.

2.2.6: Digital foreground calibration

The use of digital correction to compensate for analog errors was described by Boyacigiller [65] in which a 14 bit accurate DAC is achieved through the use of a low accuracy non-binary radix 17 bit resistor DAC and a digital look up table implemented using an on-chip EPROM. The calibration procedure described in this is cumbersome and insight added by development of analog domain calibration techniques allowed a more efficient and simpler implementation of the digital domain calibration technique as described by Lee et. al. [48]. In this scheme, the MDAC of a two-step ADC is calibrated in the digital domain. The capacitor mismatch errors are measured in the foreground by progressively connecting these capacitors to the reference voltage and a digital word representing the size of each code step is stored in memory. This stored code step term is used during code reconstruction to compensate the conversion error in the digital domain. This had a very significant advantage of reducing analog complexity and eliminated the look-up function that was necessary to perform code specific analog correction. An extension to this technique for multi-stage calibration was proposed [66] to prorate the errors in multiple stages being calibrated thus preventing an accumulation of errors causing DNL problems. A similar work was presented by Lee [67] in which the calibration cycles resulted in code values near zero, i.e. the code error is measured instead of measuring the code steps, as a result the error measurement was not affected by the gain error of lesser significant stages.

The technique described by Karanicolas et. al. [49] demonstrated the calibration of a non-binary reduced radix pipelined ADC. In this, each stage of the pipeline resolves less than an effective bit of information. Such a pipeline has the advantage that the inter-stage gain and the DAC step size can be combined into one weight term called the radix of the number system. A similar technique is used by Poulton et. al. [32] to calibrate the slice of a massively interleaved ADC capable of digitizing at 20GS/s.

2.2.7: Analog background calibration

One of the earliest attempts to correct for mismatch errors in a DAC without taking the data converter offline was proposed by Groeneveld et. al. [68]. In this work, a spare unit current source replaces any unit DAC current source in a transparent manner so that this current source can be calibrated using a master current source and then returned to the main signal path. All unit elements are calibrated in a deterministic fashion. The actual calibration takes place offline and converges rapidly. This technique is still commonly used in several modern ADCs and DACs [69]-[73] and is particularly suitable for stand-alone DACs where it is difficult to apply other calibration techniques.

Jewett et. al. [74] describe a technique to reduce the DNL in a two-step flash by inserting a pseudo-random additive dither to linearize the discontinuities in the transfer function. The dither is removed digitally during code reconstruction using an estimate of the gain in the dither path and a correlator is used to detect residual errors due to incorrectly estimating the gain. The error drives a least mean square adaptive calibration loop (LMS) that adjusts the gain of the dither DAC to drive the residual dither to zero. An advantage of this scheme is that digital multipliers are avoided by implementing the variable gain function in analog domain, which was resource efficient in the IC process technology available in 1997. The LMS algorithm is popular in the calibration of data converters because of the simplicity of implementation [75].

In the scheme proposed by Ingino [56], a spare pipelined ADC stage replaces a stage being calibrated and the calibration is performed in the analog domain. This requires additional hardware but this permits the calibration to run continuously without interrupting normal conversion cycles.

In the scheme proposed by Choe et. al. [76], the offsets in a pipelined folding ADC is calibrated in the analog domain by measuring the errors in the background using an oversampled delta-sigma ADC. A unique feature of interpolation by resistor networks permit the removal of a folding amplifier for calibration while generating the missing zero crossing point without severely compromising the normal conversion process. This allows the folding amplifier to be calibrated offline while the remaining hardware remains online thus making the calibration procedure transparent to the system. Ryu et al. [57] describe a similar scheme to correct for the capacitor mismatches in a 1.5 bit per stage pipelined ADC by implementing the correction in the analog domain to make the stage capacitors equal using a capacitor trim array.

Nagaraj [77], [78] described a method to calibrate a pipelined ADC by trimming the reference voltage for each stage. This technique is suitable for a 1.5 bit per stage pipelined ADC. This work formed the basis of the work presented by Ming et. al. [31] in which the inter-stage gain in a pipelined ADC is measured by injecting a pseudo-random dither at the input of a stage in a pipelined ADC similar to [74]. This dither is quantized by both the ADC being calibrated and by a slow but accurate delta-sigma ADC and is removed digitally during code reconstruction. A correlator detects any residual dither and this drives an LMS calibration algorithm that in turn drives a DAC to adjust the stage reference to drive the residual dither to zero.

2.2.8: Digital background calibration

Shu et. al. [16] demonstrated a true digital background calibration of the resistor DAC in a multi-step ADC. The resistor mismatch errors are continuously measured in the background by measuring the differential tap voltages across each unit resistor using a subtraction circuit followed by a slow but high-resolution delta-sigma ADC. The code correction coefficients are stored in memory and are used to correct the converted word digitally. The gain errors are corrected using the gain proration technique as described in [66].

In [50], [79], a skip-and-fill technique was proposed in which desired samples are randomly discarded (skipped) and replaced by error measurement cycles. The "skipped" samples are reconstructed using non-linear interpolation based on a polynomial fitting function that "fills" the gaps. As expected, such a system cannot work until the Nyquist frequency and the frequency band is characterized for SNR degradation based on the number of samples thrown away. The segment errors in each stage are measured in a way similar to [48].

A theoretical technique [80] uses a band-limited signal that frees up a section of the Nyquist band to be used for calibration. The quantization noise in the calibration band is used as a measure of the non-idealities in the ADC and is used to extract error information.

Erdogan et. al. [81] and Blecker et. al. [82] describe an approach using a front-end queue that samples the input at a lower frequency than the sampling frequency of the algorithmic ADC being calibrated. This difference in the sampling rate and the conversion rate allows the ADC to have spare cycles during which the calibration signal can be introduced while bypassing the input queue. The actual calibration can be performed by any digital foreground calibration scheme [49], [61], [48].

Galton [34] described a DAC noise cancellation technique to scramble the mismatch errors in the MDAC of a pipelined ADC using pseudo-random bit sequences similar to the dynamic element matching technique used in oversampled ADCs. The mismatch errors of the DAC unit elements are modulated by these sequences and these errors can be detected digitally using correlation based techniques and can be removed digitally. The DAC and feedback-capacitor averaging scheme proposed by Yu et. al. [83] uses a similar scheme. Siragusa et. al. [52] implemented the DAC noise cancellation and a gain-error correction scheme similar to [74] to correct both the sub-DAC capacitor mismatch and the inter-stage gain errors in a pipelined ADC digitally in the background.

Chang [84] applied the radix based concept [49] to group the errors in a 1.5 bit per stage pipelined ADC in to one term. This was a key idea that permitted inter-stage gain errors and capacitor mismatch errors to be corrected using one digital coefficient for a 1.5 bit per stage pipelined ADC stage. Liu et. al. [54] demonstrated a background digital calibration scheme based on this idea. Shu et. al. [85] further demonstrated the use of signal dependant pseudo-random dither to improve the speed of calibration with a similar error correction scheme.

The output code density histogram of an ADC provides information about the INL [86] and this can be used to estimate the errors. Murmann [36] demonstrated that simple open-loop amplifiers could be used instead of power hungry operational amplifiers in a pipelined ADC stage by calibrating the gain and the static third order non-linearity error by estimating these errors using code density histograms and an LMS algorithm to update the correction coefficients digitally.

Wang et. al. [87] describe a nested digital calibration scheme in which a slow but accurate algorithmic ADC is used to calibrate a fast and inaccurate pipelined ADC in the background similar to [31]. The algorithmic ADC is not in the main signal path, can be calibrated offline using foreground calibration techniques, and is used to calibrate the main ADC.

Most of these techniques that have been described are statistical calibration techniques i.e. these use pseudo-random sequences to carry the error information with the desired signal being converted. The error estimator treats the signal as the undesired portion of the sample and estimating the error is similar to looking for a needle in a haystack. These methods generally converge slowly since the signal is usually larger than the error and show strong dependence on the statistics of the signal. Deterministic background calibration schemes, on the other hand, are free of these disadvantages. In the scheme proposed by McNeill et. al. [88], the ADC is split to two halves each of which are excited using the calibration signal with opposite polarities but both halves convert the same input signal. The output codes of the ADC when added together tend to cancel the calibration signal and only the desired signal remains. The signal tends to cancel out when these outputs are subtracted leaving behind only the applied calibration signal, which carries information regarding the magnitude of the errors. This enhances the speed of detecting the errors since the desired signal is no longer a part of the error measurement. A similar scheme is presented by Fan [89] with enhancements like a pseudo-random chopper to shape the statistics of the input thus improving the robustness of the calibration process.

2.3: CORRECTION METHODS DESCRIBED IN THIS DISSERTATION

This dissertation focuses on developing a framework for the analog calibration of multi-bit DACs. Previously published research on analog calibration in pipelined ADCs has focused mainly on the 1.5 bits per stage architecture, where such techniques can be easily employed. It is cumbersome to extend similar concepts to correct the mismatch errors in multi-bit MDACs. This is reflected in the relative paucity of published research in this domain. Pipeline ADCs that resolve more than a bit in each stage can be designed to be more power and area efficient than their single effective bit per stage counterparts. The first topic is the proposal of an alternative solution equivalent to analog trimming. The algorithm is a novel technique that uses the errors in the capacitors as the corrective elements themselves. This statistical matching scheme is a generic scheme and can be used to correct for the errors in an array of many nominally identical elements like the DAC in a delta-sigma ADC or current steering DACs. This algorithm is called self-configuration. The feasibility of the proposed algorithm is demonstrated through the implementation of a switched capacitor multi-bit per stage pipelined ADC in which the mismatch errors in a multi-bit MDAC are corrected by breaking each unit DAC capacitor into smaller elements and regrouping them so that the errors can be cancelled.

The second topic is a short study of an existing calibration technique in which a current steering DAC is trimmed using redundant hardware in the background. The main contribution to this work is simplification of the existing scheme to reduce hardware implementation complexity. The proposed enhancements permit synchronous retiring and insertion of unit DAC elements with a short latency in the signal path, which makes it feasible for this scheme to be used to calibrate the feedback DAC in a continuous-time delta-sigma ADC.

Chapter 3:

Capacitor self-configuration algorithm

A novel algorithm is presented to improve the matching of nominally alike elements. The benefits and performance limitations of this algorithm is explored. The use of this algorithm to improve the matching of the elements of a multi-bit DAC is proposed and is applied towards improving the performance of a pipelined ADC.

3.1: SELF-CONFIGURATION AS AN ALTERNATIVE TO TRIMMING

The accuracy of a DAC relies on the relative weights of the elements that translate the digital bits to analog quantities. The elements are nominally weighted in powers of two in a binary weighted DAC i.e. if the LSB drives a unit element, the bit that is more significant than the LSB drives an element that is twice the size of the unit element. All elements are nominally identical in a segmented DAC. Mismatches in these unit elements translate to a reduction of the overall accuracy of the DAC and the effect of these mismatches have to be corrected as discussed in Chapter 2. Digital techniques and analog trimming based techniques have been used to compensate or correct the effect of these mismatches. This chapter describes a novel technique that is proposed as an alternative to trimming based methods to improve the matching accuracy of unit elements. This generic technique can be used to correct mismatch errors in both switched capacitor and current steering multi-bit DACs used in various data converter architectures.

3.1.1: The governing principle

Combining *n* identical elements with standard deviation σ will cause the effective standard deviation to decrease by the square root of *n*. This is true as long as the combination is done at random without prior knowledge of the individual or relative values of the elements. However, knowledge of the relative values allows elements to be combined such that the grouped elements are better matched**.**

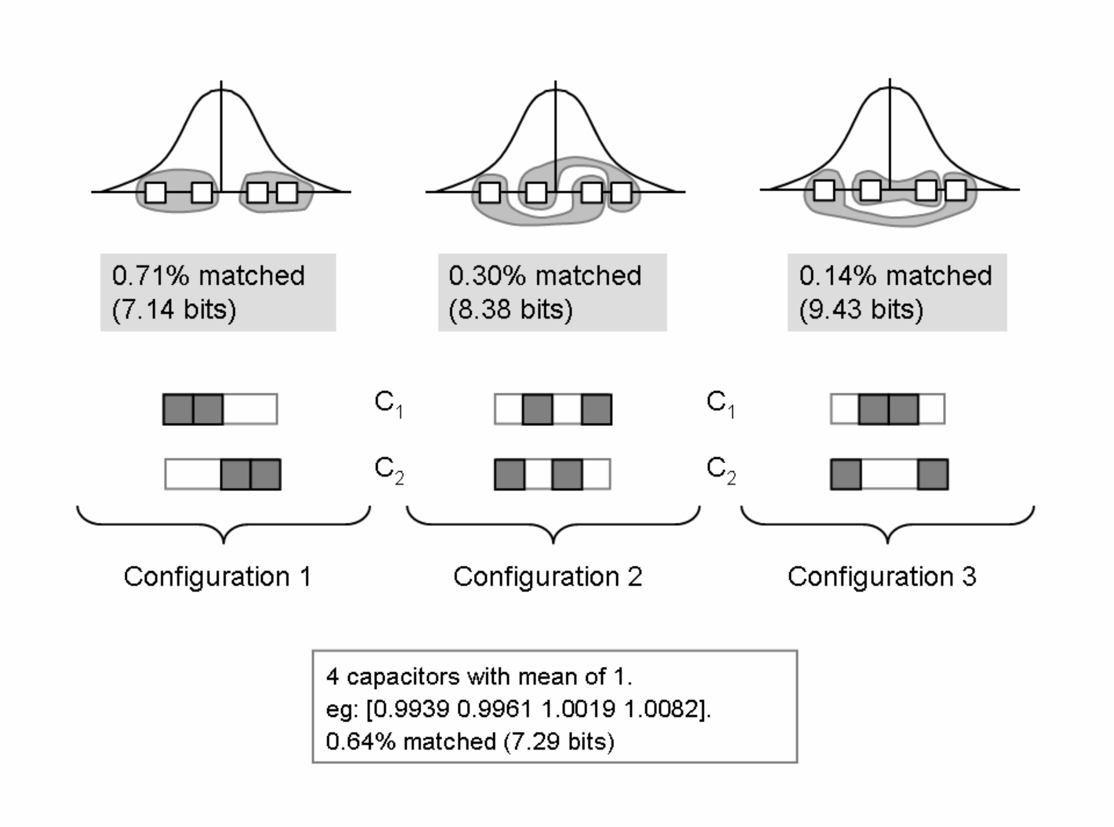


Figure 3.1: Example showing improvement in matching by configuring the sub-elements in different ways.

This can be explained by a simple numerical example shown in Fig. 3.1, where two nominally identical elements are made with two half ones each. The elements from a pool of four sub-elements can be recombined into two groups in three different ways each with a different spread of values. It is observed that the spread in component values in one of these configurations is smaller than the spread in the others. In other words, mismatch errors can be reduced by finding such a configuration in which a positive error cancels a negative error and the resulting combined capacitors have the least possible spread. It is convenient to express the mismatch of the capacitors in terms of bits by transforming the standard deviation of the error using $-log_2(\sigma)$ or $-log_2(3.\sigma)$ depending on whether the error is to be described as the standard deviation or the 3σ spread where σ is the standard deviation normalized with respect to the mean of the sub-elements.

It is interesting to note that a recent publication [90] demonstrated that the inverse of this principle could be used to generate bias currents with a vast dynamic range by generating very small bias currents as difference of nominally identical unit bias cells.

3.1.2: Improvement of DAC performance using this scheme

The technique of breaking each element into sub-elements and regrouping with the aim of finding a configuration with a lower spread can be used to design DACs with high accuracy like the MDAC in a high-resolution pipelined ADC. The procedure of combining the sub-elements in different configurations can be visualized as a process of permuting and grouping these sub-elements. This approach leads to a simple implementation of the algorithm. The same principle can be easily extended to form more matched elements using a larger set of sub-elements shown in Fig. 3.2.

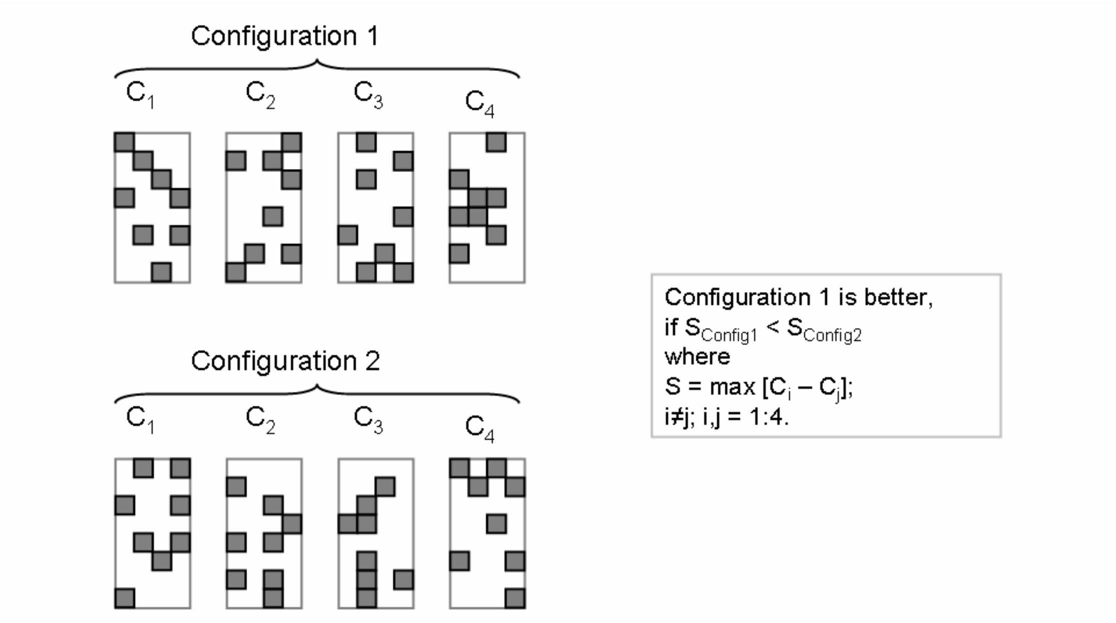


Figure 3.2: Extension of the self-configuration technique to larger arrays.

One of the objectives of this work is to find a way to design an accurate multi-bit MDAC in a high resolution/accuracy pipelined ADC. The case of a switched-capacitor MDAC with two effective bits per stage is used to demonstrate this technique. Four matched capacitors are needed in this case as will be described later. The matching accuracy for an array with more than two capacitor groups can be defined as the standard deviation of the capacitance of the grouped sub-elements normalized with respect to the mean capacitance of these capacitor groups. For a simple circuit level implementation, the spread between the smallest and the largest of the effective values of these sub-element groups for a chosen configuration is used as an indication of the matching accuracy of the array. Simulations using current mismatch data obtained from the semiconductor foundries show that 13 – 14b of matching accuracy can be achieved if each unit element is broken into eight parts to form a pool of 32 sub-elements as shown in Fig. 3.2. These sub-elements can be appropriately permuted and grouped in various ways and the spread is measured for each configuration in order to find the case with the best possible matching.

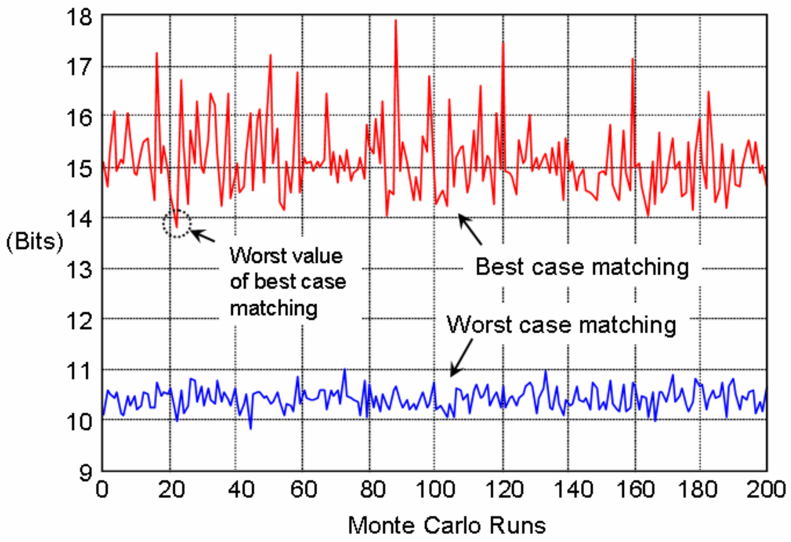


Figure 3.3: Simulated worst and best cases for 8 units per DAC capacitor and 4 DAC capacitors with 10 b initial matching after 500 permutation trials.

Fig. 3.3 shows the best and worst case results obtained from a simulation of the algorithm after 500 trials for the case of a DAC with 4 units each made using 8 sub-elements. The different Monte Carlo simulation trials represent instances of the DAC in which different initial starting capacitor values are randomly chosen from a 10 b σ matched distribution. For each Monte Carlo simulation, the best (smallest spread) and the worst-case values (largest spread) of the matching accuracy observed have been plotted. The algorithm chooses the condition for the best case matching as the solution. The worst value of the best case plot for a set of Monte Carlo trials is used as a measure of the lower bound of the best performance achieved by the search algorithm.

$$N = \frac{\prod_{i=0}^{D-1} \binom{(D-i)\cdot M}{M}}{D} \quad . \qquad \ldots \text{ Eqn.3.1}$$

Eqn. 3.1 expresses the number of unique possible configurations *N* in which *D* groups can be constructed from *M* sub-elements each.

Table 3.1: Number of possible unique configurations

Sub-Elements/Group Groups	2	4	8	16
2	3	35	6,435	3.01 x 10^{8}
4	630	1.57 x 10^{6}	2.48 x 10^{16}	1.65 x 10^{35}
8	1.02 x 10^{10}	2.98 x 10^{23}	2.27 x 10^{51}	1.31 x 10^{108}

This is also summarized in Table 3.1 for commonly used MDAC configurations (2, 4, 8 DAC capacitor groups in 1, 2 and 3 effective bits per stage MDACs) for different numbers of sub-elements in each DAC capacitor group (2, 4, 8 and 16). Note that the number of possible permutations increases rapidly as the number of sub-elements increase. A larger set of sub-elements allows more degrees of freedom in the permutation and grouping process. More combinations increase the possibility of improving the accuracy of the matched array.

Table 3.2 summarizes the simulated upper bound (UB), the average (Avg) and the lower bound (LB) of the best case matching (in 3σ bits) detected in a set of 100 point Monte Carlo experiments with 5000 permutation and grouping trials for each data point. Capacitor arrays with 0.1% (9.8 bit σ) were used as a starting point for each Monte Carlo experiment.

Table 3.2: Monte Carlo simulations for matching accuracy

Sub-Elements/Group Groups		2	4	8	16
2	UB	19.52	22.28	28.47	28.77
	Avg	10.22	13.95	21.16	22.00
	LB	8.53	11.37	18.27	19.94
4	UB	14.27	15.76	18.79	16.61
	Avg	10.53	13.52	14.12	14.60
	LB	9.38	11.93	13.09	13.69
8	UB	11.77	12.43	12.84	13.06
	Avg	10.58	11.34	11.90	12.37
	LB	9.62	10.82	11.39	11.95

This is also shown graphically in Fig. 3.4. This shows that the probability of finding opposite signs of error within a group increases as the number of sub-elements per group increases when a fixed number of groups have to be created. This is the reason why the upper bound of the best possible matching achievable is higher for more sub-elements per group.

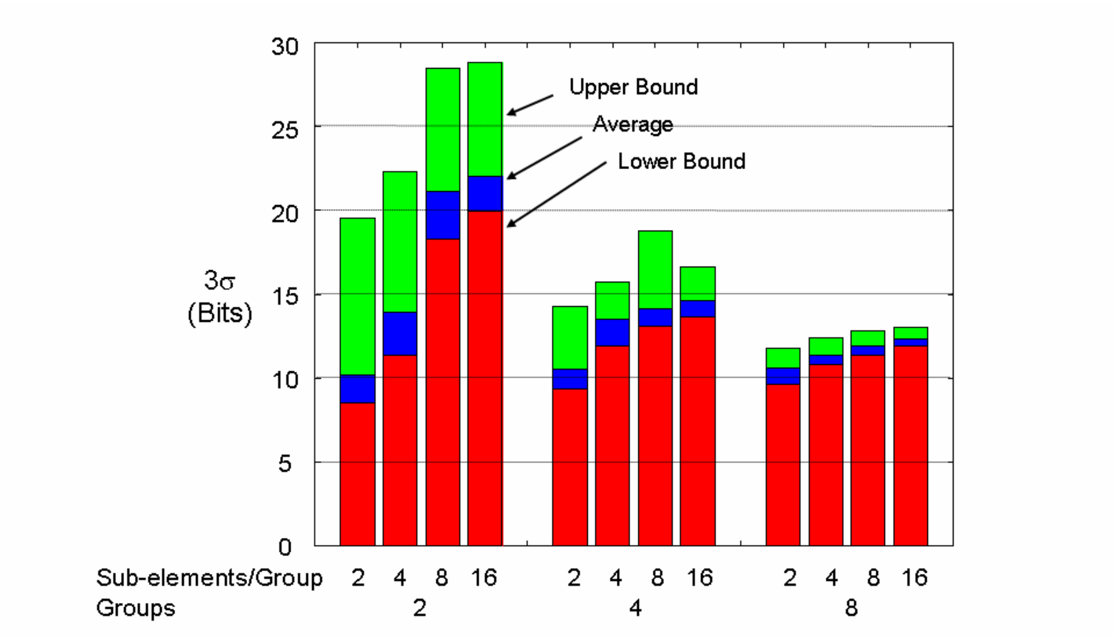


Figure 3.4: Example showing improvement in matching by configuring the sub-elements in different ways.

Another inference of interest may be drawn by examining the case of grouping the same total number of sub-elements in different number of groups, i.e. 8 groups of 2 sub-elements per group, 4 groups of 4 sub-elements per group and 2 groups of 8 sub-elements per group. The groups are distributed around the mean value by the amount of mismatch. Increasing the number of groups makes it harder to find a common value for each group to converge. As a result, the maximum amount of best case matching that can be achieved decreases. However, the difference between the lower and the upper bounds become tighter implying better yield. A more predictable solution is obtained because of the increased degrees of freedom of selecting sub-elements from a larger available pool of sub-elements. From Table 3.2, it can be concluded that in order to make an MDAC with two effective bits, 4 and 8 sub-elements per DAC capacitor group are good design choices.

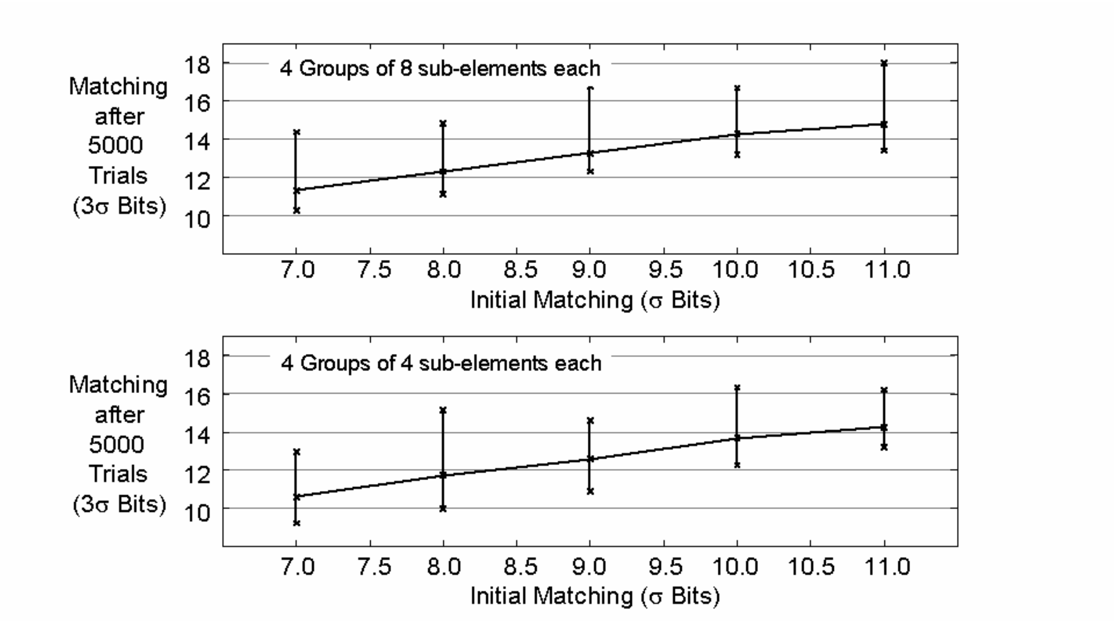


Figure 3.5: Simulated matching after self-configuration (5000 permutation and grouping trials) vs. initial matching of capacitor array.

Fig. 3.5 shows how the average value of matching relates to the initial matching level of the capacitor array for 4 DAC capacitor groups with 4 and 8 sub-elements each. The error bars correspond to the upper and lower bounds of the best case matching obtained for 100 Monte Carlo simulation experiments with 5000 trials each. The standard deviation initial error in unit sub-elements increases as each element is broken in more pieces. Usually the desired total capacitor in an MDAC is chosen from a thermal noise requirement. The act of breaking this to several sub-elements results poorer initial matching for each sub-elements as the number of sub-elements increase. This will affect the final matching performance achieved after self-configuration. Fig. 3.6 shows the matching performance obtained when breaking a

1pF capacitor with 0.1% initial matching into several sub-elements. It is seen that there is degradation in the improvement when the total number of sub-elements increase but this improvement is still good enough for the system being designed.

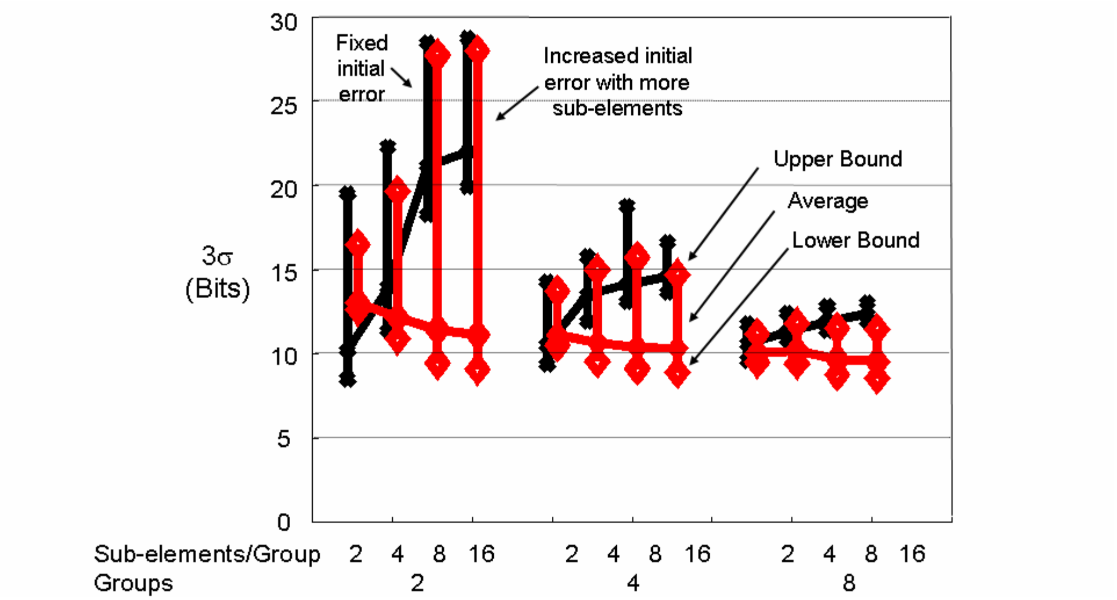


Figure 3.6: Change of improvement when inherent accuracy reduces as number of sub-elements increase.

Fig. 3.7 shows how matching improves very rapidly within the first 500 – 1000 trials so the matching value at 5000 trials is used as a computationally convenient asymptotic value. Four elements can be made from the pool of 32 sub-elements by combining 8 sub-elements in each group in around 2.5 x 10^{16} unique ways. It is not practical to examine all possible configurations (which might take over 19 years when explored using a 40MHz system clock). The proposed algorithm explores the permutation space in a random fashion. Simulations show rapid convergence to a 13b 3σ matching from a 10b σ matched array after about 500 to 1000 random tries.

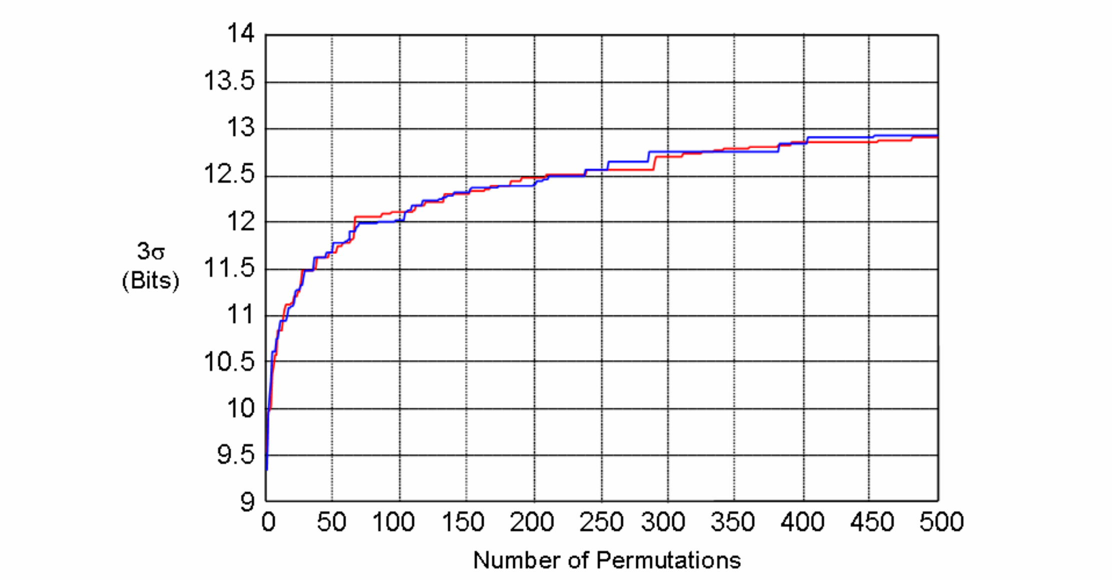
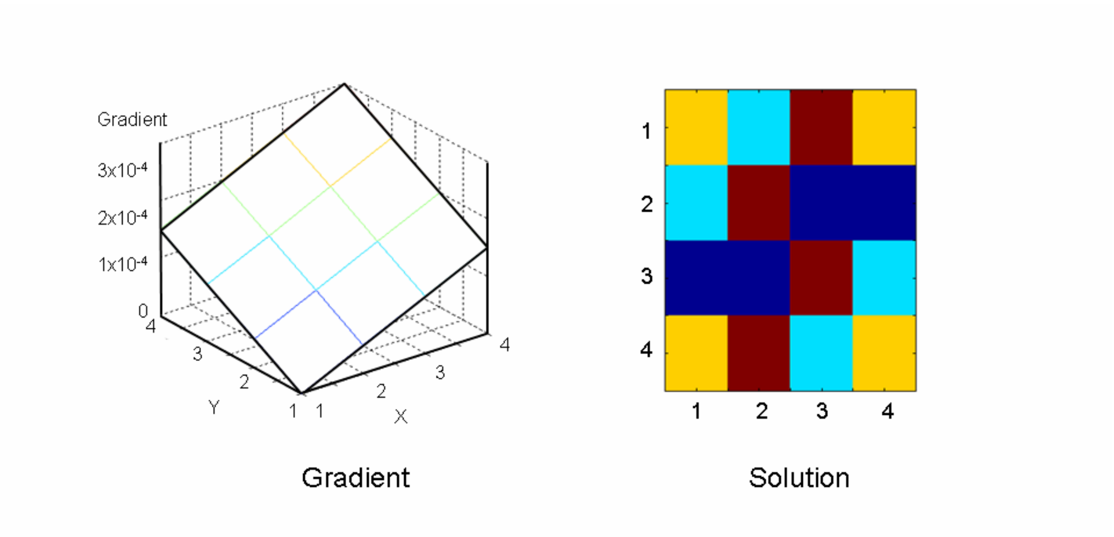


Figure 3.7: Two simulation cases showing improvement in matching vs. number of permutation trials.

Each point in the curve shown is a result of a Monte Carlo experiment with 500 samples. The worst of the series of best case matching values obtained after performing the required number of trials in each Monte Carlo experiment have been plotted so the curve represents the lower bound of the algorithm. As the algorithm explores the permutation space, the better matched values are easily encountered as seen in the rapid improvement in the matching during the first few configuration trials. However the process of finding configurations with even better matching becomes progressively difficult. This causes the rate of convergence to decrease as the number of iterations increase.

It is observed that in the presence of systematic errors in a chip, like gradient errors, and in the absence of random mismatch errors, the algorithm converges to the same solution as a common centroid layout of the capacitor array. Fig. 3.8 shows the case of an array generated with gradient errors. One of the solutions of the algorithm clearly shows that the groups are configured in a common centroid pattern.



Gradient

Solution

Figure 3.8: A solution of self-configuration algorithm for the case of systematic gradient errors in the absence of random mismatch errors.

Such an array has been used to design a pipelined ADC of 13-14b resolution. This technique is called self-configuration since the matching of the elements is improved through the use of a configuration of unit sub-elements and as demonstrated in Chapter 4, the algorithm is completely contained in the same chip as the ADC.

3.1.3: Analysis of the scheme

The principle can be mathematically explained using the concept of ordered statistics. Consider the case of n nominally identical sub-elements $x_1, x_2, \ldots x_n$, which are independent and identically distributed (i.i.d.) with a uniform distribution having mean μ and spread $\pm\lambda/2$. The mean and the variance of these uniformly distributed elements are given by Eqn. 3.2 and 3.3. When two elements are combined, the mean and variance are given by Eqn. 3.4 and 3.5.

$$E[x_i] = \mu \ . \qquad \text{… Eqn. 3.2}$$

$$Var[x_i] = \frac{\lambda^2}{12} \ . \qquad \text{… Eqn. 3.2}$$

$$E[x_i + x_j] = \mu + \mu = 2 \cdot \mu \ . \qquad \text{… Eqn. 3.4}$$

$$Var[x_i + x_j] = \frac{\lambda^2}{12} + \frac{\lambda^2}{12} = \frac{\lambda^2}{6} \ . \qquad \text{… Eqn. 3.5}$$

These elements can be sorted in terms of its size $y_1 < y_2 < \ldots < y_n$. The $y_1, y_2 \ldots y_n$ are called the ordered statistics [91], [92]. The ordered statistics have the following properties [93]. The mean value of an ordered element E[y_i] (i = 1, 2 … n) is given by Eqn. 3.7 and the variance and the covariance are given by Eqn. 3.8 and 3.9.

$$E[y_i] = \mu + \lambda \cdot \left(\frac{i}{n+1} - \frac{1}{2} \right) \ . \qquad \text{… Eqn. 3.7}$$

$$Var[y_i] = \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot i \cdot (n - i + 1) \ . \qquad \text{…. Eqn. 3.8}$$

$$Cov[y_i, y_j] = \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot i \cdot (n - j + 1) \ . \qquad \text{…. Eqn. 3.9}$$

$$i < j$$

When two sub-elements, say the end elements i.e. the 1st and nth ordered elements are combined, the new mean is computed by Eqn. 3.10 and the new variance is computed by Eqn. 3.11.

$$E[y_1 + y_n] = E[y_1] + E[y_n] = \mu + \lambda \cdot \left(\frac{1}{n+1} - \frac{1}{2} \right) + \mu + \lambda \cdot \left(\frac{n}{n+1} - \frac{1}{2} \right)$$

$$E[y_1 + y_n] = 2 \cdot \mu \ . \qquad \text{… Eqn. 3.10}$$

$$\begin{aligned} Var[y_1 + y_n] &= Var[y_1] + Var[y_n] + 2 \cdot Cov[y_1, y_n] \\ &= \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot 1 \cdot (n-1+1) \\ &\quad + \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot n \cdot (n-n+1) \\ &\quad + 2 \cdot \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot 1 \cdot (n-n+1) \end{aligned}$$

$$Var[y_1 + y_n] = \frac{2 \cdot \lambda^2}{(n+1) \cdot (n+2)} \ . \qquad \text{… Eqn. 3.11}$$

Similarly when the 2nd and the (n-1)th elements are combined, the new mean and variance is computed by Eqn. 3.12 and 3.13.

$$E[y_2 + y_{n-1}] = E[y_2] + E[y_{n-1}] = \mu + \lambda \cdot \left(\frac{2}{n+1} - \frac{1}{2} \right) + \mu + \lambda \cdot \left(\frac{n-1}{n+1} - \frac{1}{2} \right)$$

$$E[y_2 + y_{n-1}] = 2 \cdot \mu \ . \qquad \text{… Eqn. 3.12}$$

$$\begin{aligned} Var[y_2 + y_{n-1}] &= Var[y_2] + Var[y_{n-1}] + 2 \cdot Cov[y_2, y_{n-1}] \\ &= \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot 2 \cdot (n-2+1) \\ &\quad + \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot (n-1) \cdot (n-(n-1)+1) \\ &\quad + 2 \cdot \frac{\lambda^2}{(n+1)^2 \cdot (n+2)} \cdot 2 \cdot (n-(n-1)+1) \end{aligned}$$

$$Var[y_2 + y_{n-1}] = \frac{4 \cdot n \cdot \lambda^2}{(n+1)^2 \cdot (n+2)} \ . \qquad \text{… Eqn. 3.13}$$

Consider the simple case of 4 sub-elements grouped in 2 element groups of 2 sub-elements each. If the sub-elements were combined without the knowledge of the order, the variance of these groups is $\lambda^2/6$. If the 1st and the 4th sub-elements are combined together, the variance of the group is $\lambda^2/15$ and if the 2nd and the 3rd sub-elements are grouped together, the variance of the group is $\lambda^2/9.4$. If the 1st and the 3rd sub-elements are combined and the 2nd and the 4th sub-elements are combined, the variances can be derived to be $\lambda^2/8.33$ and $\lambda^2/10.71$ respectively. Similarly when the 1st and 2nd sub-elements are combined and the 3rd and 4th sub-elements are combined, the variances can be derived to be $\lambda^2/9.38$ and $\lambda^2/6.82$ respectively. This clearly shows that the mismatch between the two elements can be improved if the combination is done correctly and with knowledge of the relative sizes.

3.2: CALIBRATION OF A MULTI-BIT MDAC

A pipelined ADC is a suitable architecture for the implementation of high-speed high-resolution ADCs. As described in Chapter 2, several analog calibration techniques for pipelined ADCs target the 1.5 bit per stage implementation because the algorithms are relatively simple to implement. Many of these algorithms use the fact that the switched capacitor circuit used to implement an MDAC stage requires only two capacitors as shown in Fig. 1.18. It is conceptually simple to compare two elements with each other and correct the effect of the mismatch errors. A pipelined ADC that resolves more than one effective bit per pipelined stage is called a multi-bit per stage pipelined ADC and these use multi-bit sub-DACs commonly implemented using switched capacitor circuits. A multi-bit switched capacitor MDAC stage of a pipelined ADC similar to the one shown in Fig. 2.2 is used as the test structure to demonstrate the practical feasibility of this technique as a calibration method.

3.2.1: Foreground / offline error measurement method

As described in Chapter 2, the first step in the calibration process is to determine that an error has occurred and to measure the amount of error in the system being calibrated. The simplest way to do this is to use a foreground error-measurement technique.

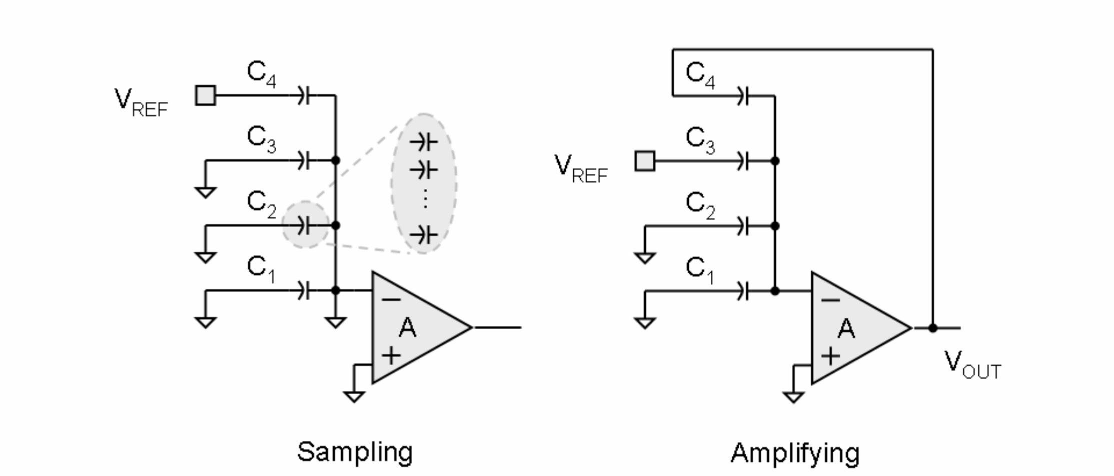


Figure 3.9: MDAC stages during calibration showing how the mismatch error between two capacitors can be measured.

The ADC is taken offline i.e. the input signal is not connected and the errors in the MDAC capacitors are measured by connecting these capacitors to the same reference voltages that are used for the normal operation. The MDAC errors in the permuted and combined capacitor arrays are measured in a way similar to [48] in order to choose the best-match configuration among many tried ones. Each capacitor shown in Fig. 3.9 is actually a group of sub-element unit capacitors configured using a permutation address. The procedure starts by choosing a configuration of the sub-element unit capacitors. A pseudo-random number is used to choose this configuration and this is similar to shuffling the available sub-elements. The act of choosing such a configuration and grouping the sub-elements to form the different capacitor groups in the MDAC is also called a "Shuffle". The mismatch error between any two capacitor groups say C_3 and C_4 can be measured using the following steps: (1) In the sampling phase, the capacitor group C_4 is pre-charged to the reference voltage while all other capacitor groups are discharged to ground. (2) This pre-charged capacitor group C_4 is connected in feedback during the amplifying phase while the capacitor group C_3 is connected to the same reference voltage. Charge conservation at the virtual ground node shows that the output at the end of the amplifying phase is proportional to the mismatch between the two capacitor groups given by Eqn. 3.14.

$$V_{OUT} = \frac{V_{REF} \cdot \left(1 - \frac{C_3}{C_4}\right)}{1 + \left(\frac{C_1 + C_2 + C_3 + C_4}{A \cdot C_4}\right)} \quad \text{... Eqn. 3.14}$$

This voltage is converted into a digital word using the pipelined ADC formed by the lesser significant stages. Similarly, each capacitor group in the array can be in turn configured as the feedback capacitor or the capacitor being compared against the feedback capacitor. This leads to a set of measurements representing the mismatch error, called the code-spread, in the capacitor array from which the spread of component values can be determined. These steps are visually represented in Fig.

3.10. The advantage of this method is that the measurement is not sensitive to offset errors in the operational amplifiers. The capacitors are compared against themselves giving an absolute measure of spread. The high accuracy objectives are met when the configured capacitor groups are equal or as close to each other as possible and this is indicated when the code-spread is zero or as small as possible. Since the measurement is ideally close to zero, the effects of finite operational amplifier gain affect the measurement to a lesser extent.

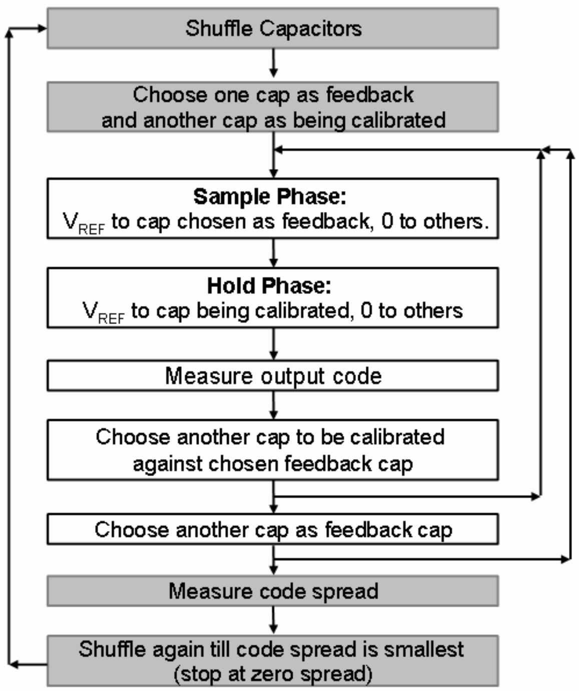

Shuffle Capacitors
Choose one cap as feedback and another cap as being calibrated
Sample Phase:
V_{REF} to cap chosen as feedback, 0 to others.
Hold Phase:
V_{REF} to cap being calibrated, 0 to others
Measure output code
Choose another cap to be calibrated against chosen feedback cap
Choose another cap as feedback cap
Measure code spread
Shuffle again till code spread is smallest (stop at zero spread)

Figure 3.10: Sequence of steps to measure spread of a configuration.

The effect of thermal noise is counteracted by performing each measurement cycle for several conversion cycles and by averaging the end result. The disadvantage of the foreground error-measurement method is that the data converter system is taken offline. The entire calibration sequence has to be repeated if the errors change due to component drift or aging and errors are not tracked in time. A simulation of this scheme in a MATLAB program is shown in Fig. 3.11. The THD at the output of the ADC is used as a figure of merit for this simulation to reduce the simulation time instead of the INL, which is a better figure of merit but increases the simulation time. It is observed that this algorithm improves the performance of the ADC beyond that achievable using a 10 b matched array in the MDAC. This technique has been implemented in the prototype chip described in Chapter 4.

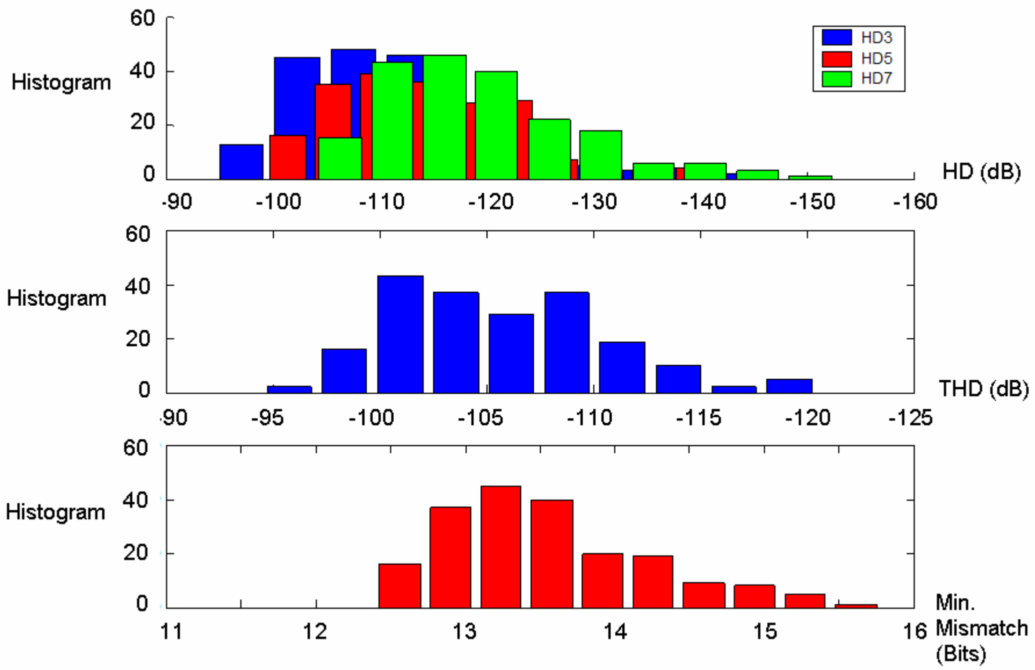


Figure 3.11: Foreground calibration simulations.

3.3: CONCLUSION AND PROPOSED FURTHER WORK

A statistical capacitor matching technique is proposed to improve the matching between nominally alike elements in order to design an accurate DAC in a multi-bit per stage pipelined ADC. Smaller sized elements are combined in various configurations with the aim to reduce the spread between the reconstructed elements. A random search algorithm is proposed that converges quickly to a useable configuration. An offline error measurement technique is described that will be used to achieve a high accuracy pipelined ADC.

Future work should focus on applying this scheme to other data converter architectures. A rigorous theoretical study of the basic scheme should be carried out.

Parts of this chapter have been published in S. Ray, Bang-Sup Song, "A 13b linear 40MS/s pipelined ADC with self-configured capacitor matching," *ISSCC Dig. of Tech. Papers.* pp. 852-861, Feb. 2006 (© 2006 IEEE) and Sourja Ray; Bang-Sup Song, "A 13-b Linear, 40-MS/s Pipelined ADC With Self-Configured Capacitor Matching," *IEEE J. Solid-State Circuits*, vol. 42, no.3, pp.463-474, Mar. 2007 (© 2007 IEEE) with the dissertation author as the primary author of these papers. The relevant sections have been reprinted with permission of the IEEE.

Chapter 4:

Calibration of the unit capacitors in a multi-bit per stage MDAC used in a pipelined ADC using self-configuration

A practical implementation of the statistics based self-configuration technique described in Chapter 3 to improve the accuracy of a multi-bit per stage pipelined ADC is presented.

4.1: SYSTEM DESCRIPTION

A multi-bit per stage pipelined ADC is used as a test system to verify the feasibility of the self-configuration scheme described in Chapter 3. The ADC is implemented using seven switched capacitor pipelined MDAC stages each resolving two effective bits followed by a three bit flash ADC to resolve a total of 16 bits after digital reconstruction. The capacitor-matching algorithm is used to calibrate the first three stages of this ADC as shown in Fig. 4.1. The back-end stages are not calibrated since the matching requirements of the capacitors is sufficiently relaxed to achieve the accuracy requirements of these stages. More stages have been implemented on the chip than strictly needed in order to examine the performance of the algorithm. The algorithm is run offline (foreground) in this work since capacitor errors do not change much over time and temperature. The configuration of the capacitive sub-elements that results in the least error is detected using the self-configuration engine during the calibration procedure and stored in the on-chip memory. This configuration can also be offloaded to the host system using the serial port. This configuration can be quickly loaded during normal operation and used to improve the accuracy of the converted word.

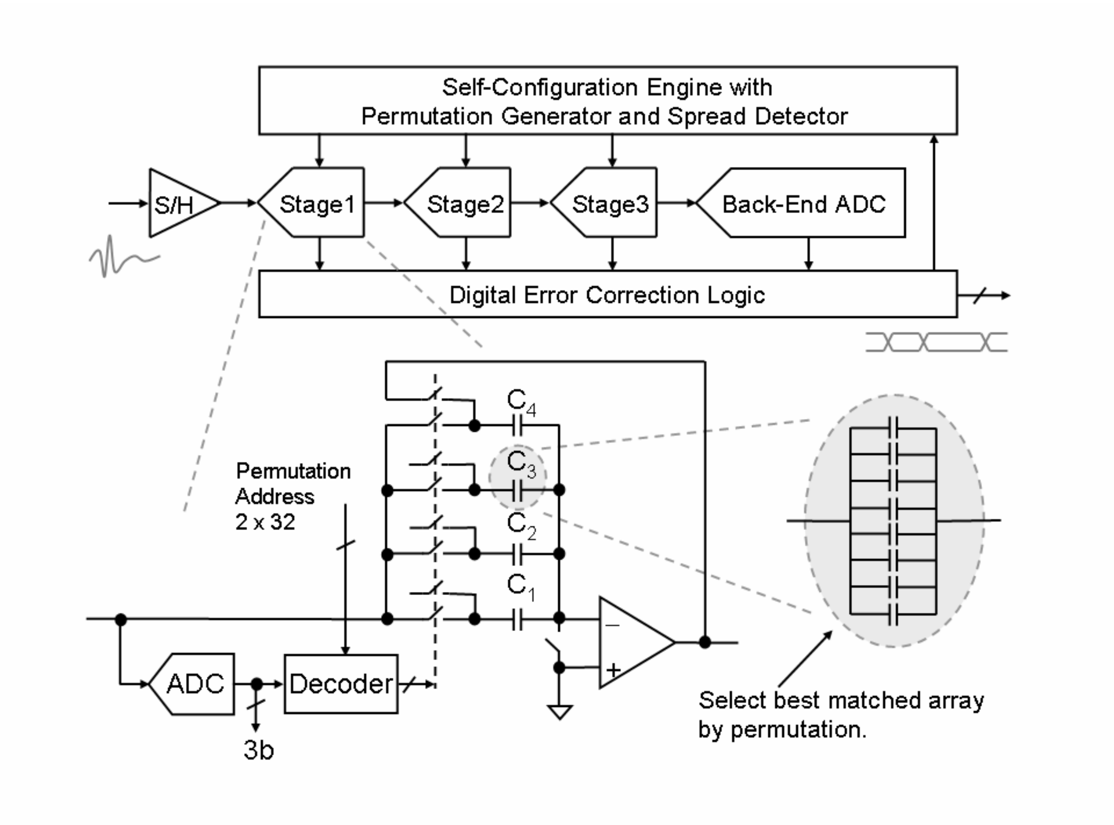

Self-Configuration Engine with
Permutation Generator and Spread Detector
S/H
Stage1
Stage2
Stage3
Back-End ADC
Digital Error Correction Logic
C_4
C_3
C_2
C_1
Permutation
Address
2 x 32
ADC
Decoder
3b
−
+
Select best matched array
by permutation.

Figure 4.1: Block diagram showing digital self-configuration engine driving the MDAC stage with self-configured capacitor array.

4.2: ANALOG CMOS IMPLEMENTATION

This section describes the implementation details of the various analog circuits in the ADC. Designing circuits to operate off a low voltage supply using transistors available in fine line CMOS processes pose many design challenges. A combination of circuit and architectural techniques has been used to make this implementation feasible.

4.2.1: Front end sample and hold amplifier

The front end sample and hold amplifier (SHA) samples the analog input of the chip and freezes the sample in time thus converting the continuous-time continuous-valued analog input to a discrete-time continuous-valued voltage that is subsequently quantized by the ADC. The performance of this block sets a limit to the achievable SNR and linearity of the overall ADC. Both the sampling network and the operational amplifier are critical circuits that limit the noise and linearity in this block. The top and the bottom plate switches in the sampling network are implemented using NMOS transistors as shown in Fig. 4.2. The low supply voltage supplies used in modern CMOS processes imply that the signal swing is a significant fraction of the available supply and clock voltage. This implies that the transistors in the front-end switch have a low overdrive voltage during the "on" phase that also changes with the signal, which is known to be a significant source of non-linearity [94], [95]. This is avoided by using a bootstrapped scheme with a circuit similar to that in [7] in order to obtain good linearity. A capacitive level shifter C_{SHIFT} is pre-charged to the supply voltage V_{DD} during the inactive phase and during the sampling phase is connected between the source and the gate of the switch transistor M_1 thus providing a constant gate to source voltage independent of the input signal. This improves the linearity of the input sampling network. It has been reported that residual signal dependent component in the overdrive voltage due to the body effect increases harmonic distortion in the switch [96] but this was not determined to be a source of error for this circuit. Boosted clocks using voltage-doubling circuits generated by the clock generator drive the top plate switches of the switched capacitor network. The active switches are sources of thermal noise and this noise is integrated on the sampling capacitor.

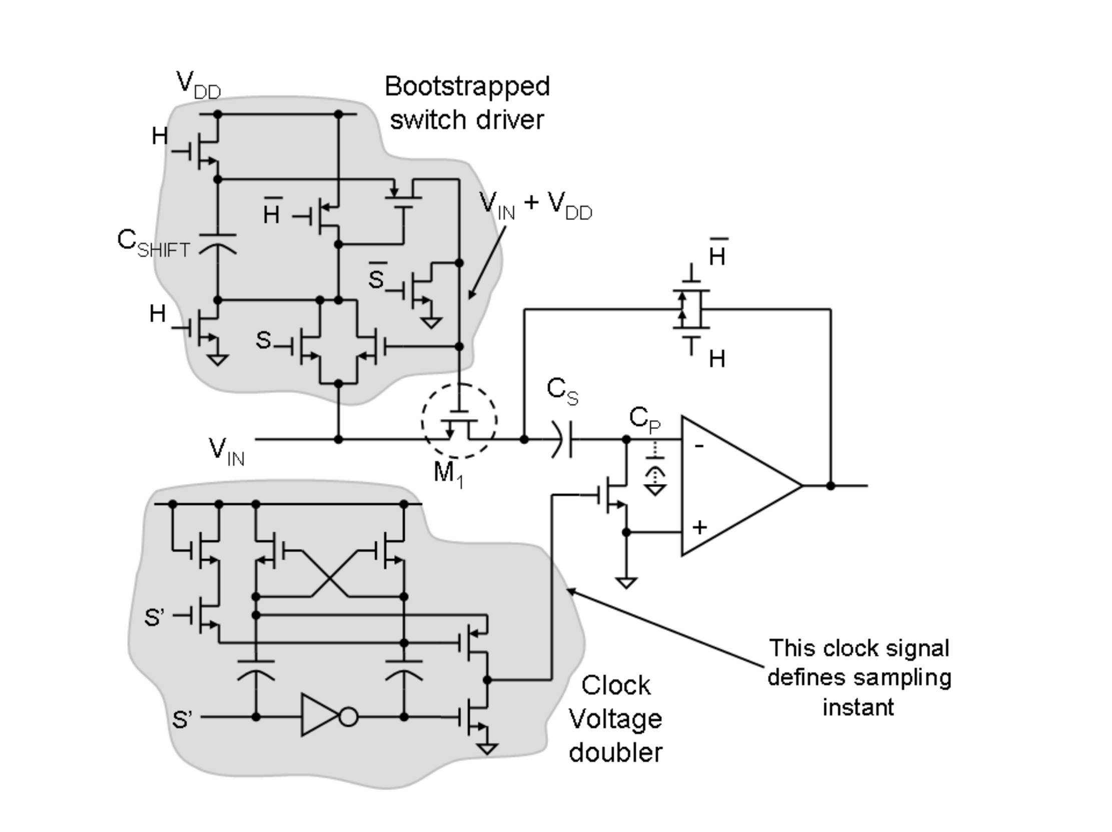


Figure 4.2: Sample and hold amplifier.

A 2pF capacitor C_S is chosen for the sampling capacitor as per the system noise budget. During hold phase, this capacitor is connected across the operational amplifier thus leading to a unity gain configuration. The operational amplifier is implemented using a two-stage Miller compensated amplifier as discussed later in this chapter. The amplifier is connected in unity gain feedback with a very low feedback factor *(C_S / C_S + C_P)* thus making it difficult to meet the stability requirements. This increases the power consumption in the output stage of the amplifier.

4.2.2: MDAC

The two effective bits per stage MDAC without calibration is implemented using four unit capacitors and tri-level DAC control switches as shown in Fig. 2.1. These are used in the back-end stages. The total sampling capacitor in the fourth and fifth stages is chosen to be 1pF with each unit capacitor implemented using a 250fF M-I-M structure. The total sampling capacitor in the sixth and seventh stages is chosen to be 500fF with each unit capacitor implemented using a 125fF M-I-M structure. The sub-ADC thermometric code is converted to the tri-level code driving the MDAC switches by the DAC decoder block. The truth table for this block is shown in Table 4.1. A similar block is used in the MDACs being calibrated. The two effective bits per stage MDAC with calibration is an extension of this architecture. Four capacitor groups are used instead of the four capacitors in the first three stages being calibrated as shown in Fig 4.1. The sampling capacitors have been chosen to be 2pF in the first stage and 1pF each in the second and third stages from thermal noise calculations. Simulations of the capacitor-matching algorithm show that the sampling capacitor has to be broken into 32 sub-elements in the first stage and 16 sub-elements in the second and third stages in order to meet the desired matching accuracy. This leads to sub-element sizes of 62.5fF in each of the MDACs being re-configured. The residual error in the matching of the four groups of capacitor determines the accuracy of this

MDAC. The capacitor-matching algorithm is implemented by permuting and grouping the sub-elements to each of these capacitor groups.

Table 4.1: Truth table for thermometric code to tri-level DAC decoder

Sub-ADC Thermometer Code						Code	Tri-level DAC Control Code								
F0	**F1**	**F2**	**F3**	**F4**	**F5**		**D0 P**	**D0 C**	**D0 N**	**D1 P**	**D1 C**	**D1 N**	**D2 P**	**D2 C**	**D2 N**
0	0	0	0	0	0	**0**	1	0	0	1	0	0	1	0	0
1	0	0	0	0	0	**1**	1	0	0	1	0	0	0	1	0
1	1	0	0	0	0	**2**	1	0	0	0	1	0	0	1	0
1	1	1	0	0	0	**3**	0	1	0	0	1	0	0	1	0
1	1	1	1	0	0	**4**	0	0	1	0	1	0	0	1	0
1	1	1	1	1	0	**5**	0	0	1	0	0	1	0	1	0
1	1	1	1	1	1	**6**	0	0	1	0	0	1	0	0	1

An address is assigned to each of these capacitor groups. Address 0 (implemented using two bits {0,0}) is used as the address for the feedback capacitor group. Addresses 1, 2 and 3 (implemented using two bits {0,1}, {1,0} and {1,1}) are used as the addresses for the three DAC capacitor groups. As described in Section 3.2.1, address 1 is also used to identify the capacitor group that is compared with the feedback capacitor group during foreground calibration. Each sub-element can be assigned one of these four addresses. All the sub-elements are connected to the output of the previous stage during the sample phase during normal operation. The addresses are asserted during the hold phase thus assigning a function to each of these sub-elements to act as the feedback or as one of the DAC capacitors. The outputs of the sub-ADC control the value to which the sub-elements configured as the DAC capacitors connect to during the hold phase. The outputs of the tri-level DAC control decoder are combined with the permutation/remapped address to form the actual DAC switch control signals as shown in Fig. 4.3.

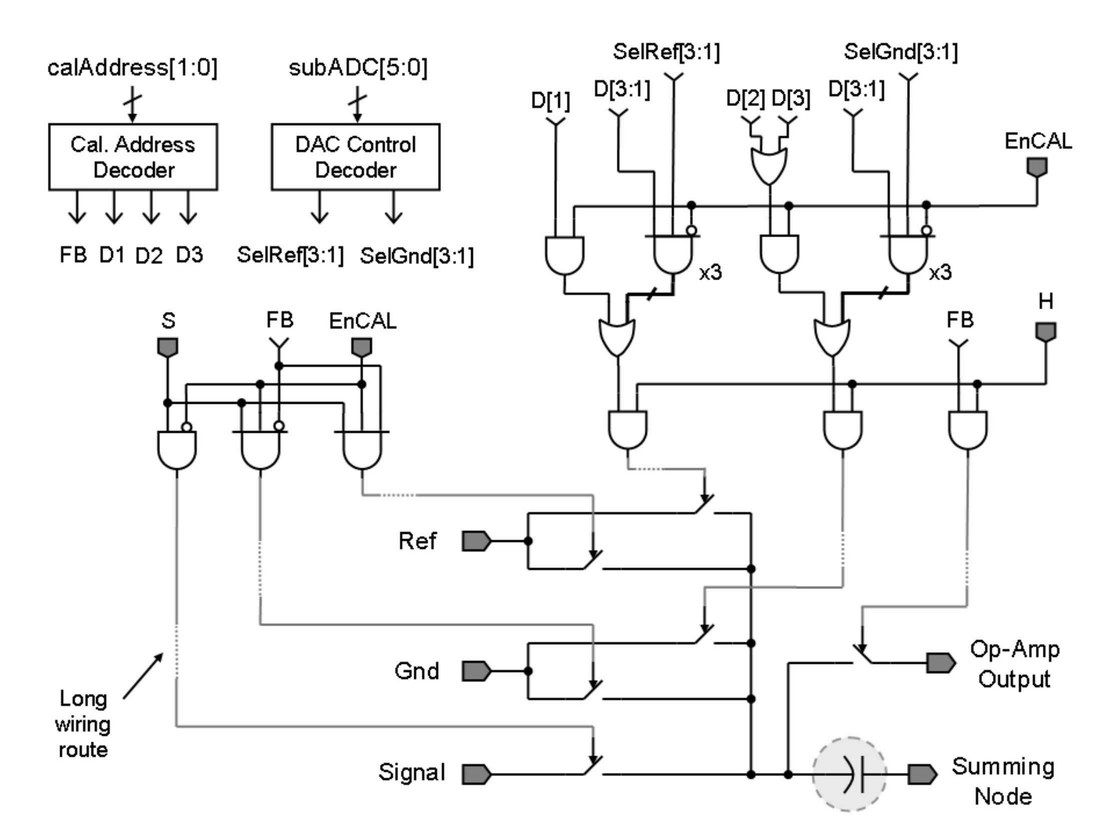


Figure 4.3: Structure of each sub-element along with analog switches.

The output of the sub-ADC (subADC[5:0]) is decoded by the "DAC Control Decoder" block to form the DAC control signals SelRef[3:1] and SelGnd[3:1] required for the operation of a conventional MDAC stage without calibration. The

"Calibration Address Decoder" block decodes the two-bit sub-element address (calAddress[1:0]) to a one-hot code. One of the lines FB, D_1, D_2 or D_3 is asserted based on the function of the sub-element as a feedback or one of the three DAC capacitors. During the normal mode of operation (EnCAL=0), the switches sample the output of the previous stage in the sampling phase (S=1, H=0). During the hold phase (S=0, H=1), the switches are connected so that the capacitor is connected as a DAC element or in feedback. When the MDAC stage is being calibrated (EnCAL=1), the switches are connected to appropriate voltages as described in Section 3.2.1 during both the sample and the hold phases. A single-ended version is shown for simplicity though a differential version is used in the actual design. This timing critical path is designed using custom logic to minimize path delay. The switch control signals run for a long distance on the chip which makes the routing parasitic capacitance on these lines to be large (around 100fF – 200fF). The logic gates have been designed to drive this additional load.

Fig. 4.4 shows the layout of stage 1, i.e. a stage whose performance is being improved using self-configuration. The layout of stages 2 and 3 is similar to this. The wiring that connects the decoder to the switching network is the most striking feature of this floorplan. Fig. 4.5 shows the layout of stage 4, which is a conventional switched capacitor MDAC without calibration.

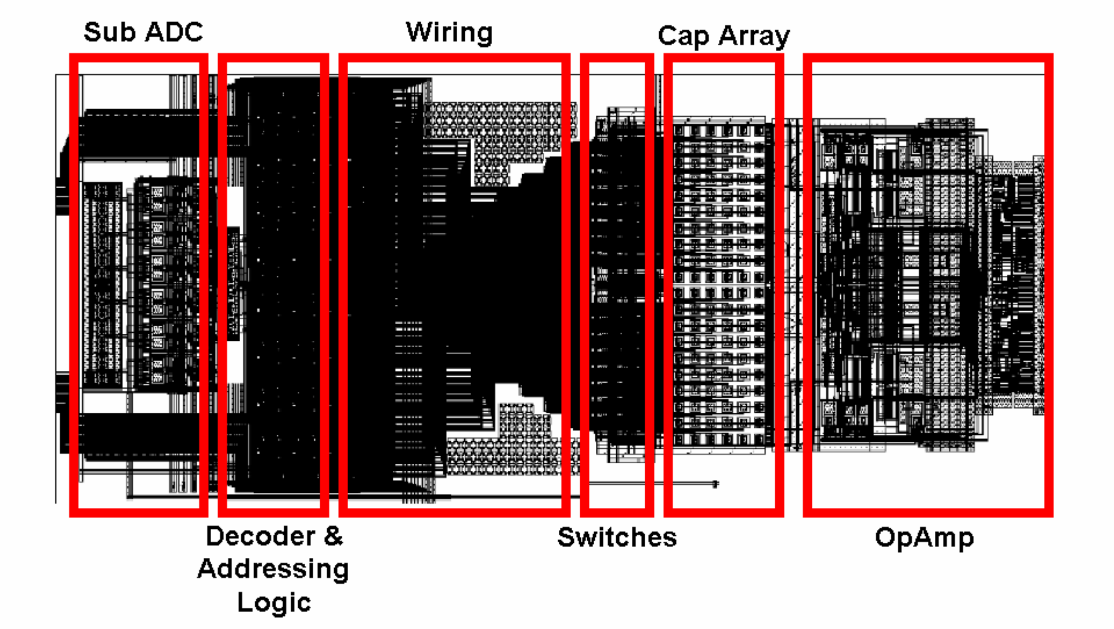


Figure 4.4: Layout of a stage with self-configuration (Stage 1).

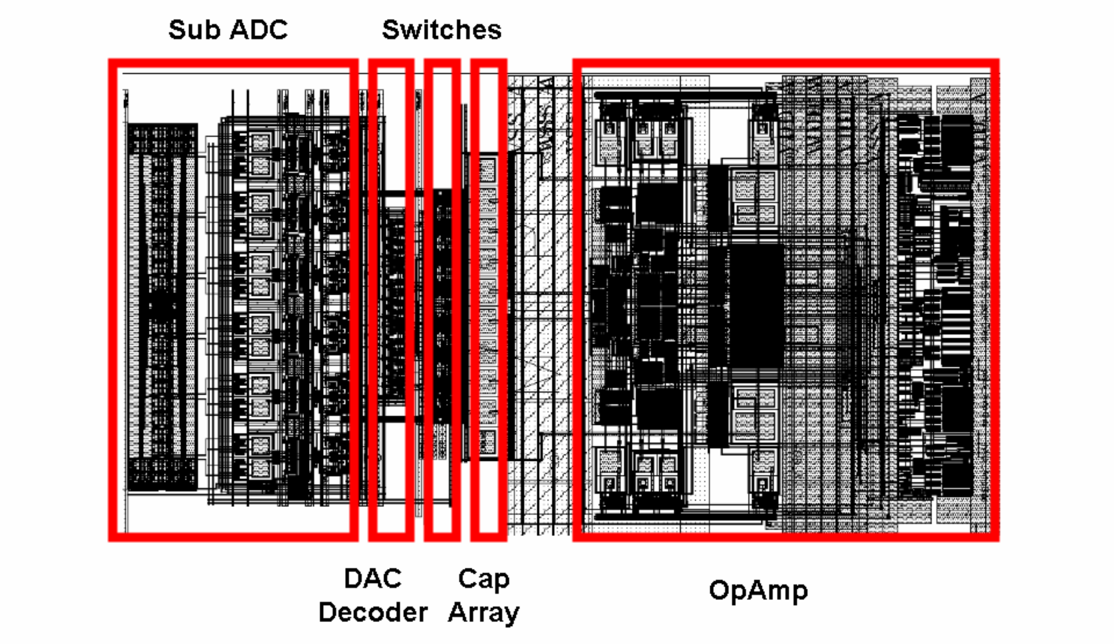


Figure 4.5: Layout of a stage without self-configuration (Stage 4).

It is clear that the present implementation of the switching network does not have a very clean layout and an alternate switching scheme should be investigated to make a simpler interconnection scheme.

4.2.3: Operational amplifier

High resolution switched capacitor MDACs need high DC-gain operational amplifiers (op-amps) to reduce the static error due to finite operational amplifier gain. The settling accuracy is dictated by the unity-gain frequency of the amplifier. A high unity-gain frequency is required in order to sample at high clock speeds and settle at the desired accuracy. The amplifier outputs should be able to support a large voltage swing for good signal to thermal noise ratio. It is advantageous to use the digital transistors with a small channel length in fine geometry CMOS processes in the design of these amplifiers because of the high transition frequency f_T (g_m/C_{gs}) of these transistors. However, these transistors have a low inherent gain ($g_m.r_{out}$). It is necessary to connect several transistors in a cascade or cascode configuration in order to obtain the necessary gain. The obvious choice is the telescopic cascode configuration since this has the fewest additional poles and can be designed to be the fastest amplifier with more than one transistor in the signal path. In order to meet the gain requirements, it is necessary to add gain boosting both on the N and on the P sides. This configuration by itself could be designed to obtain the necessary output swing. However since these amplifiers are used in a multi-bit MDAC with a gain of four, the feedback factor is at least 12 dB which makes the power efficiency poor when driving the large sampling capacitor of the next stage. A two-stage Miller compensated op-amp structure with a gain boosted telescopic cascode first stage followed by a common source second stage as shown in Fig. 4.6 is chosen as the op-amp that meets the gain and the bandwidth specifications with adequate power efficiency.

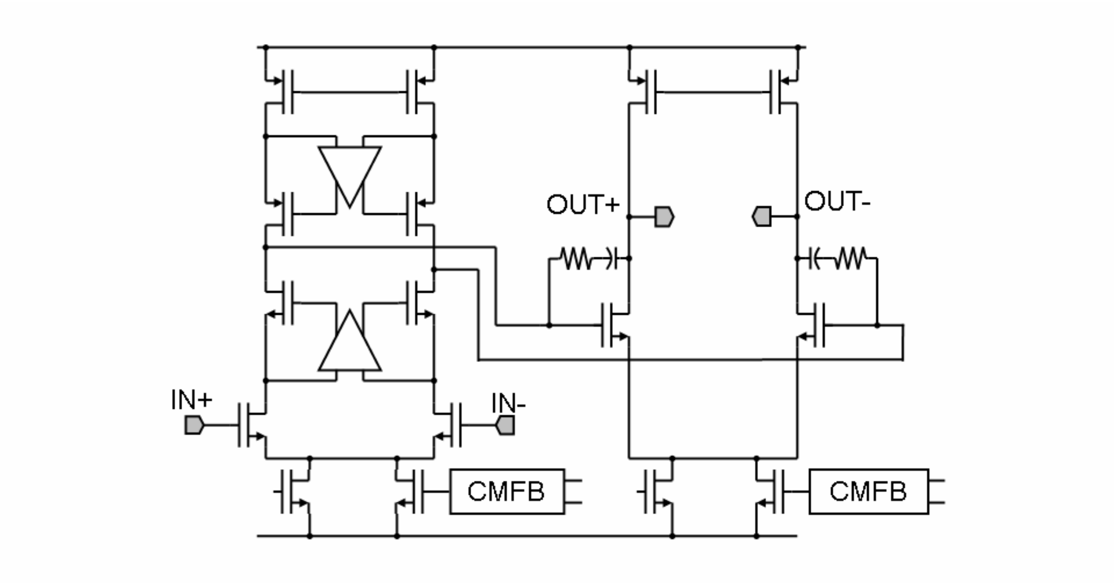


Figure 4.6: Miller compensated operational amplifier.

The closed-loop bandwidth ω_{CL} of an operational amplifier is dictated by the settling accuracy required at the output of the stage i.e. this depends on the resolution of the ADC N and the number of effective bits resolved by the stage n as given by Eqn. 4.1. The available time for settling T is usually slightly smaller than half the sampling period of the ADC to account for the rise and fall times of the clocks and the non-overlap period introduced by the clock generator. The settling accuracy is chosen as ¼th LSB of the accuracy requirements of the lesser significant stages of the pipelined ADC.

$$\omega_{CL} = \frac{(N - n + 2) \cdot \log_e(2)}{T} \quad . \qquad \text{… Eqn. 4.1}$$

The design procedure for a Miller amplifier typically begins with an initial arbitrary choice of compensation capacitor usually based on experience and rules of thumb [97]. The phase margin is selected based on the desired amplifier transient settling response. This is achieved through the choice of the location of the dominant and the non-dominant poles. Such a design technique requires several iterations since the parasitic capacitances due to the size of transistors are not known beforehand and these affect the solution. A new parasitic and device scaling-aware technique is proposed that relies on the concept of a unit amplifier. This simplifies the design procedure and leads to an optimum solution of a Miller compensated amplifier that has a unique compensation capacitor and device sizes for a chosen phase margin and signal to noise ratio. This is described in further detail in Appendix A.

A common technique to save power in Miller compensated amplifiers involves placing the non-dominant pole ω_{NDP} very close to the unity-gain frequency and position the zero ω_Z from the right half to the left half of the S-plane using a zero setting resistor R_Z in a way to cancel the pole thus resulting in a good phase margin. The trans-conductance of the MOS transistor in the output stage g_{mO} of the op-amp varies with output voltage swing due to channel length modulation as shown in Fig 4.7. This effect is pronounced in fine line CMOS processes at a low power supply level where the MOS transistors are biased with a low drain to source saturation voltage V_{DSAT}. This variation causes the position of the non-dominant pole and the zero to change differently as shown by the sensitivity functions in Eqn. 4.2.

$$S(\omega_{NDP}, g_{mO}) = \frac{d\omega_{NDP}}{dg_{mO}} \bigg/ \frac{\omega_{NDP}}{g_{mO}} = -g_{mO}$$
$$S(\omega_{Z}, g_{mO}) = \frac{d\omega_{Z}}{dg_{mO}} \bigg/ \frac{\omega_{Z}}{g_{mO}} = -\frac{1}{1 - R_Z \cdot g_{mO}} \quad . \qquad \text{… Eqn. 4.2}$$

This results in a pole-zero doublet close to the unity-gain frequency, as shown in Fig. 4.7, causing a slow settling behavior [98]. Good phase margin has been obtained in this design by placing the output pole at a frequency higher than the unity-gain frequency. The zero is then placed at a frequency higher than the non-dominant pole instead of attempting to improve the phase margin by canceling the non-dominant pole.

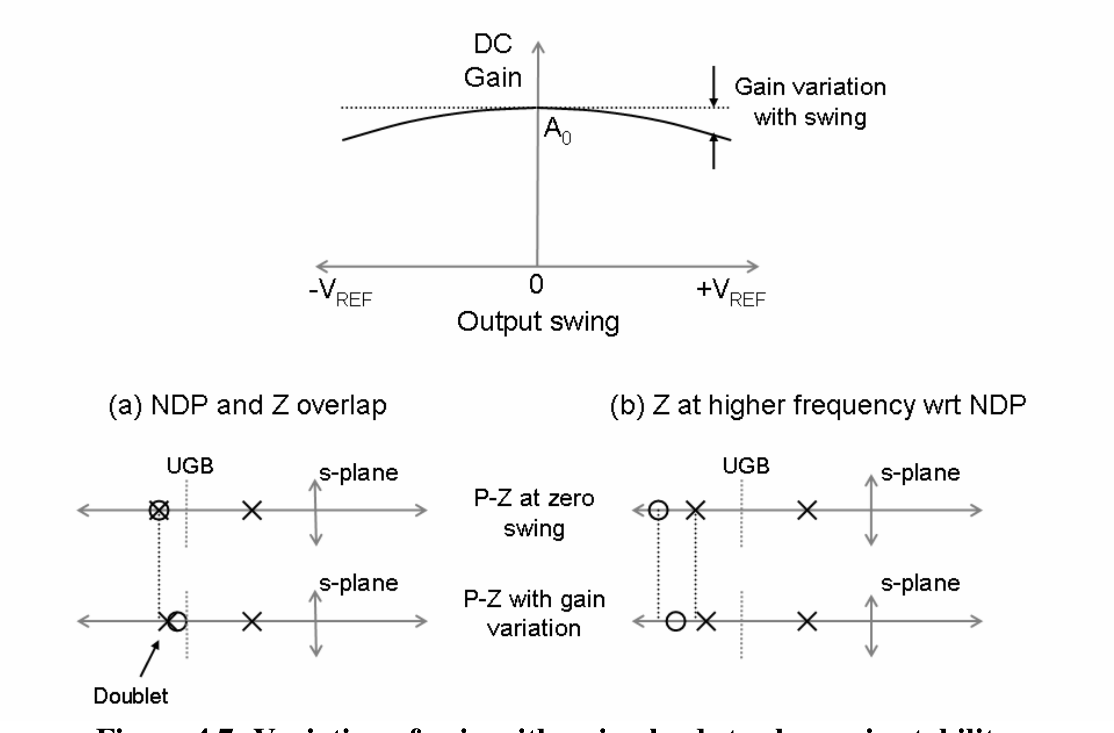


Figure 4.7: Variation of gain with swing leads to change in stability.

It is common practice to implement the Miller zero setting resistor using a transistor in the triode region with the assumption that the conductance of this transistor tracks the transconductance of the output transistor. This transistor is usually biased using a V_{GS} above the input common mode of the second stage. This is difficult to achieve in the available supply voltage. Poly resistors are used to position the zero in this design since the zero is placed beyond the non-dominant pole. It is verified that the zero does not move back to the right half of the s-plane with process and temperature variations.

4.2.4: Sub-ADC

Each stage in this prototype resolves two effective bits with redundancy for digital error correction. This is implemented by using a sub-ADC with six comparators in each stage. The sub-ADC partitions the input range into seven regions thus resolving 2.8 bits of information. Each stage in this prototype resolves two effective bits with 0.8 bit of redundancy for digital error correction. The analog threshold voltages required for the six sub-ADC comparators are generated differentially from a resistor ladder connected across externally supplied reference voltages. The unit resistor shown is chosen to be 200 ohm so that the reference nodes settle within a 50MHz clock cycle.

The comparator schematic is shown in Fig. 4.8. Each comparator consists of an offset canceled pre-amplifier followed by a regenerative latch. The pre-amplifier provides a gain of about 10. A switched capacitor network operates on the S and the H clock phases as shown. S’ and H’ clock phases shown are early clock phases. The capacitor *C* is pre-charged during a hold (H) phase to the reference value V_{REF}. Also during this phase, the preamplifier is connected in feedback because of which a voltage proportional to the input referred offset V_{OFFSET} is also stored in this capacitor. During the next sample (S) phase, the input parasitic capacitance C_P of the pre-amplifier is used to sample the input V_{IN} of the comparator, which is also the output of the previous pipelined stage.

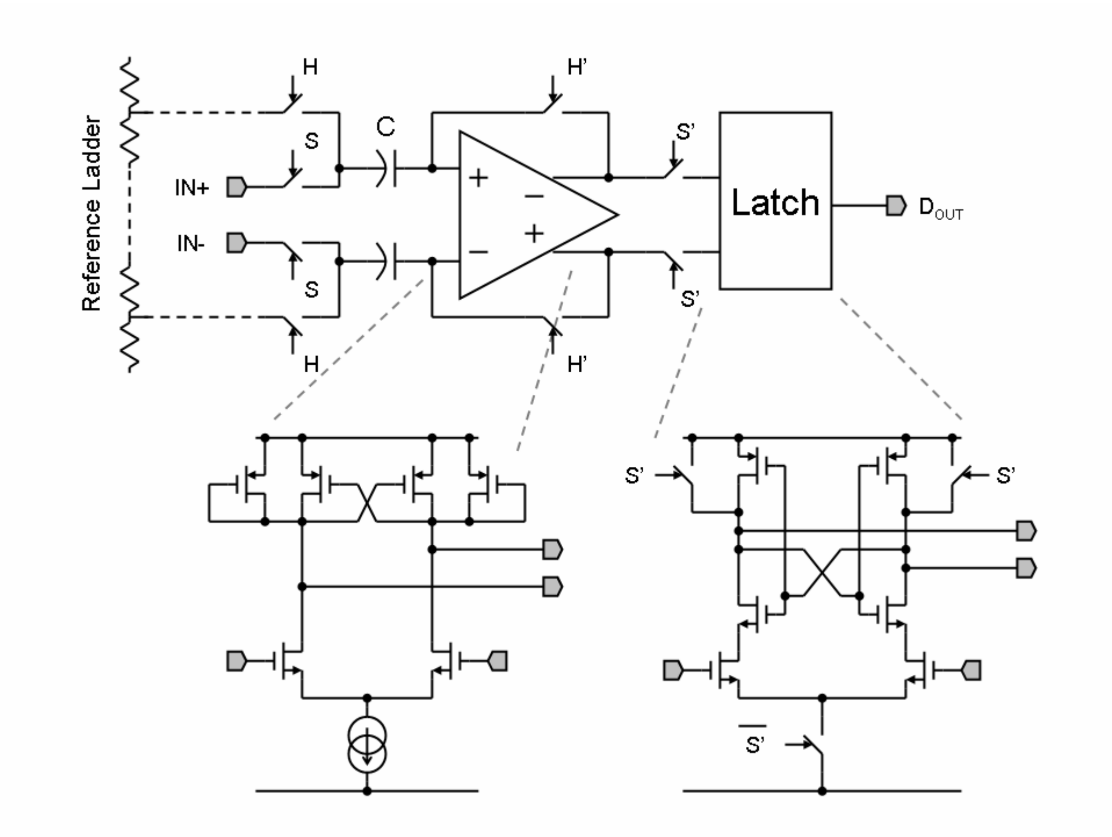


Figure 4.8: Sub-ADC comparator.

Eqn. 4.3 shows that the offset of the preamplifier is suppressed by the gain of the preamplifier A, in this case suppressed by a factor of 11. The output is proportional to the difference between the input and the reference value.

$$V_{PREAMP} = -A \cdot \left(\frac{V_{OFFSET}}{1+A} + \frac{C}{C+C_P}\left(V_{IN} - V_{REF}\right) \right) \qquad \ldots \text{ Eqn. 4.3}$$

The latch is enabled at the end of the sample phase so that DAC control signals are ready during the hold phase of the gain stage.

4.2.5: Clock generator

The clock generator generates non-overlapping clocks necessary for the operation of all the switched capacitor circuits. The charge transfer time in high speed ADCs is extremely short. Accurate charge transfer is also necessary to maintain the fidelity of the data conversion process. Large switches are needed in the switched capacitor circuits in order to maintain low on-resistance. This presents a large capacitive load that has to be driven by the clock generator. A cascade of progressively increasing size inverter stages is needed in the clock path in order to drive the switches with short rise and fall times. The design is simplified by splitting the output driver to multiple parallel paths as shown in Fig. 4.9. The different clock paths are used to drive different parts of the pipelined ADC. Two paths shown in this figure generate the A and B clock outputs for each phase. This scheme can be extended for more parallel paths depending on the total load capacitance to be driven. The output clock waveforms have different rise and fall times since these paths drive different capacitances. Non-overlapping phases can be guaranteed if the feedback points are tapped at the output nodes. The OR gates in the feedback path synchronize the multiple output paths to a single feedback signal. This ensures that the distributed clock phases remain non-overlapping.

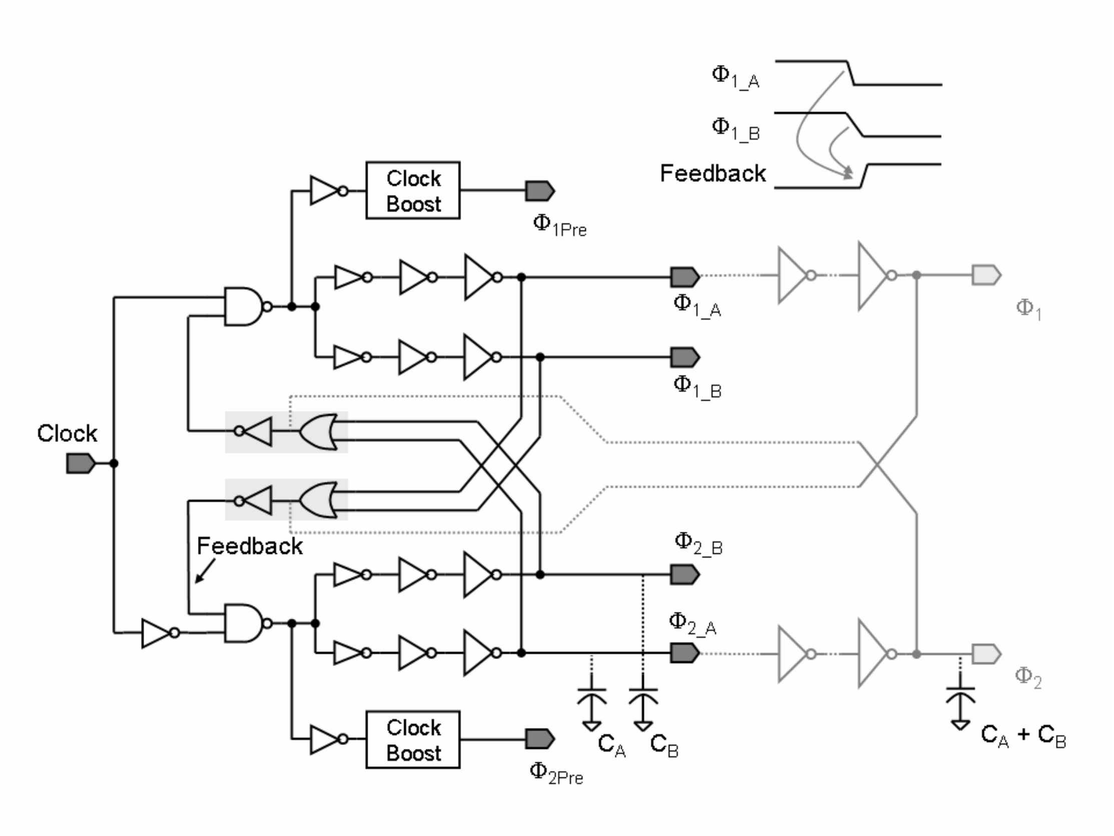


Figure 4.9: Clock generator.

Voltage doubling circuits (Clock Boost) are also provided to drive the top plate early phase switches. The sampling instant is defined by the falling edge of the early phase driving the top plate switches in the front end Sample and Hold amplifier. This is a critical clock signal and is kept as clean as possible by using a relatively early clock voltage driving a single pull-down NMOS transistor similar to that shown in Fig. 4.2.

4.3: DIGITAL SYSTEM IMPLEMENTATION

The block diagram of the digital system that implements the self-configuration logic is shown in Fig. 4.10. The digital system is coded using Verilog Hardware Description Language (HDL) and is realized in hardware form using logic synthesis and automatic place and route procedures. The digital logic is implemented using less than 400k gates.

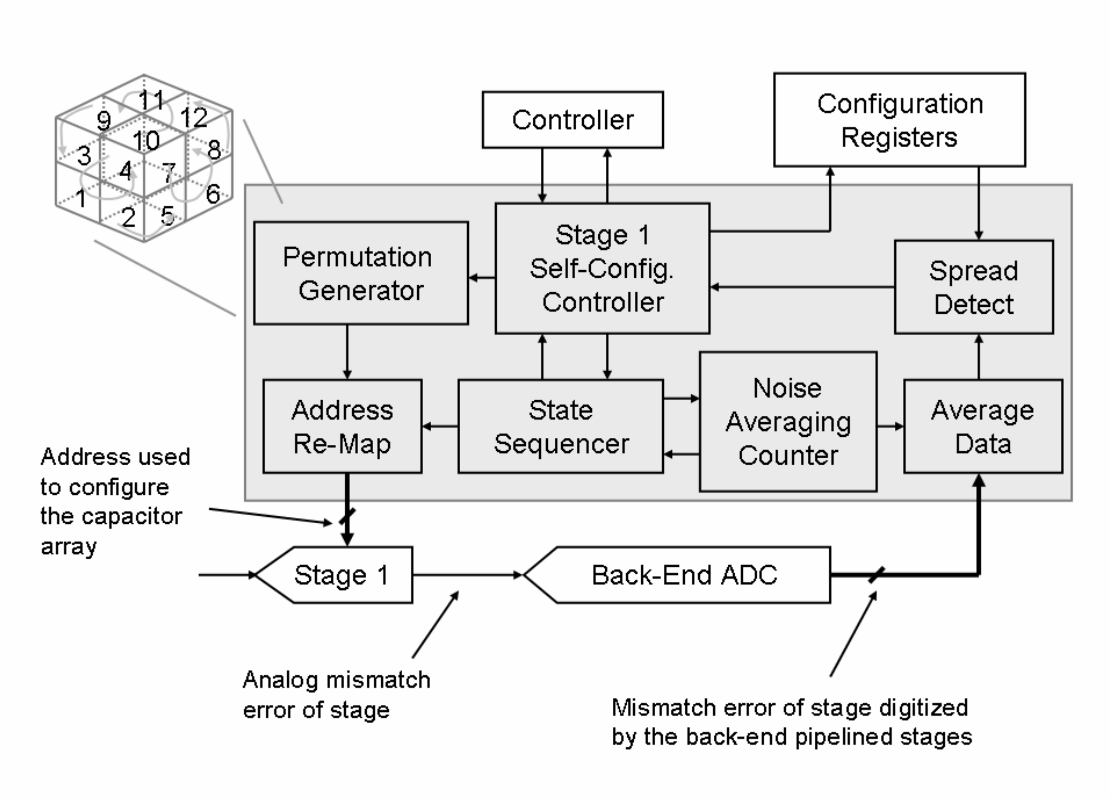

11
9
12
10
3
8
4
7
1
6
2
5
Controller
Configuration Registers
Permutation Generator
Stage 1 Self-Config. Controller
Spread Detect
Address Re-Map
State Sequencer
Noise Averaging Counter
Average Data
Address used to configure the capacitor array
Stage 1
Back-End ADC
Analog mismatch error of stage
Mismatch error of stage digitized by the back-end pipelined stages

Figure 4.10: Control logic: Stage self-configuration engine.

The standard cell library available for this project was not very well characterized. Setup violations in flip-flops and wiring delay problems are avoided by designing the logic to work up to 100MHz clock frequency. Hold violations are avoided by using both clock edges to handle data. Data launched on one clock edge is captured on the opposite clock edge. A transaction-based architecture is used to connect the different modules. A module sends a request to the next module along with the data required and waits for an acknowledge signal which is exerted on completion of the command/request upon which the result is transferred back to the requesting module.

The “Controller” is the master block that directs the self-configuration process. This initializes the digital sections of the chip and insures that the stages that resolve the lesser significant bits are self-configured before the stages that resolve the more significant bits. This ensures that the errors of the subsequent stages are corrected before these stages are used to accurately determine the errors of the more significant stages. The “Configuration Registers” stores the state of the self-configuration algorithm including information regarding the spread of the configuration, the best spread and the best detected configuration. This information can be read back through a serial interface. The “Self-Configuration Controller” controls the self-configuration flow for a stage being configured. Upon receiving a request from the controller, this block requests the “Permutation Generator” for a new permuted configuration and also requests the “State Sequencer” to begin a new measurement sequence as described in Fig. 3.10 and Section 3.2.1. The effect of noise in the measurement of the error can be mitigated by repeating the same measurement several times, this is controlled by the “Noise Averaging Counter”, and the repeatedly measured mismatch error information is averaged by the “Average Data” block. The “Spread Detect” block measures the spread in the capacitors for the chosen configuration at the end of a measurement sequence and informs the controller if a better configuration is found. The self-configuration process continues either until a configuration with zero spread is found or if the number of measurement cycles exceeds the time-out value programmed in the configuration registers.

4.3.1: Permutation generator

The “Permutation Generator” is a key component in the system. The array configuration for each trial is initially determined using this block, which shuffles the allowable set of address words to form a unique grouping of sub-elements. A 2-b word as described in Section 3.2.1 is used to assign each capacitor sub-element to one of the four possible DAC capacitor groups. The effectiveness of a permutation generator implementation can be quantified using the state repetition ratio (SRR), defined as ratio of the total number of configuration states generated so far to the number of unique configuration states generated.

Fig. 4.11 shows the case where two and four elements are permuted using unit swap blocks. Using unit swaps, two elements can be permuted using one swap block controlled by one bit. There are 2! i.e. 2 unique states and the control bit can take 2^1 i.e. 2 control states. In this case, the state repetition ratio is $2^1/2!$ i.e. 1.0. Permuting four elements using unit swaps uses six swap blocks with six control bits with 2^6 i.e. 64 control states. However, four elements can be configured in 4! i.e. 24 unique states i.e. the state repetition ratio is $2^6/4!$ i.e. 2.6. It is obvious that using combinatorial logic to permute many elements does not scale well with an increase in the number of elements. States repeat often and it is difficult to generate unique permutations.

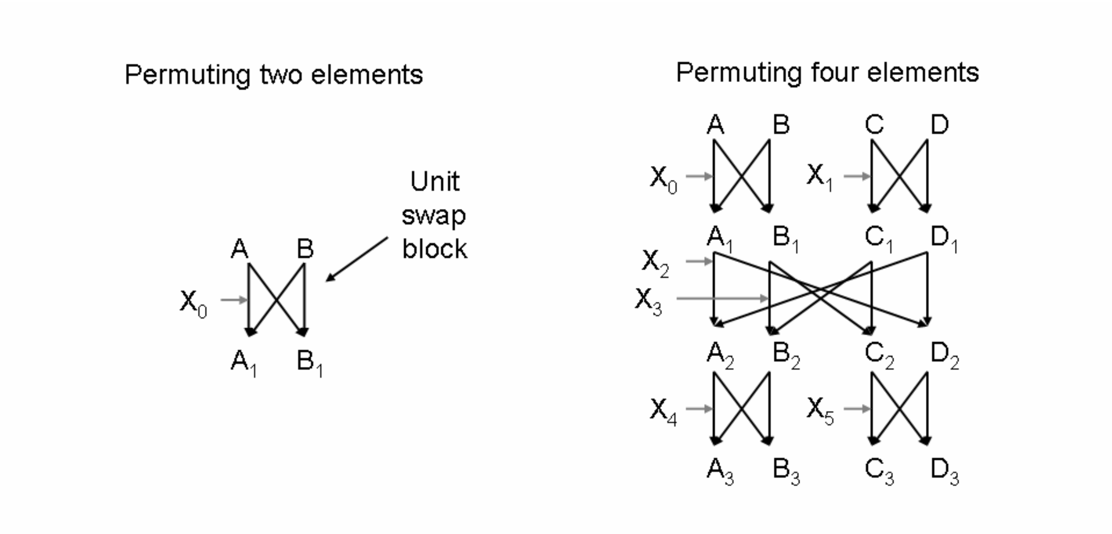


Figure 4.11: Permuting using unit swap blocks.

Permuting large number of elements using existing techniques based on unit swaps become very complicated [99] and simplifications like butterfly shufflers and barrel shifters cover only a small deterministic subset of the possible permutation space [100]. A state machine approach is taken, where the state transitions are derived from elementary transformations of elements moved on the surface of a cube as shown in Fig. 4.12. Let the elements (1, 2 … 24) that are permuted occupy positions on the surface of a cube. This forms a known valid configuration. Define six paths on the surface of the cube (e.g.: 9 → 10; 10 → 12; 12 → 11; 11 → 9) and three paths along the volume (e.g.: 1→ 2; 2 → 5; 5 → 6 etc). One of these nine paths is chosen by the output of a pseudo-random word generator along which the elements move to form a new permutation. The example in Fig. 4.14 shows how an initial state representing a valid permutation is transformed by a volume followed by a surface transition. Each of the three cases shown represents a distinct overall arrangement of the elements representing unique permutation configurations. This 24-element case is easy to visualize, but 32 or 16 elements can be also efficiently permuted with simple logic circuits. Each cell contains one of the four 2-bit addresses {0,0} … {1,1} and each address is repeated eight or four times (for the 32 and the 16 sub-element cases). Each cell position is mapped to one of the sub-elements. Appropriately permuting these addresses is equivalent to permuting and grouping the sub-elements in various ways.

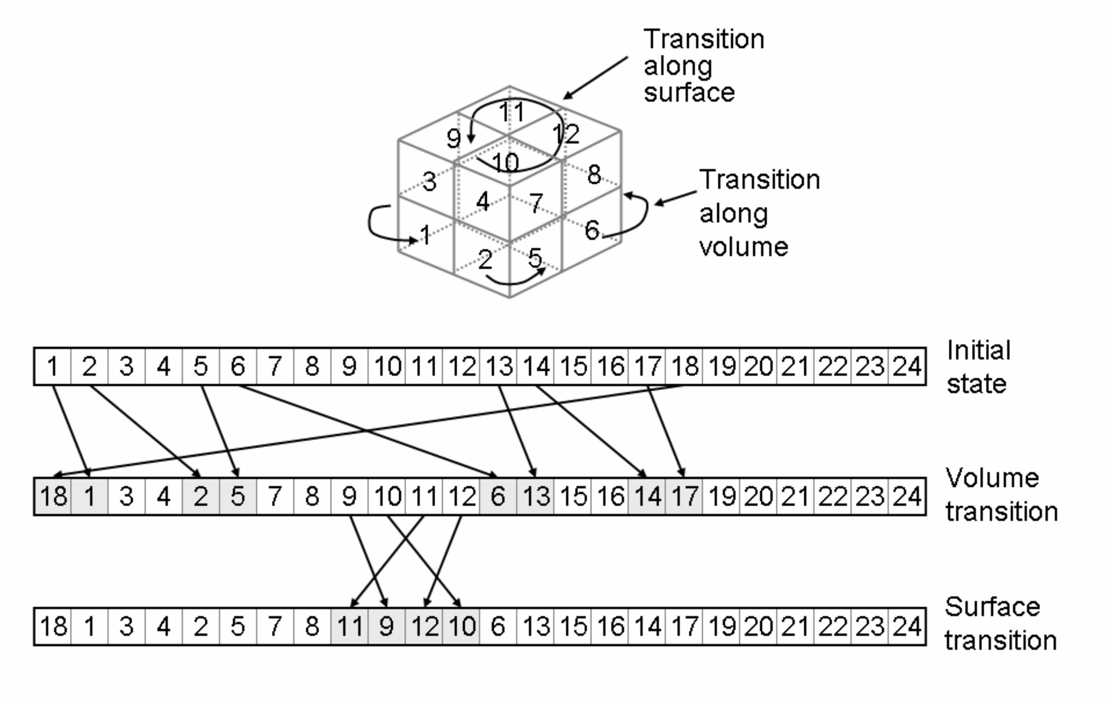


Figure 4.12: Operating concept underlying the permutation generator for a 24 element case showing examples of volume and surface transitions.

An ideal permutation generator has a state repetition ratio of 1.0. Fig. 4.13 shows that the proposed technique is effective in generating near unique permutations once every state machine event using simple transition rules.

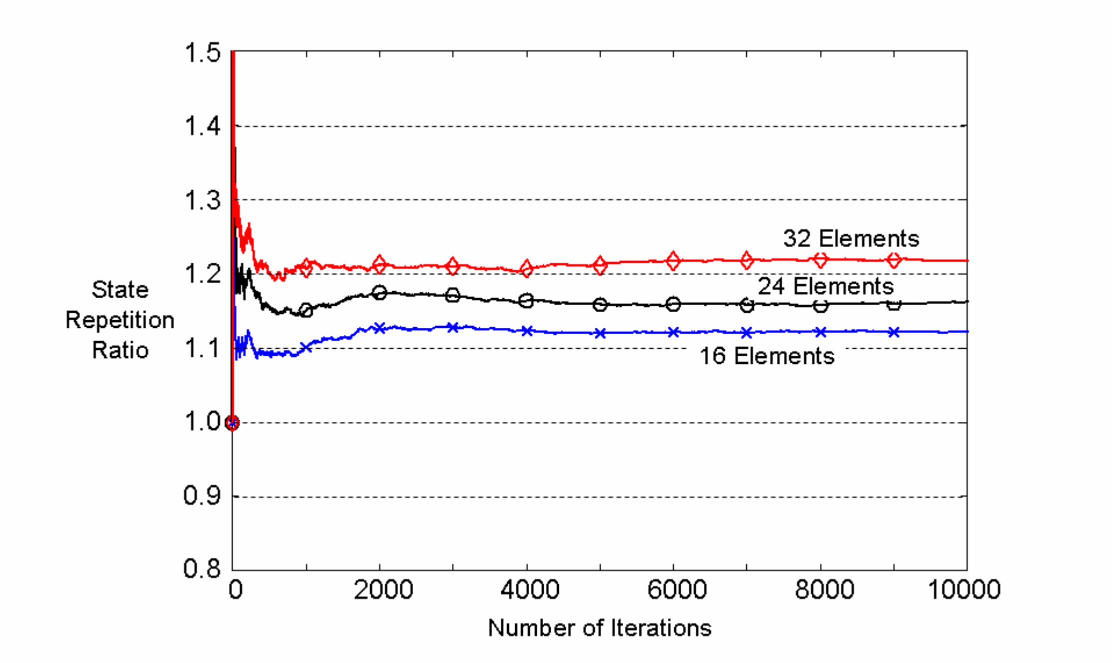


Figure 4.13: Simulation of the permutation generator showing state repetition ratio vs. number of iterations.

4.3.2: Address re-map

The permutation generator permutes and groups the capacitor sub-elements. The exact function of each of these groups is assigned by the “Address Re-Map” block. This block assigns functionality to these grouped sub-elements to form the feedback or the DAC capacitors. That is, sub-elements with address {0,0} correspond to the feedback capacitor, those with address {0,1} corresponds to the first DAC capacitor, etc. Two examples of the address re-mapping process are shown in Fig. 4.14. The example shows one of the capacitor groups being configured as a feedback capacitor or the first DAC capacitor depending on the remapping rule. This block allows the switched capacitor network topology to be changed easily as required in order to measure the relative value of the grouped DAC capacitors. The permutation generator and the address re-map block together allow the sub-elements to be permuted, logically grouped and assigned a function in the MDAC array structure.

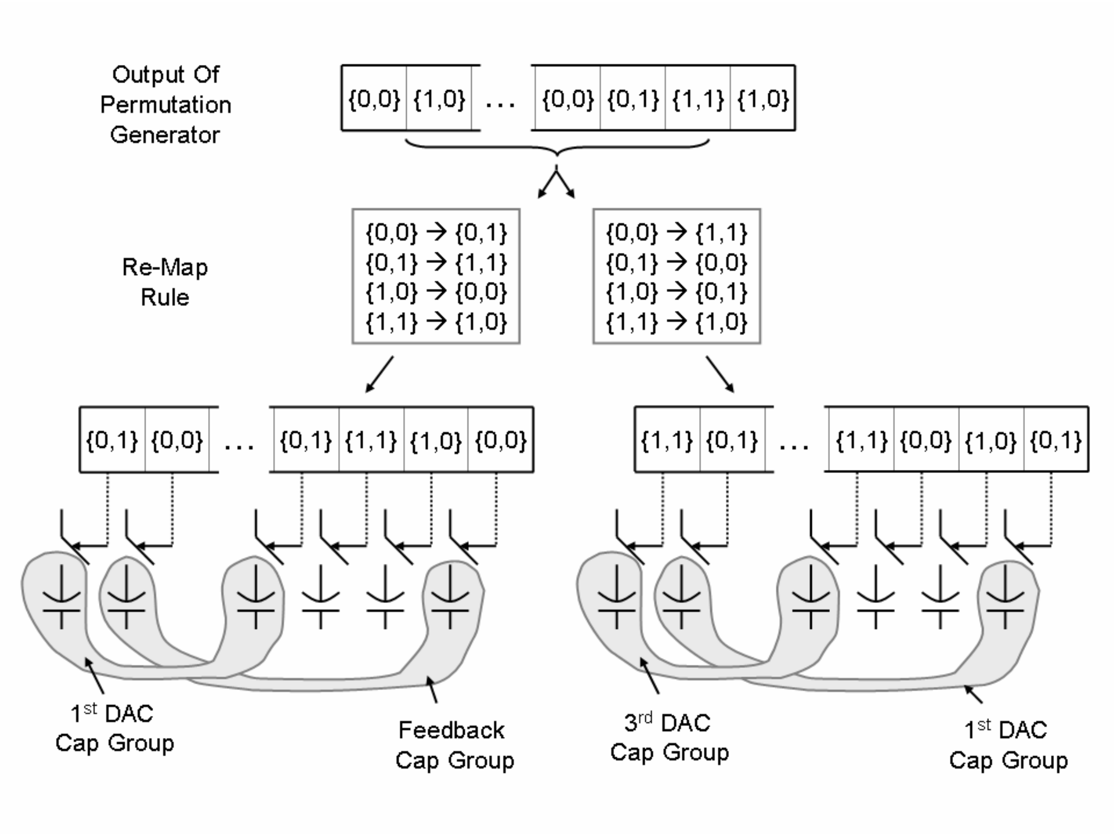


Figure 4.14: Change of topology and capacitor functionality by the address re-map block.

4.4: EXPERIMENTAL RESULTS

The prototype ADC with the self-configured capacitor matching algorithm is fabricated in a 0.18μm digital CMOS process. The chip shown in Fig. 4.15 occupies 3.6mm^2, and consumes 168/100mW (analog/digital) at 40MS/s with 1.8V supply. The area overhead for the self-configuration system is the wiring necessary to support the capacitor re-configuration procedure. This interconnection between the digital DAC decoder and the switches occupies around 5% of the total analog area. The algorithm is completely implemented on the chip using synthesized logic.

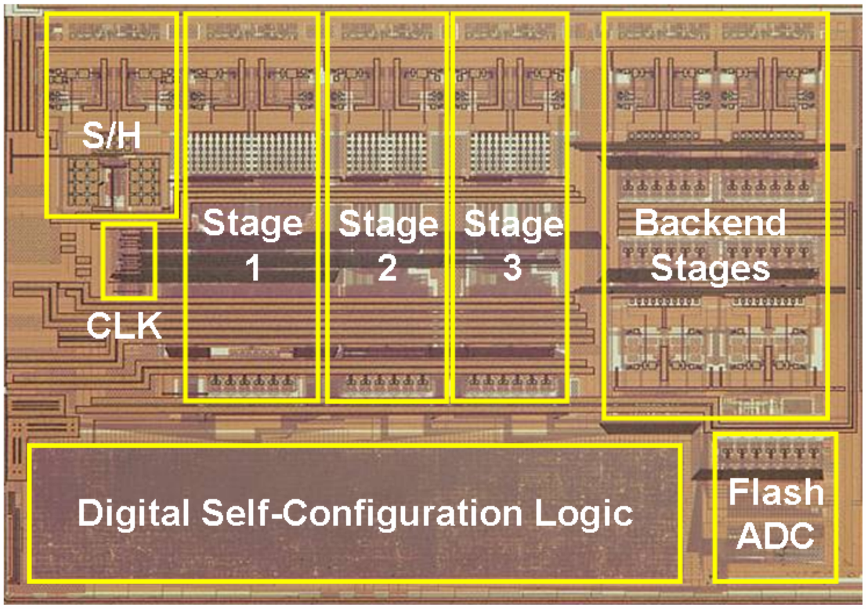


Figure 4.15: Chip die photo.

The chip is directly mounted on a printed circuit board (PCB) on a soft gold contact and protected using optically transparent epoxy held in place using a dam and fill arrangement. Gold wires are used to bond the pads to the contacts on the PCB. The PCB, shown in Fig. 4.16, is fabricated using 0.062” thick FR-4 dielectric and six copper layers with an immersion gold finish. Two layers are used for routing signals and four layers are used for routing power supplies. The digital and analog power supplies are kept separate and the grounds are tied together at the voltage regulators. Input signals and clock are connected using SMA connectors terminated using 50Ω resistors. The sinusoidal input required to test the ADC is derived from a bench top signal source and filtered using band-pass filters. This is a single ended signal and this is converted to a differential signal required to drive the ADC using either an amplifier (Analog Devices AD8138) or a transformer (Mini-Circuit T1-1T) depending on the desired input frequency. Output bits are latched using D flip-flops and are recorded using a logic analyzer and analyzed using software running on a PC.

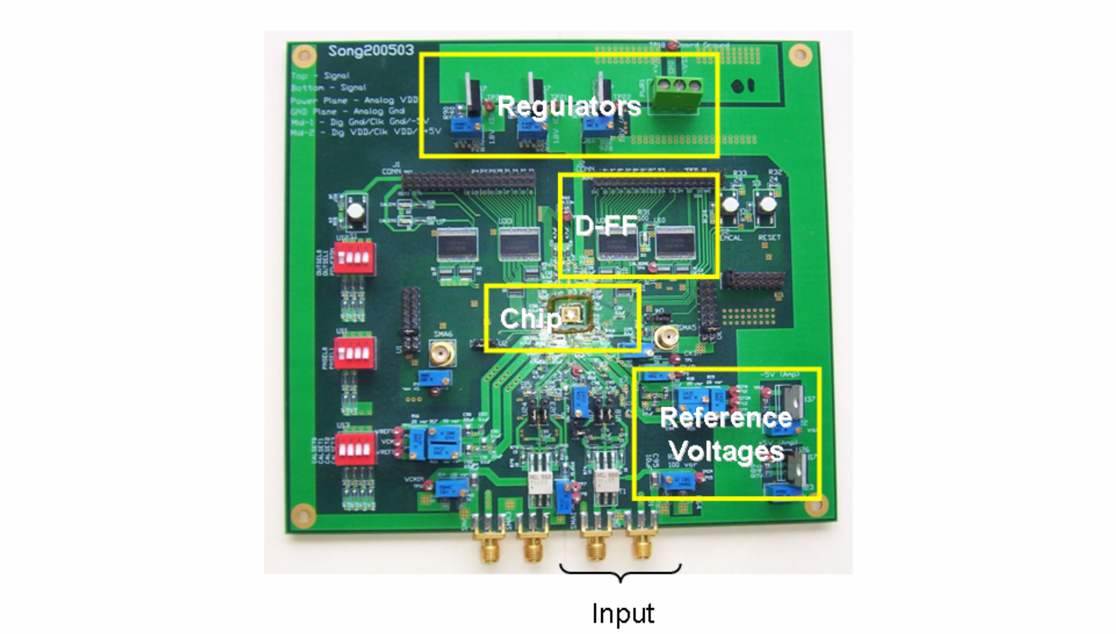


Figure 4.16: Test board.

After configured for low spread, capacitor matching is visibly improved in the INL measurement from 3 LSB to 0.8 LSB as shown in Fig. 4.17. The residual error seen in the INL plot is thought to be due to op-amp DC gain that was lower than designed. If the DC gain term is ignored or removed using digital post processing, then the residual error due to capacitor matching is within ½ LSB. The measured DNL is shown in Fig. 4.18.

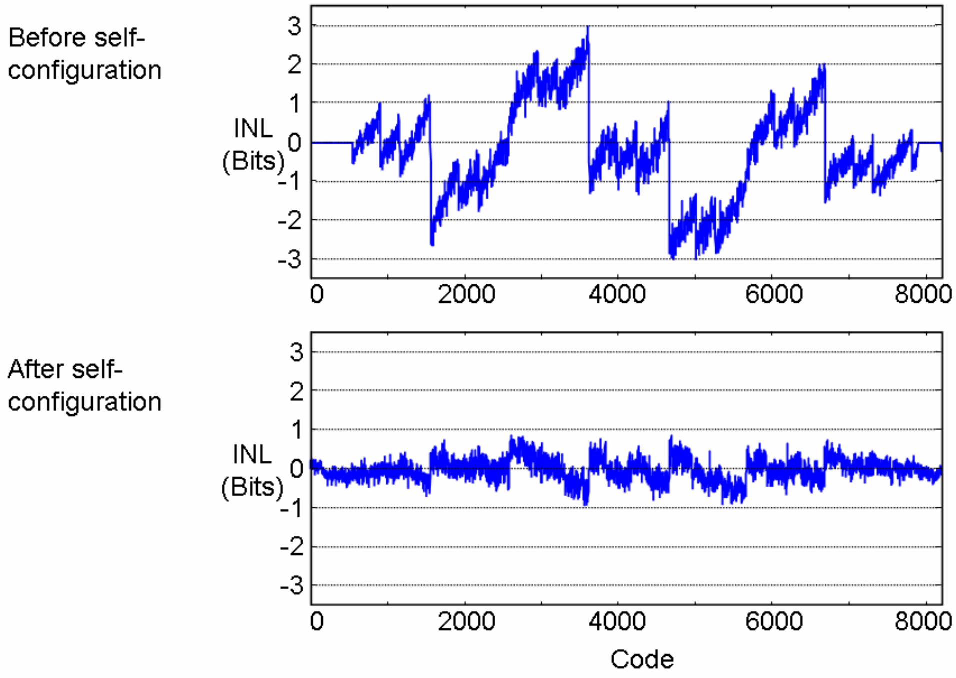


Figure 4.17: Measured INL at 13b level before and after self-configured for low spread.

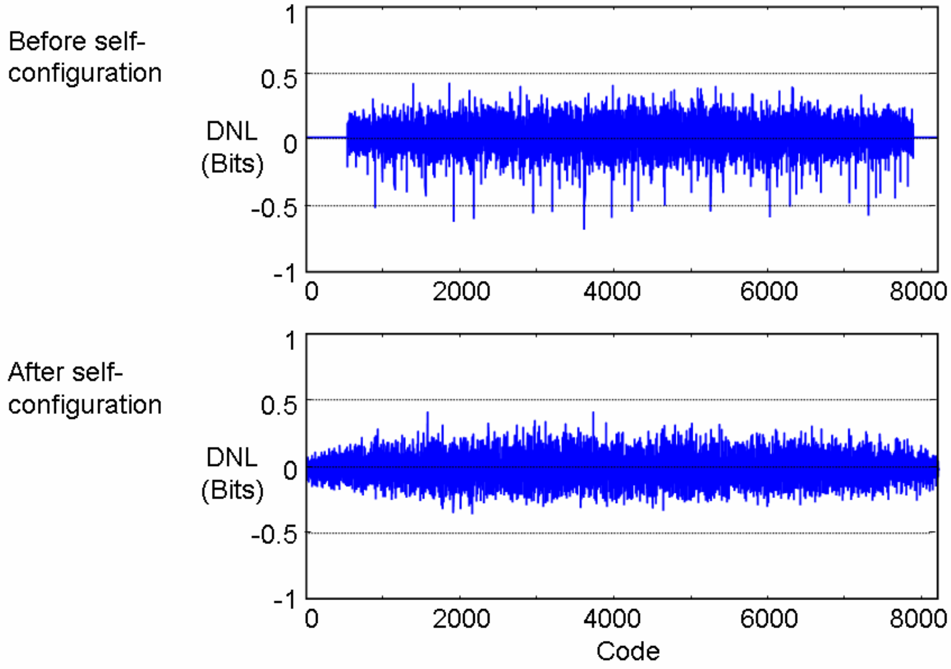


Figure 4.18: Measured DNL at 13b level before and after self-configured for low spread.

The improvement in the linearity is reflected in the FFT of a 1MHz input sampled at 14MS/s before and after self-configuration as shown in Fig. 4.19. The measured SFDR is improved from 70 dB to 89 dB. With a 14 MHz input sampled at 43 MS/s, measured SFDR is also improved from around 70 dB to better than 80 dB after self-configuration as shown in Fig. 4.20. The performance metrics (SFDR, THD and SNDR) after self-configuration is plotted vs. input signal frequency in Fig. 4.20 for 14 MS/s and 43 MS/s sampling rates. Fig. 4.21 shows the performance metrics after self-configuration vs. sampling rate for 1 MHz and 14 MHz signal frequency. The low SNDR is thought to be because of excess thermal noise contributed by the operational amplifiers due to a design error. The measured performance is summarized in Table 4.2. The contents of the configuration registers could be read using serial port functionality on the chip. The capacitor-array MDACs in the 1st, 2nd and 3rd stages exhibit linearity of 5, 2, and 1 LSBs at 16-b level, respectively, after 5018, 1285 and 5122 random trials. It takes about 2 seconds to configure the three MDAC stages for low spread.

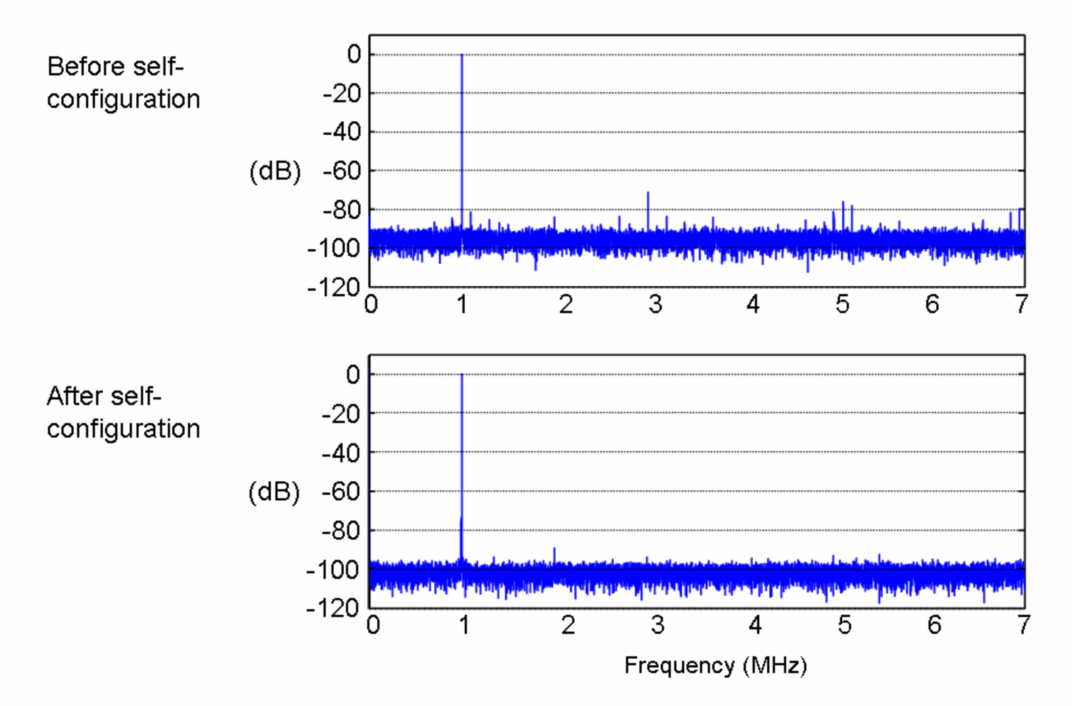


Figure 4.19: 1 MHz sampled at 14 MS/s before and after self-configured for low spread.

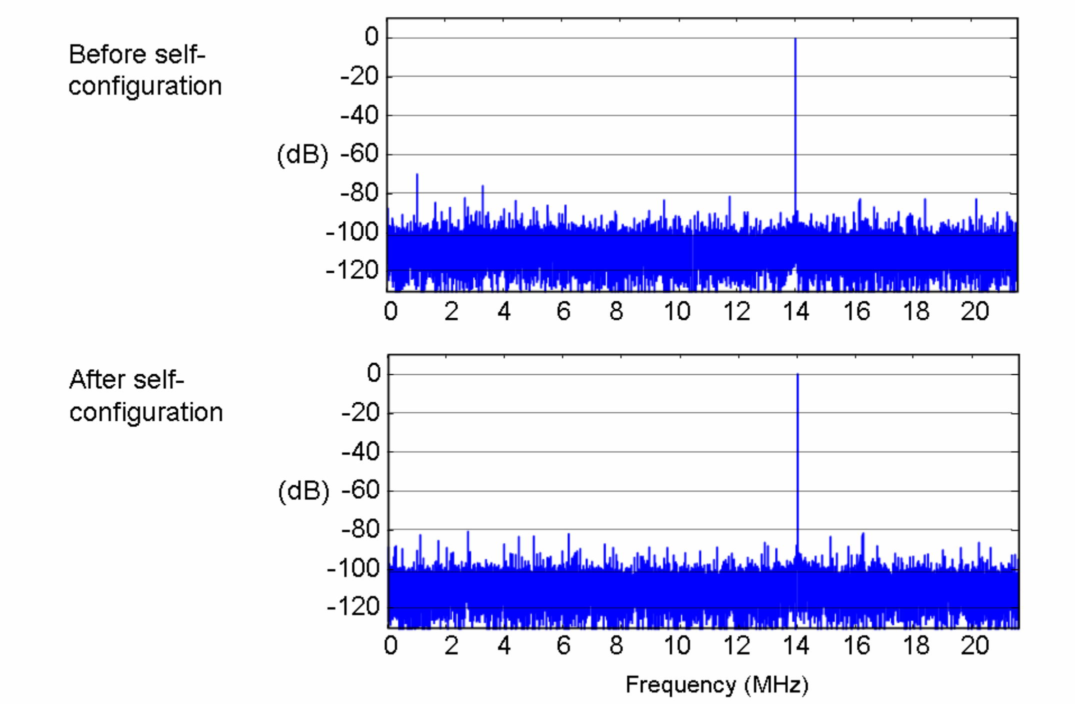


Figure 4.20: 14 MHz sampled at 43 MS/s before and after self-configured for low spread.

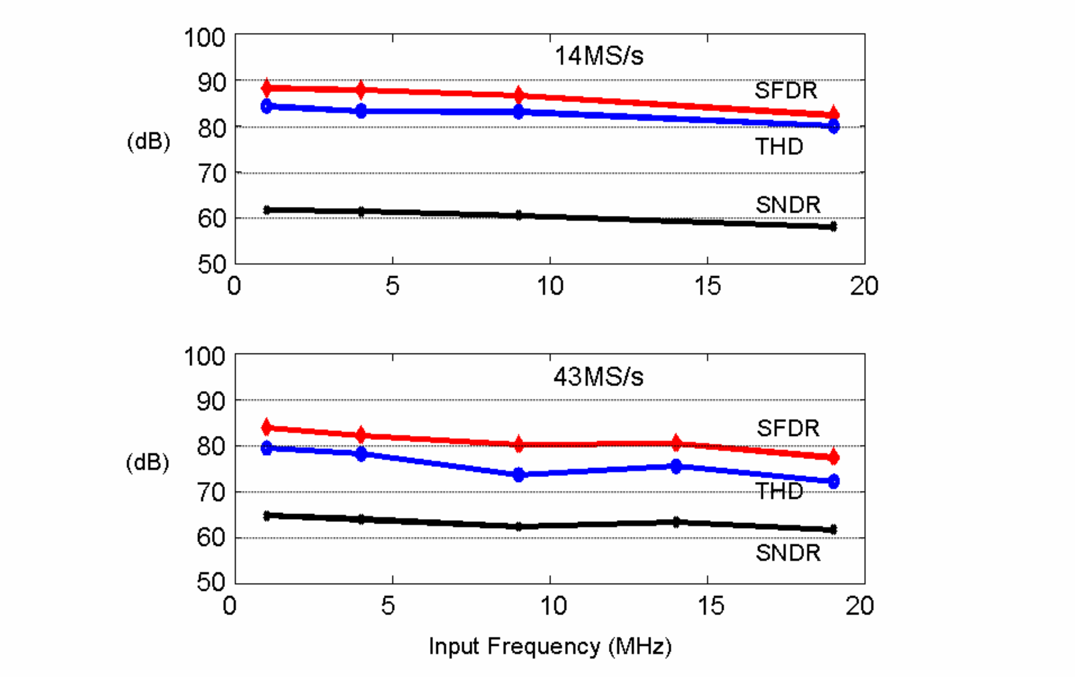


Figure 4.21: SFDR, THD and SNDR vs. input frequency at 14 and 43 MS/s after self-configuration.

Table 4.2: Summary of Measured Performance

Technology	0.18μ CMOS
Resolution	13b
Conversion Rate	40MS/s
Self-Configuration Time	2sec
Supply Voltage	1.8V
Input Signal Range	Differential 1.6V_{PP}
DNL/INL	0.4 / 0.8LSB
SFDR	> 80dB @43MS/s > 89dB @14MS/s
THD	-73dB
Peak SNDR	67dB
Active Area	3.6mm^2
Power @ 1.8V (Analog/Digital)	268mW (168/100mW)

4.5: CONCLUSIONS AND FURTHER WORK

An ADC system is designed with the self-configuration algorithm contained in the same die in a standard digital CMOS process with M-I-M capacitors and improvement in capacitor matching is demonstrated after the self-configuration process. A permutation generator structure that allows near unique permutations of the sub-elements to be generated is proposed to implement the random search algorithm in a hardware efficient manner. An optimization technique based on the unit amplifier concept is proposed to design Miller compensated operational amplifiers.

Future work should focus on an improved switched scheme that simplifies the layout. A background calibration scheme can be implemented to self-configure the capacitors. The operational amplifiers should be designed to be more robust than in the present chip. In addition, a background calibration scheme to address the finite gain errors in the operational amplifier can also be implemented to improve the performance of the ADC.

Parts of this chapter have been published in S. Ray, Bang-Sup Song, "A 13b linear 40MS/s pipelined ADC with self-configured capacitor matching," *ISSCC Dig. of Tech. Papers.* pp. 852-861, Feb. 2006 (© 2006 IEEE) and Sourja Ray; Bang-Sup Song, "A 13-b Linear, 40-MS/s Pipelined ADC With Self-Configured Capacitor Matching," *IEEE J. Solid-State Circuits*, vol. 42, no.3, pp.463-474, Mar. 2007 (© 2007 IEEE) with the dissertation author as the primary author of these papers. The relevant sections have been reprinted with permission of the IEEE.

Chapter 5:

Calibration of the multi-bit current DAC used in a continuous-time delta-sigma ADC

This chapter describes an analog domain technique to calibrate current source DAC arrays. A discussion of common techniques of calibration of delta-sigma ADCs is followed by a description of this calibration technique applied towards the calibration of the DAC in a continuous-time delta-sigma ADC. Architecture choices and simulation results are discussed.

5.1: DELTA-SIGMA ADCS WITH MULTI-BIT QUANTIZERS

A summary of the working principles of delta-sigma ADCs is described in Section 1.5.4 in this dissertation. Delta-sigma ADCs with multi-bit quantizers have better stability properties than those with single-bit quantizers. However, the use of multi-bit quantizers implies that a multi-bit DAC is used in the feedback path. Mismatches in the unit elements of the feedback DAC affect the SNR and limit the achievable performance. Fig. 5.1 shows the output spectrum from a simulation of a delta-sigma ADC with a fourth order loop filter without and with mismatches in the feedback DAC. It is observed that mismatches lead to spurs in the spectrum. It is therefore necessary to mitigate the effects of the mismatches to improve the fidelity of the data conversion process.

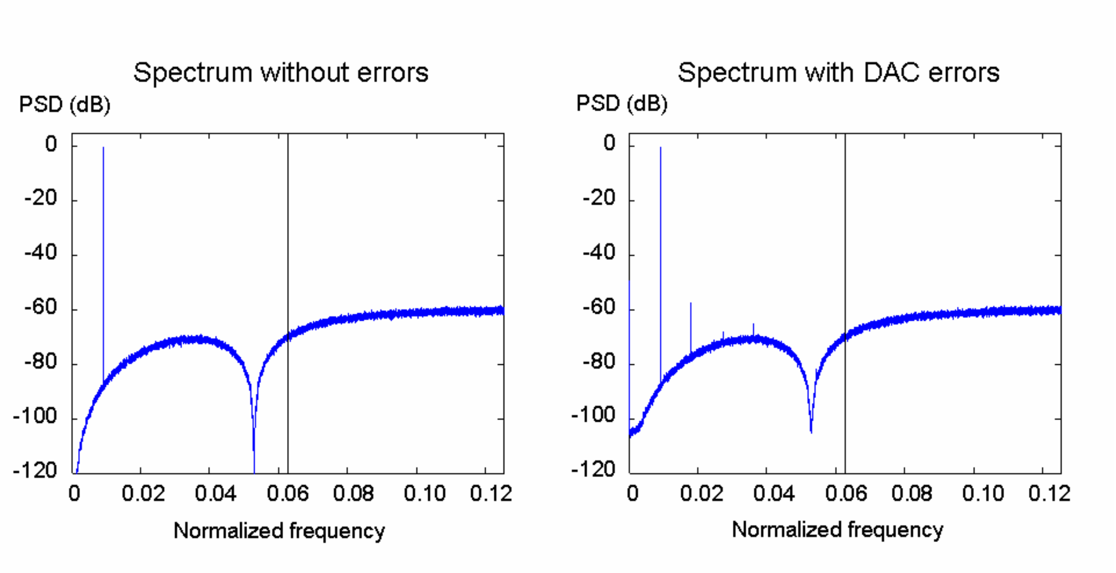


Figure 5.1: Spectrum of a delta-sigma ADC without and with mismatches in the feedback DAC

5.1.1: Ways to alleviate the effects of DAC mismatch errors

Several techniques have been proposed to alleviate the effects of these mismatches. A technique proposed by Naiknaware et. al. [101] uses a VCO and an accumulator as a multi-bit quantizer and the feedback DAC is a single path that employs a digital frequency difference detector and a low pass filter to emulate a multi-bit DAC. This architecture demonstrates the improvement of SNR as expected in a multi-bit delta-sigma ADC without requiring calibration or dynamic element matching in the feedback DAC. However, this technique is difficult to implement requiring complicated analysis and is very sensitive to non-linear effects in the VCO. Most multi-bit delta-sigma ADC or DAC architectures, thus, usually rely on some form of correction mechanism.

The oversampling nature of the delta-sigma ADCs makes it attractive to use dynamic element matching [33] (DEM) or mismatch shaping. In these schemes, the error due to DAC mismatch is modulated in a way such that the error is spread either uniformly over the frequency domain or in a frequency selective manner in a way to move the error related power out of band that optionally can also be filtered [102], [103], [104]. DEM can be done in a data dependent manner or using randomization. The most significant limitation of perfect randomization is that this requires an arbitrary connection switch box between the output of the flash and the input of the DAC. This design is not simple [105] - barrel shifters and butterfly randomizers have been proposed in order to simplify this requirement. The arbitrary connection switch box used in the ADC described in Chapter 4 had a complicated physical design and it is necessary either to design a simpler structure or to avoid using this. These schemes become progressively more difficult to implement as the number of elements in the DAC increase. The DEM scheme should be designed to convert the error due to mismatches to out-of-band noise since the loop filter does not shape the DAC noise. DEM techniques by themselves do not correct for the mismatch. Digital and analog calibration techniques have been proposed to compensate or correct for the errors due to mismatches in the feedback DAC and are frequently implemented with DEM as part of the overall system.

The digital group of techniques compensates for the analog mismatches by measuring the errors and storing a correction term in digital registers. These values are used during the conversion process to mitigate the effects of the errors. The presence of the loop filter makes this technique complicated to implement as compared to implementations in Nyquist ADCs. In [106], the authors excite each unit DAC element in turn using an out-of-band digital signal and the errors in the unit elements are estimated by detecting this calibration signal in the digital domain using correlation. This permits the ADC to be calibrated without interrupting normal conversion cycles and the correction is performed digitally. However, the dynamic range of the circuits decreases due to the signal added by the extra element. This is suitable when the oversampling ratio is large and the tone can be placed adequately away from the signal bandwidth and with the desired noise transfer function shape to improve calibration convergence. The analog group of techniques correct for the mismatches in the analog domain. The relative error between the elements is measured and a suitable correction term is applied that improves the matching between the DAC elements [107]. Both analog and digital techniques can be implemented either using offline (foreground) or online (background) error measurement schemes.

5.1.2: Continuous-time delta-sigma ADCs

Loop filters designed using switched capacitor circuits have inherently high coefficient accuracy and stability across environment changes since the coefficient values are determined by ratio of capacitors. Filter design for discrete-time delta-sigma ADCs using discrete-time transfer functions is a mature and well-understood technique. These factors made the discrete-time implementation of delta-sigma ADCs popular. The filters are typically implemented using cascaded integrators. The circuits with the most stringent design requirements in these ADCs are the front-end sampling network and the first integrator in the loop filter. The use of continuous-time filters in the loop filter has become more popular recently because these designs have significantly better power efficiency [108]. It is possible to have higher bandwidths in continuous-time filters as compared to discrete-time filters for a given amount of power consumption. This has made continuous-time delta-sigma ADCs (Fig. 5.2) a strong contender for high bandwidth applications like broadband communication receivers that are traditionally implemented using pipelined ADCs.

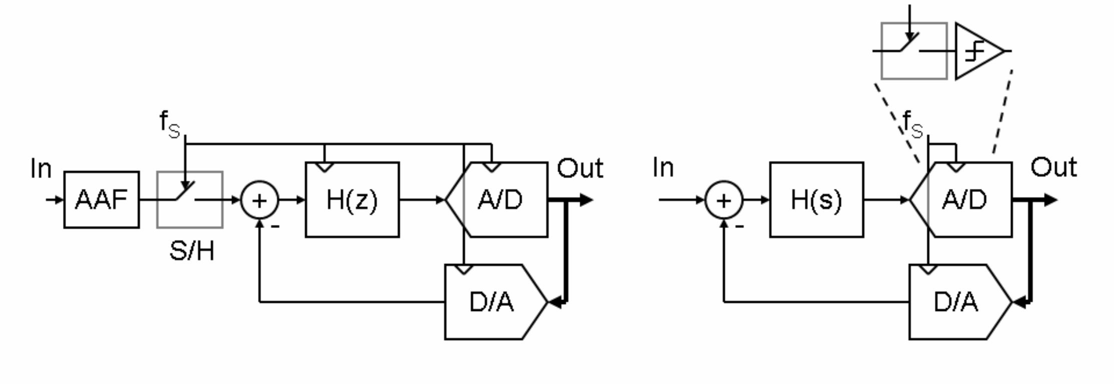


Figure 5.2: Discrete-time and continuous-time delta-sigma ADCs

Additional advantages are realized in continuous-time delta-sigma ADCs because the sampling takes place at the quantizer. The loop filter also performs the anti-aliasing function leading to a more efficient system level implementation. The linearity and accuracy requirements of the sampling operation is relaxed since the forward path gain of the loop filter and enclosing the quantizer in the feedback loop cause the sampling error to be reduced and noise shaped. The coefficients are implemented using Gm/C or active R-C filters. Several techniques have been proposed to compensate the variations in the coefficient values [109], [110]. The advantages of the continuous-time delta-sigma ADC architecture are realized with the added penalty of the system stability and performance being sensitive to the complete shape of the waveforms at various internal nodes especially at the output of the feedback DAC. Metastability in the quantizer [111] causes the comparators to have a signal dependent delay in the decision. The output of a comparator in the quantizer is delayed by a larger amount when the input is close to the switching threshold as compared to when the input is large compared to the threshold value thus causing a modulation in the feedback value. This causes the SNR to degrade and fills up the notches in the noise transfer function. The preferred solution is to add half a clock cycle of delay before applying the feedback to the DAC and the stability of the loop transfer function is insured by using a local feedback around the quantizer [112]. This reduces the probability of errors related to metastability to the extent that this is a problem only if the delay is more than half a cycle. Commonly, a non-return to zero (NRZ) DAC is used in delta-sigma ADCs. Unequal rise and fall times in the DAC control signal introduces a signal dependent feedback value [113] that affects the performance of the delta-sigma. Several solutions have been proposed like the use of return to zero (RTZ) [114] to equalize the number of rise and fall transitions per clock cycle. The jitter in the clock signal does not affect the quantizer timing instant but affects the DAC timing by introducing additional noise in the feedback path. RTZ DACs have two edges per clock cycle that increase the total amount of noise injection. NRZ DACs are used if the jitter-induced noise makes the overall noise performance difficult to achieve.

Continuous-time delta-sigma ADCs used for wide-band quantization need bandwidths of the order of 20 - 40 MHz where the maximum clock speed, limited by the process technology parameters like f_T, is of the order of 400MHz - 800MHz. This implies that the OSR is low i.e. 8 to 16 times. A multi-bit quantizer with a high order loop is essential to obtain the necessary SNR.

5.2: REDUCTION OF MISMATCH ERRORS IN CONTINUOUS-TIME DELTA-SIGMA ADC USING ANALOG BACKGROUND CALIBRATION

This chapter describes an analog background calibration scheme for a switched current feedback DAC in a continuous-time delta-sigma ADC. The feedback DAC has an extra unit element that replaces one of the main DAC elements during calibration cycles in a way similar to [68]. The contribution in this work is the extension of this scheme to work with continuous-time delta-sigma ADCs at high clock frequencies. The continuous-time delta-sigma architecture requires a clean way of retiring an element for calibration and returning it to the main signal path. Circuit techniques are described to perform this exchange synchronously with the system clock, with low latency and without adding additional switching noise or glitches. This scheme is demonstrated with simulations using TSMC 0.18μm CMOS process models.

5.2.1: The feedback DAC

The target system is a continuous-time delta-sigma ADC with 450MHz clock frequency with a four-bit coarse quantization flash ADC. This flash ADC has 15 comparators and the output of this is connected to a four-bit current steering DAC in the feedback path. Such a DAC would normally use 15 unit current sources. The outputs of the comparators are latched half a cycle [69] after the comparators are triggered in order to reduce errors due to data dependent delay in the comparator. A spare unit current source to calibrate the unit DAC element current sources is added as described by Groeneveld et. al. [68]. The principle of the calibration scheme is shown in Fig. 5.3.

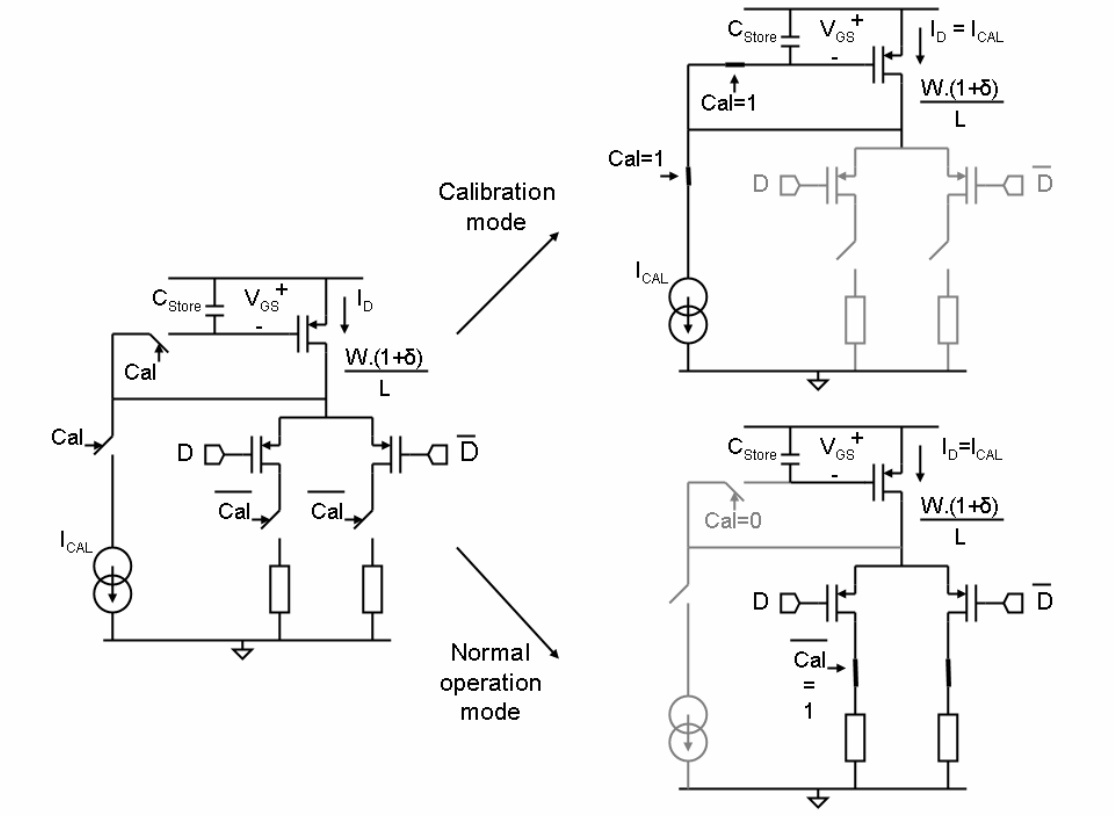


Figure 5.3: Calibration principle.

Each DAC element can be connected either in the normal operation mode i.e. the switches are activated by the decision from the coarse ADC or in the calibration mode, i.e. the DAC element is not a part of the delta-sigma feedback system and is retired from the main path. The spare current source replaces any current source that is being calibrated.

Several techniques have been described to perform this replacement. The original scheme [68] uses an additional analog switch in series with the DAC switch to retire or insert the spare unit DAC element and is applied towards the calibration of a high-resolution DAC. Falakshahi et. al. [70] describe a delta-sigma DAC in which the spare unit DAC element replaces the unit DAC element being calibrated by routing the digital bit away from the element being calibrated towards the spare element. A switching scheme using only PMOS devices is used to obtain the necessary cross-over point for the DAC control signals. Yan and Sanchez-Sinencio [71] use a similar digital path switching scheme in a continuous-time delta-sigma ADC where the importance of synchronously switching in the replacement current source becomes apparent. The DAC switching signals are synchronously switched followed by a high crossing switch driver to provide switching signals needed for an NMOS current source DAC. Li et. al. [72] describe a continuous-time delta-sigma ADC with a similar calibration scheme and with a simpler scheme to reduce the effects of charge injection during the transition from calibration to the normal operation phase. All the schemes described have a disadvantage that since the spare DAC element is used to replace the DAC element being calibrated, the number of switching paths that connect to the spare element is equal to the number of elements in the DAC. This number grows exponentially with the resolution of the DAC. This increases routing complexity and increases the parasitic capacitance due to the switching network.

The work being described in this chapter is applied towards a continuous-time delta-sigma with a 450MHz sampling rate where the available switching time is extremely short. The output of the quantizer has to be presented at the input of the correct DAC elements within half a clock cycle i.e. within 1.11ns. Reducing the complexity of the switching system is a key factor in order to achieve such high clock speeds and this is achieved by eliminating the complex routing to the dedicated spare element. Each flash comparator output maps to one of two adjacent unit DAC elements similar to the method described in [106] i.e. each DAC element behaves like the spare for the adjacent DAC element. At the most two switches are in the path of each element and this reduces the complexity and simplifies the physical routing of the scheme. The element being calibrated is chosen and the swap is pre-computed and configured in the digital domain before the ADC decision is applied as in [71]. Custom S-R latches are designed that have an extremely low latency and with the required crossover point in the DAC control signals, as a result the swap takes place synchronously with the system clock and without additional glitches.

5.2.2: Calibration of a unit current source

The working principle of this scheme is that it is equivalent to state that if all the unit elements in a DAC are matched to a master element, then all the unit elements should match with each other. Each unit element has to be matched with the master element one at a time and this procedure can be performed when this DAC element is not used in the signal path. This procedure is further simplified in the current domain where a unit DAC element current source that uses PMOS transistors can be re-configured in the "diode configuration" i.e. in a feedback configuration while being driven by the master current source that is made using NMOS transistors as shown in Fig. 5.3. The feedback causes the gate to source voltage of the current source MOSFET to be driven to a value that supports the drain current as given by Eqn. 5.1. This scheme does not need additional components like a comparator that would have been necessary if this procedure was implemented in the capacitive/charge or the voltage domain. The unit DAC element has to be retired from the main signal path in order to be placed in calibration mode.

$$I_D = I_{CAL} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W \cdot (1+\delta)}{L} \cdot (V_{GS} - V_{Th})^2$$

$$V_{STORE} = V_{GS} = V_{Th} + \sqrt{\frac{2 \cdot I_{CAL}}{\mu \cdot C_{OX} \cdot \frac{W \cdot (1+\delta)}{L}}} \quad . \qquad \ldots \text{ Eqn. 5.1}$$

Eqn. 5.1 is the square law model of a MOSFET with width W with an effective error δ and channel length L and is valid for long channel devices. C_{OX} is the gate oxide capacitance per unit area; μ is the charge carrier mobility. I_D is the drain current for a gate to source voltage V_{GS}. V_{Th} is the threshold voltage of the transistor. Even though the MOSFET square law equation is not accurate for MOSFET with short channel lengths in deep-sub micron process nodes, the overall concept remains the same since this works by configuring the current source in feedback to obtain a gate to source voltage that supports the reference or the master current. Let the calibration current be equal to the expected value of the PMOS drain current when the current source transistor is error-free. Eqn. 5.2 shows that the stored voltage is a function of the current source error δ.

$$I_{CAL} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W}{L} \cdot \left(V_{GSNom} - V_{Th}\right)^2$$

$$V_{STORE} = V_{Th} + \frac{V_{GSNom} - V_{Th}}{\sqrt{1+\delta}} \approx V_{Th} + \left(V_{GSNom} - V_{Th}\right) \cdot \sqrt{1-\delta} \ . \qquad \text{… Eqn. 5.2}$$

At the end of the chosen calibration phase, the unit DAC element current source is taken out of calibration mode and used in the active signal path. The gate to source voltage that was set during calibration is held steady by preserving the charge on the storage node thus insuring that the output current matches the current that was forced by the calibration current source. This procedure is repeated in turn for all the current sources in the array and all the output currents should match with each other at the end of the calibration cycle. However, the charge storage node is not completely electrically isolated. The switch is implemented using a MOSFET and the drain / source implants form a reverse biased diode with the N-well (PMOS used as a switch).

$$I_{LEAK} = A \cdot \frac{q \cdot D_n n_i^2 \left(e^{\frac{q \cdot V}{k \cdot T}} - 1 \right)}{p_{p0} \cdot L_n \cdot \tanh\left(\frac{w}{L_n} \right)}$$

$$I_{LEAK} \approx -A \cdot \frac{q \cdot D_n n_i^2}{p_{p0} \cdot L_n \cdot \tanh\left(\frac{w}{L_n} \right)} \ . \qquad \text{… Eqn. 5.3}$$

$$I_{LEAK} = -A \cdot Const$$

$$T_{ERR} = \frac{C_{STORE}}{I_{LEAK}} \left(V_{GS2}(t=0) - V_{Th}\right) \cdot \left(1 - \sqrt{1 - 2^{-n}\left(\frac{I_{Fixed}}{I_{Variable}} + 1\right)}\right) . \qquad \ldots \text{ Eqn. 5.4}$$

The reverse bias current through this diode having an area *A*, expressed by Eqn. 5.3, will leak the electrons from the storage node causing the calibration charge to change and will cause an error in the output current comparable to the accuracy requirements of the DAC at time T_{ERR} expressed by Eqn. 5.4. This loss in the accuracy at the output of the DAC is prevented by repeating the calibration procedure on all the current sources in correct sequence. In this equation D_n is the electron diffusion constant, L_n is the electron diffusion length, n_i is the intrinsic carrier concentration, p_{p0} is the hole concentration in the P^+ diffusion, *w* is the distance of the ohmic contact from the junction. In reverse biased condition, the exponential term is very small as a result the leakage current just depends on the area of the junction and other semiconductor process related parameters.

The calibration system proposed for this system is shown in Fig. 5.4. Fig. 5.5 shows the case when one of the unit DAC elements is retired from the signal path and put in calibration mode. Using this scheme to divert the flash comparator decision away from the DAC element being calibrated allows a simpler analog switching network to be used thus minimizing the routing and the parasitic capacitance associated with a complicated switching network.

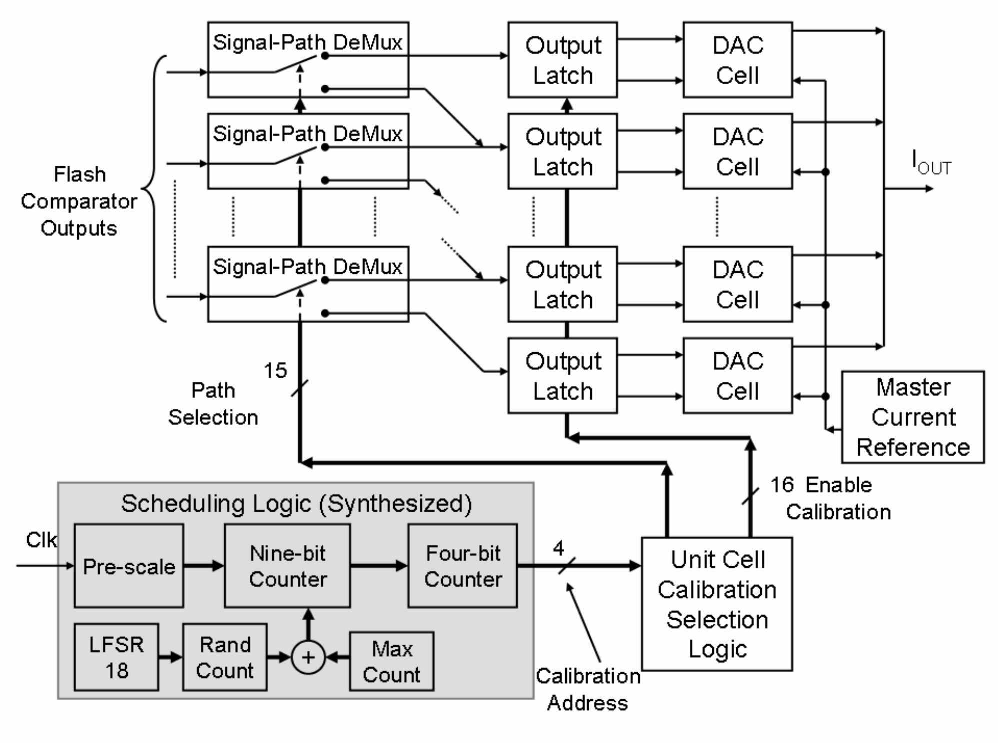


Figure 5.4: Calibration system

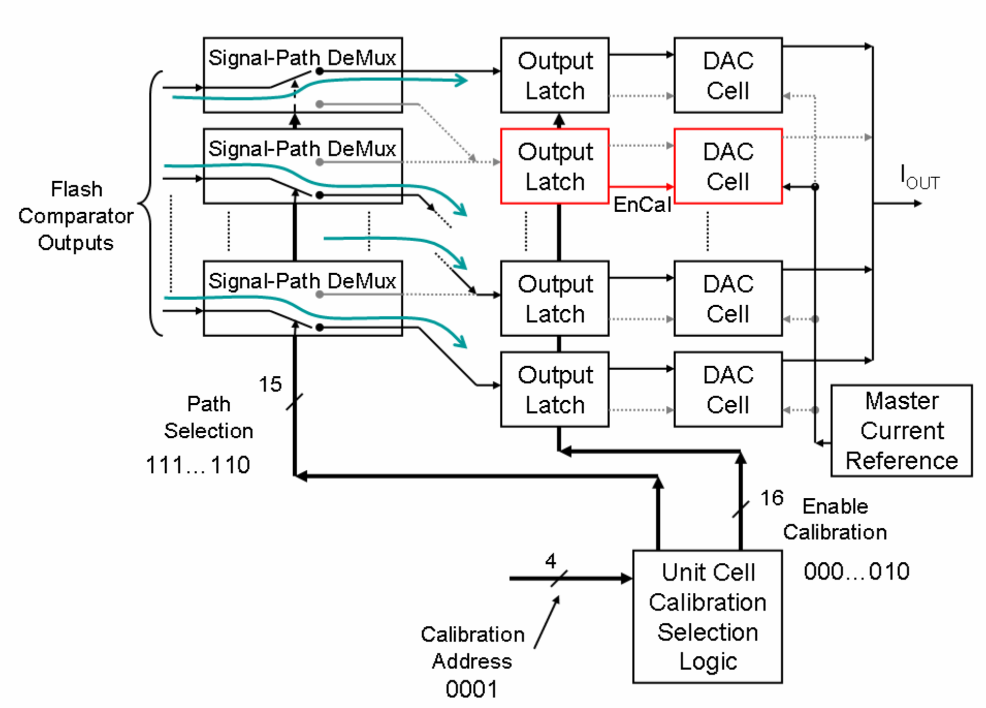


Figure 5.5: Calibration system showing DAC cell #1 being calibrated

5.2.3: Scheduling logic

The scheduling logic is a state machine controls the sequence and duration of the calibration process. Provision for freezing the calibration at a chosen time is provided. The “Pre-Scale” block divides the clock frequency f_{Clk} by a factor of “Scale” set as 4 i.e. the digital blocks are clocked at 450MHz/4 ~ 112.5MHz. This allows the rest of the blocks to be easily synthesized using the standard cell library available for this process.

The “Nine-bit Counter” counts the number of cycles that each current source is held in calibration mode and the maximum value “MaxCount” can be programmed during system start-up. An internal pseudo-random number generator “LFSR18” is based on an 18th order linear feedback shift register and can be used to pseudo-randomly pulse-width modulate the duration of calibration by changing the maximum count length by an amount “RandCount” that varies from -8 to +7. This prevents a fixed tone in the spectrum that might appear at the output due to the repetitive nature of the calibration [73]. A new pseudo-random number is generated at the end of each calibration cycle. The pseudo-random number generator can be switched off if desired. Eqn. 5.5 sets the actual time “T” that each current cell is held in calibration mode.

$$T = \frac{(MaxCount - RandCount)\cdot Scale}{f_{Clk}} \quad . \qquad \text{… Eqn. 5.5}$$

When this counter reaches the maximum count, the value in the “Four-bit Counter” is incremented. The output of this counter is a 4-b code that addresses the unit DAC element that is to be retired for calibration. The calibration scheduling logic is written in Verilog HDL and synthesized at 1.5 x 450MHz = 675MHz clock frequency. This is realized using approximately 8.6 k gates.

5.2.4: Design of unit cell calibration selection logic

The 4-bit output of the scheduling logic is decoded by the "Unit Cell Calibration Selection Logic" to form two sets of signals that are used to select the cells in the signal path and isolate the cell being calibrated. The "Path Selection" group of signals is decoded by the truth table shown in Table 5.1.

Table 5.1: Truth Table for the Path Selection Signal

A3	A2	A1	A0	#	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
0	0	0	0	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
0	0	0	1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0
0	0	1	0	**2**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0
0	0	1	1	**3**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0
0	1	0	0	**4**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0
0	1	0	1	**5**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0
0	1	1	0	**6**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0
0	1	1	1	**7**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0
1	0	0	0	**8**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0
1	0	0	1	**9**	**1**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0
1	0	1	0	**10**	**1**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0
1	0	1	1	**11**	**1**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	**12**	**1**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	**13**	**1**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	**14**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	**15**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These signals direct the flash comparator outputs away from the cell being calibrated and towards the cells that remain in the signal path. The "Enable Calibration" group of signals is decoded by the truth table shown in Table 5.2.

Table 5.2: Truth Table for the Enable Calibration Signal

A3	A2	A1	A0	#	P15	P14	P13	P12	P11	P10	P09	P08	P07	P06	P05	P04	P03	P02	P01	P00
0	0	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
0	0	0	1	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0
0	0	1	0	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0
0	0	1	1	**3**	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
0	1	0	0	**4**	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0
0	1	0	1	**5**	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0
0	1	1	0	**6**	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0
0	1	1	1	**7**	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0
1	0	0	0	**8**	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0
1	0	0	1	**9**	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
1	0	1	0	**10**	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0
1	0	1	1	**11**	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	**12**	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	**13**	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	**14**	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	**15**	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

These signals direct the calibration enable signal to the cell being calibrated. This is essentially a 4-bit to one-hot code decoder that performs the path selection. The logic is implemented using custom logic cells and complex gates that permit the digital outputs to be asserted with required rise and fall times and with small enough delay so that the logic outputs are ready before the next clock period. This permits the inputs to be routed quickly to the appropriate DAC switches. The maximum delay in the “Path Selection” path is less than 610ps and that in the “Enable Calibration” path is less than 760ps in the slow process corner as determined from simulations shown in Fig. 5.6.

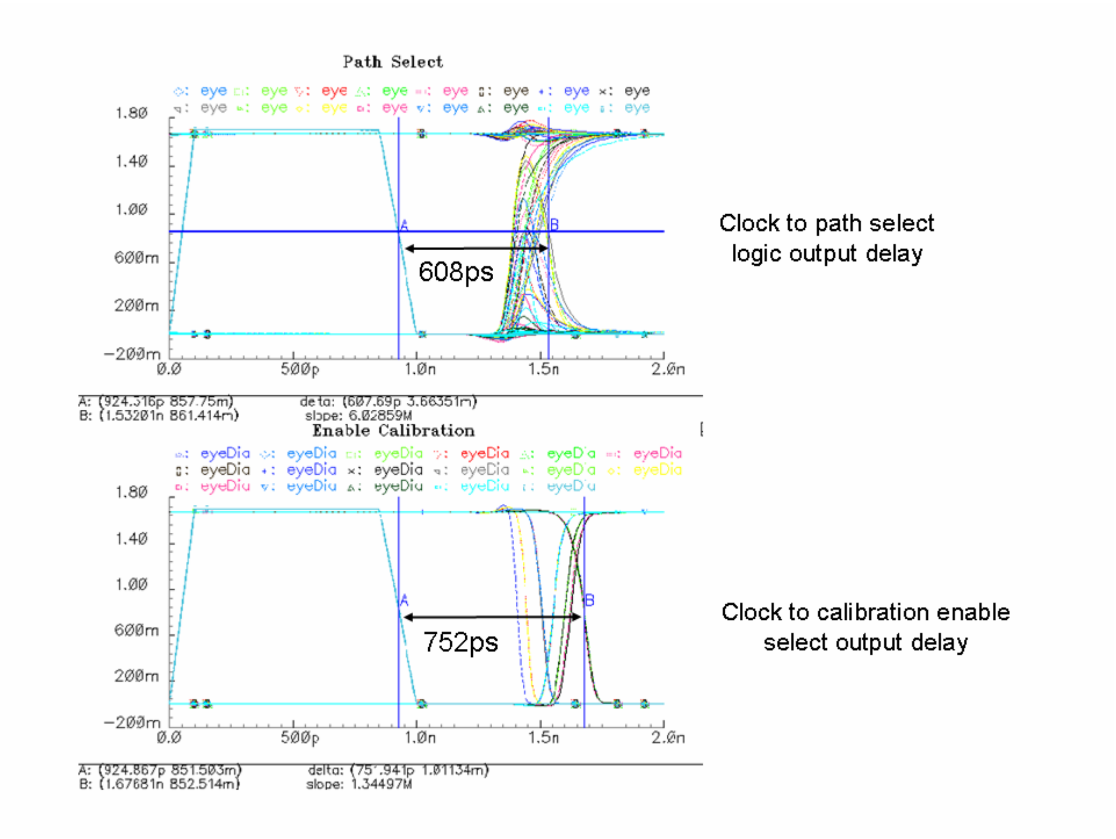


Figure 5.6: Simulation results for the unit cell calibration selection logic

5.2.5: Design of signal path de-multiplexer

The signal path de-multiplexer maps the 15-bit thermometric code output from the flash quantizer to a 16-bit thermometric code input to the DAC with an extra bit identifying the cell being calibrated. Each flash comparator output can be mapped to one of two possible unit DAC element current sources. Fig. 5.7 shows the circuit based on tri-state gates. These gates drive the signal from the flash comparator to the current source currently being used in the signal path and away from the current source being calibrated depending on an output-direction selection bit generated by the Calibration Selection Logic block.

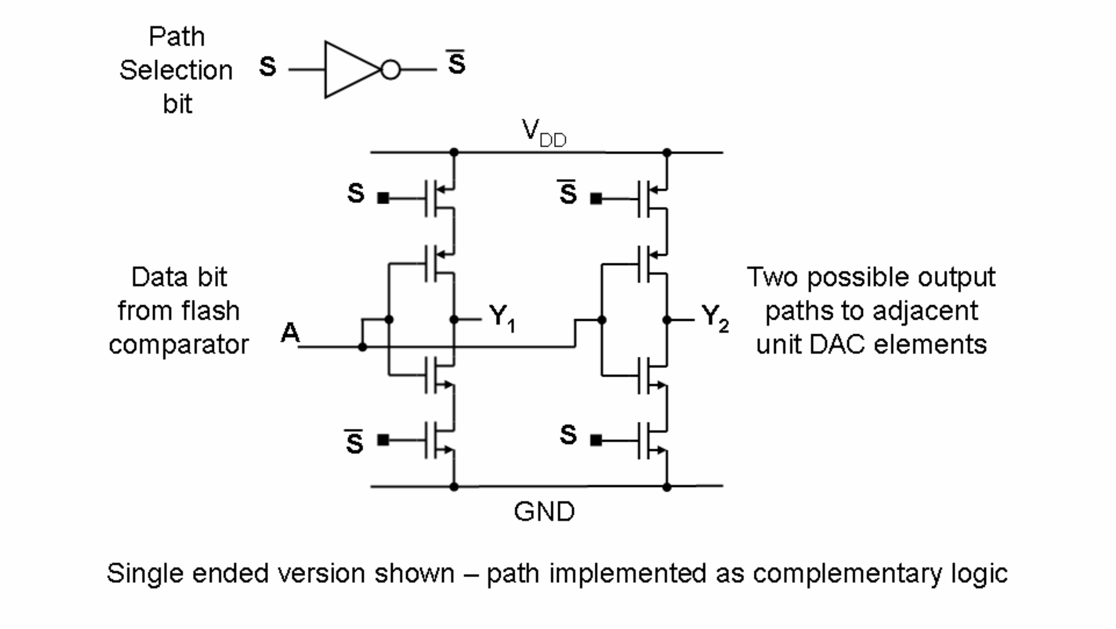

Path Selection bit
S
$\overline{S}$
V_{DD}
S
$\overline{S}$
Data bit from flash comparator
A
Y_1
Y_2
Two possible output paths to adjacent unit DAC elements
$\overline{S}$
S
GND
Single ended version shown – path implemented as complementary logic

Figure 5.7: Signal path de-multiplexer

5.2.6: Design of S-R latch with low switching latency

As indicated earlier in this chapter, the DAC elements are driven using a latch that delays the outputs of the comparator by half a clock cycle during normal operation. A novel S-R latch circuit is proposed that allows the design of a fast latch that asserts the DAC control signals at precise time points as defined by the clock as required for the operation of a continuous-time delta-sigma ADC. This latch also synchronizes the "Enable Calibration" signal to the cell being calibrated. These signals are asserted synchronous to the clock with very low latency (< 200ps). The schematic shown in Fig. 5.8a shows a conventional S-R latch based on cross-coupled NAND gates with an additional gate for the clock. These gates are merged to form a complex logic NAND gate, which are cross-coupled to form the new S-R latch shown in Fig. 5.8b.

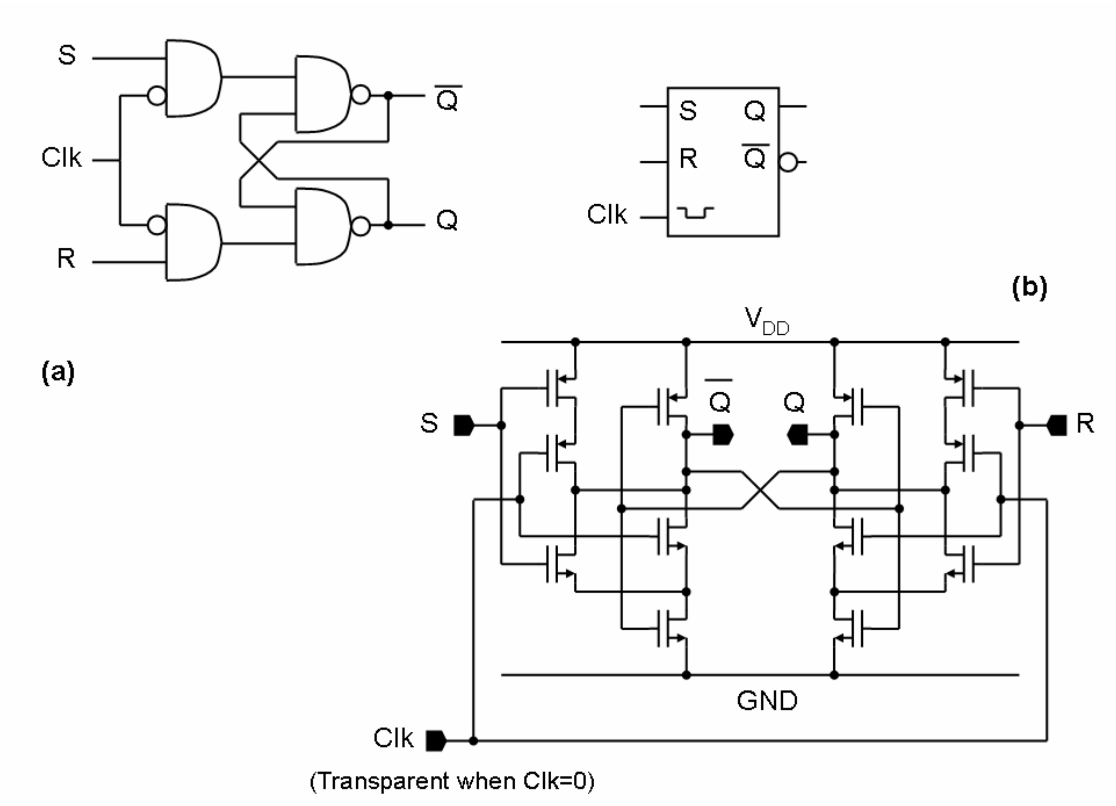


Figure 5.8: Low latency S-R latch

These S-R latches are combined in a master-slave configuration to form the switching network with added logic called the output latch as shown in Fig. 5.9. The data inputs to the DAC cell are forced to a high logic level i.e. the OFF position for PMOS DAC switches when calibration enable signal is asserted and the calibration enable outputs are asserted during this time. The data is passed through to the PMOS DAC switch control signals when calibration is disabled.

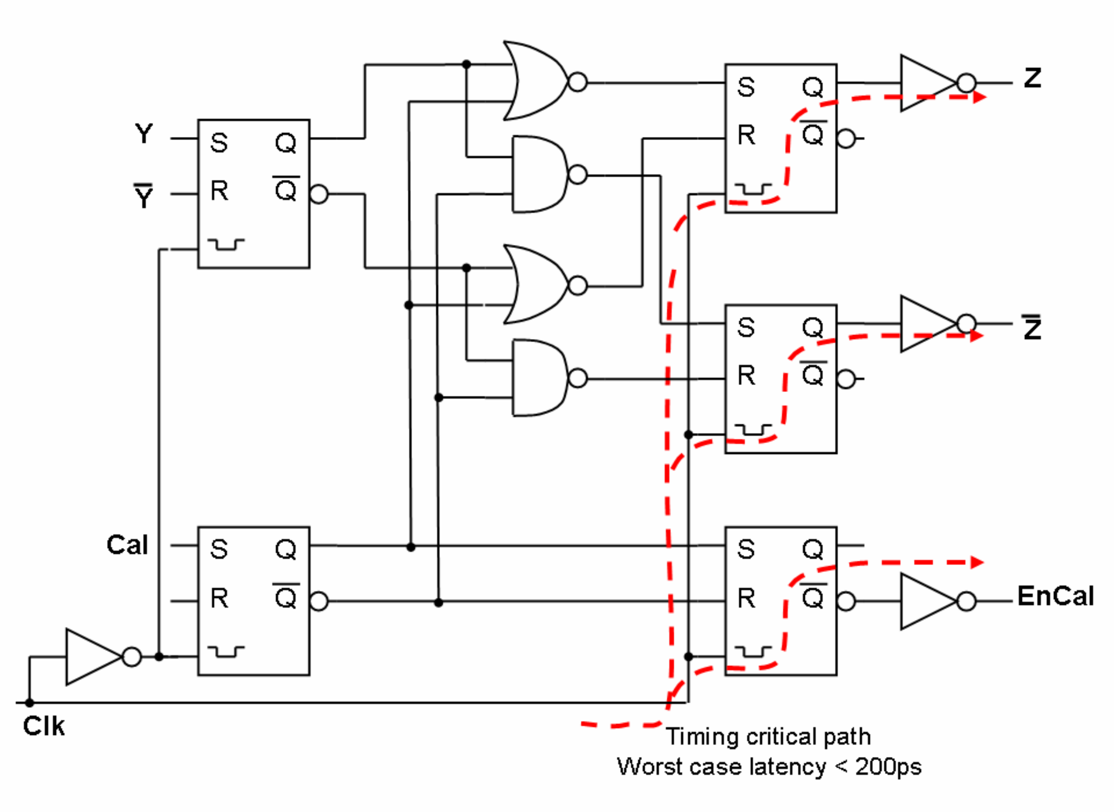


Figure 5.9: Output latch design

Note that the output signals can change state only when the clock to the output latches changes as shown in Fig. 5.10. This implies that the control signals to the DAC cells are completely synchronous. The S-R latch output exhibit clock feed-through that cause the DAC performance to degrade and this is avoided by buffering the outputs using inverters. A low crossover point is preferred by the PMOS DAC switches, which can be provided by the NAND S-R latch followed by the inverters. The root mean square jitter is determined from simulations to be 32fs, which is well within the 1ps specification determined for the system being designed.

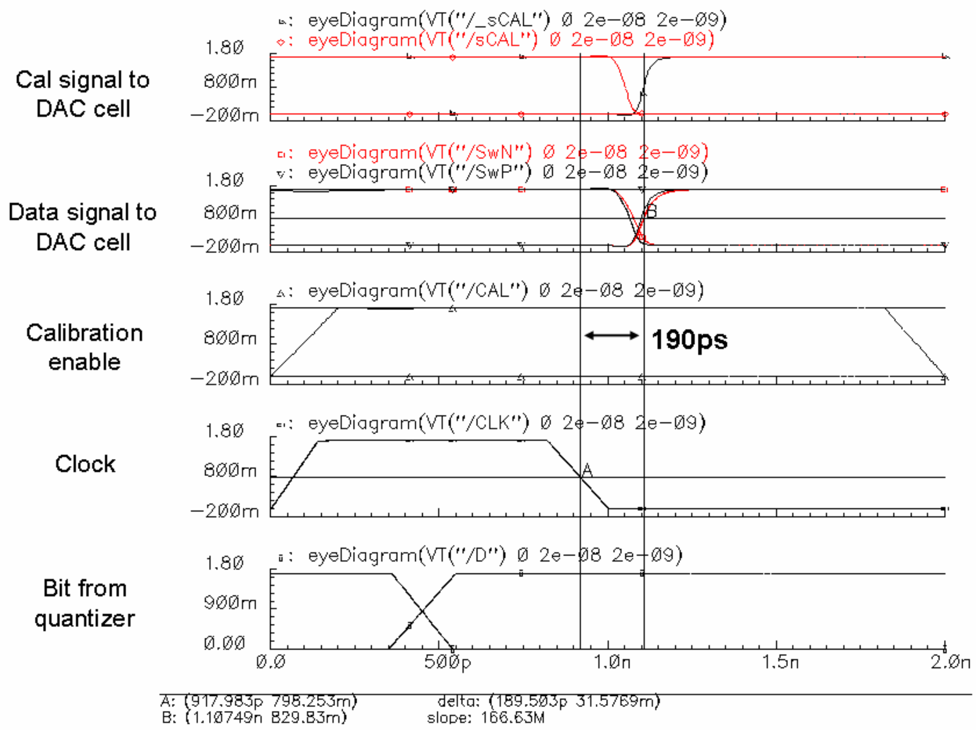


Figure 5.10: Output latch simulation result.

5.2.7: Design of unit current source

The unit DAC element current source is designed using a PMOS transistor with 2μm length that is larger than the minimum length and a PMOS cascode transistor with the minimum channel length of 0.18μm in this process. The nominal value of the unit DAC current is 34.21μA. Instead of connecting the entire current source device in feedback during calibration, the unit DAC current source element having an effective width *W* is split into a large static current source with width *(1-α).W* biased using a fixed bias voltage and a smaller variable current source with width *α.W* as shown in Fig. 5.11. The loop gain during calibration is enhanced by the ratio of the sizes of the two sources '*α*'. This means that a larger voltage is stored across the calibration storage capacitor than if the entire current source was connected in feedback as shown in Eqn. 5.6.

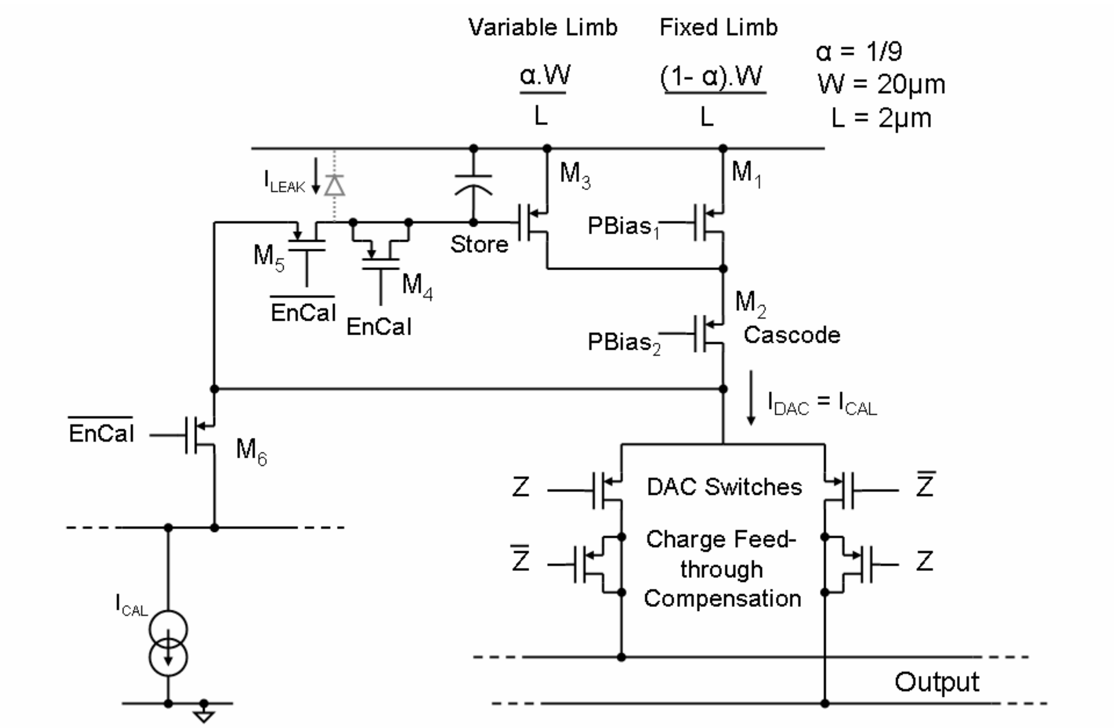


Figure 5.11: Current source

$$I_{VAR} = I_{CAL} - I_{FIXED}$$

$$I_{VAR} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W}{L} \cdot \left(V_{GSNom} - V_{Th}\right)^2 - \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W \cdot \left(1 - \alpha - \delta\right)}{L} \cdot \left(V_{GSNom} - V_{Th}\right)^2$$

$$I_{VAR} = \frac{\mu \cdot C_{OX}}{2} \cdot \frac{W}{L} \cdot \left(V_{GSNom} - V_{Th}\right)^2 \cdot \left(\alpha + \delta\right)$$

$$V_{STORE} = V_{Th} + \sqrt{\frac{2 \cdot I_{VAR}}{\mu \cdot C_{OX} \cdot \frac{W \cdot \alpha}{L}}} = V_{Th} + \left(V_{GSNom} - V_{Th}\right) \cdot \sqrt{1 + \frac{\delta}{\alpha}} \quad . \qquad \ldots \text{Eqn. 5.6}$$

In this design, the widths of the variable and the fixed current source transistors are chosen to be 2μm and 18μm leading to an attenuation of nine in the transconductance from the calibration charge storage node to the output current. During the transition from calibration to normal operation mode, the switch used to connect the current source in calibration mode suffers from charge injection through the overlap capacitance and channel charge feed-through.

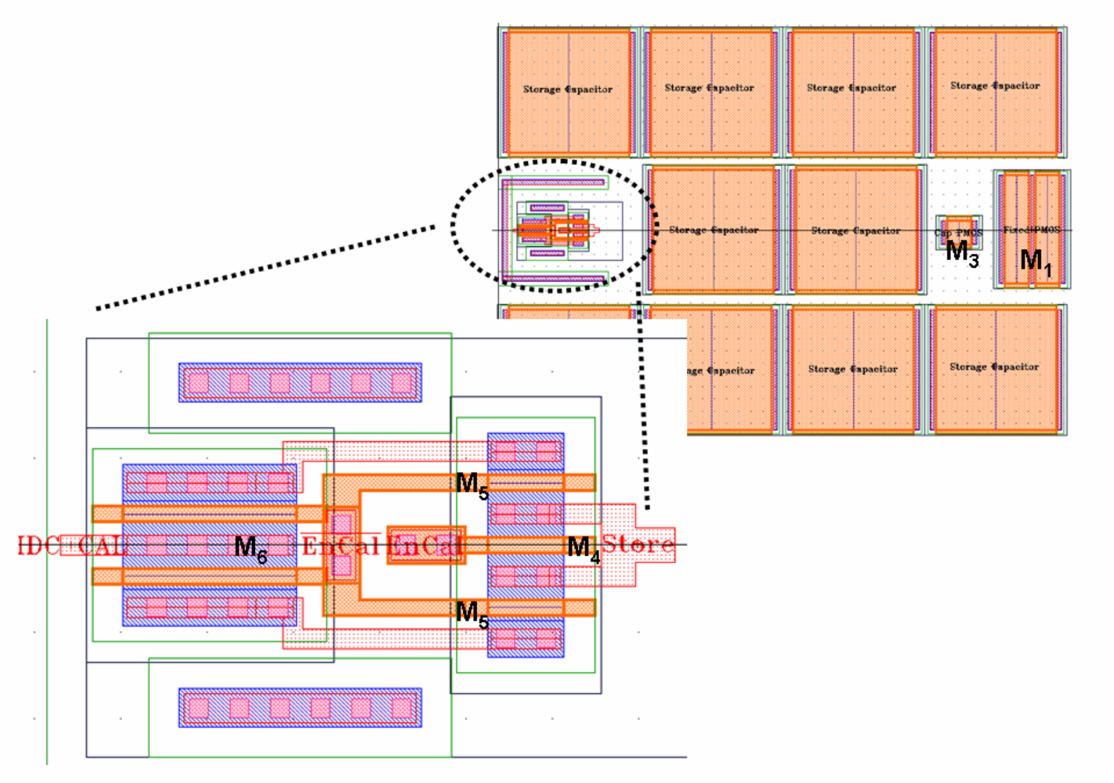


Figure 5.12: Physical layout for minimum storage penalty

Since the calibration can initially take a few cycles to converge, a small MOSFET is used for the switch M_5 with a dummy switch M_4 with half the width of the main switch to reduce the charge injection. An intentional capacitance is added to the charge storage node to reduce the impact of the leakage current and to reduce the voltage jump during the transition from calibration to active operation. Because of the split in the current source to static and variable parts, the gain from the storage node to the output is reduced by the ratio ‘α’. This reduces the impact of any residual charge injection on the output current during the calibration to active operation transition. The layout that is suitable for this design is shown in Fig. 5.12. This picture shows the relative sizes of the additional storage capacitors and the relative sizes of the switch. The switch layout minimizes the total diode area connected to the storage node by sharing the drain implant of the switch PMOS with the dummy switch PMOS.

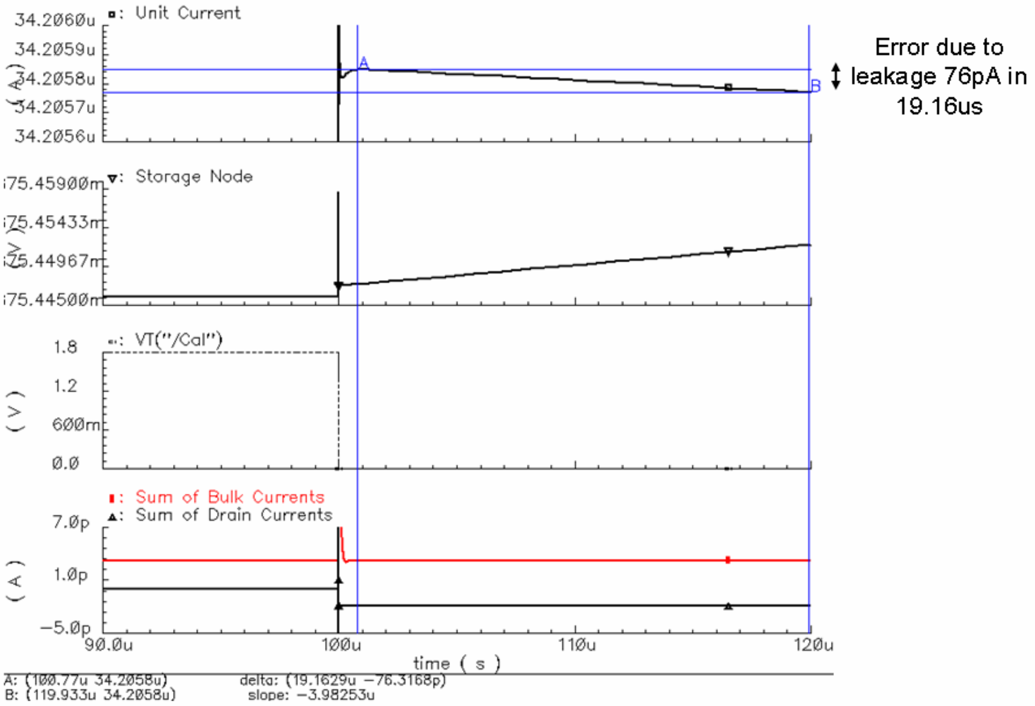


Figure 5.13: Change of mode from calibration to normal operation

Waveforms from circuit simulations in Fig. 5.13 show that a jump in the storage node voltage is observed when the current cell changes state from calibration to normal operation mode. This is the same level for all calibration-to-normal operation transitions and this causes a pedestal error for all the unit elements, which to the first order is not a mismatch error. This jump is minimized by reducing the charge feed-through due to gate to source/drain overlap capacitance and by reducing the channel charge injection and this is possible by using the smallest transistors in the CMOS process and by the additional capacitance on the storage node. The pedestal error is further attenuated by the factor α. These design choices lead to an absolute pedestal error in the output to be of the order of an LSB at the 15-b accuracy level. The mismatch in this pedestal between the different unit DAC elements will be smaller than the absolute error and hence is not significant in the effective DAC performance. This leakage current when the cell is in the normal operation mode is determined to be of the order of 2pA based on circuit simulations. This implies that an error of the order of an LSB at the 18-bit accuracy level will occur over a period of 20µs i.e. when the unit current changes by 76pA around the nominal value of 34.21µA. Each unit DAC element is held in the calibration configuration for 800ns. In order to calibrate 16 unit DAC elements, each cell is calibrated once every 16 x 800ns i.e. 12.8µs, which is less than the 20µs period over which the DAC cell could potentially accumulate significant errors due to charge leakage. This safety margin is programmable through the configuration set up in the digital scheduling engine.

5.2.8: Simulation of the calibration system

The complete calibration system was simulated using the Synopsys Nanosim mixed-mode event-driven simulator. Fig. 5.14 shows the calibration process. It is observed that the charge storage nodes converge towards the desired value very rapidly. A full simulation is not practical in a circuit simulator like SPICE or Spectre because of simulation time limitations.

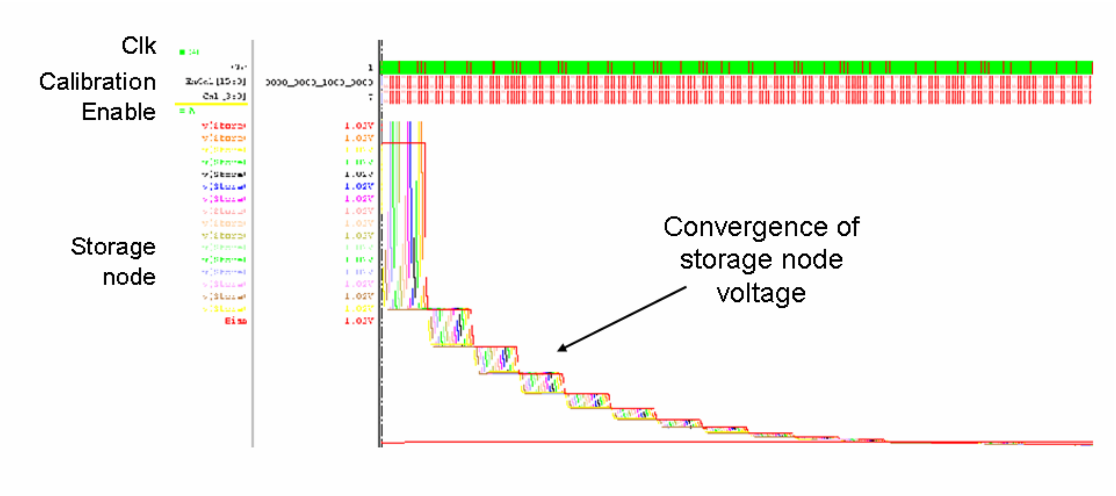


Figure 5.14: Top level simulation of calibration system.

5.3: CONCLUSION AND PROPOSED FURTHER WORK

This chapter describes the use of a known calibration scheme extended to perform calibration of the feedback DAC in a continuous-time delta-sigma ADC with a multi-bit quantizer. This scheme can be used as a part of a toolbox of techniques that is applied to the design of a delta-sigma ADC for broadband communications. Future work should focus on implementing the scheme in a high-speed continuous-time delta-sigma data converter chip to verify the operation of the proposed circuits.

Appendix A:

Design of a Miller compensated amplifier using the unit amplifier method

A method to design an amplifier with knowledge of device parasitic capacitances is described. This technique is used to optimize the design of a Miller compensated amplifier.

As described in Section 4.2.3, the procedure of designing a Miller compensated amplifier often involves several assumptions and the basic design equations are not directly compatible with the knowledge of the device parasitic capacitances. Eqns. A.1-A.7 are used during the process of designing a Miller compensated amplifier in a feedback configuration as used in a switched capacitor amplifier. Eqn. A.1 expresses the phase margin Φ_M as a function of the closed loop bandwidth ω_{CL} and the frequency of the non-dominant pole ω_{NDP}. Eqn. A.2 expresses the feedback factor β in terms of the number of bits resolved by the stage n, the total stage capacitance C_{Stg1} and the total parasitic capacitance C_{P1} at the input node marked (1) in Fig. A.1. Eqn. A.3 relates the closed loop bandwidth with the unity-gain bandwidth ω_{UGB} of the amplifier. Eqn. A.4 expresses the total load capacitance C_L in terms of the feedback network capacitances, the stage capacitance of the next stage C_{Stg2} and the total parasitic capacitance C_{P3} at the output node marked (3). Eqn. A.5-A.7 are the Miller compensation equations that express the unity-gain bandwidth, the non-dominant pole frequency and the Miller compensation zero frequency in terms of device parasitic capacitances, the transconductances of the input (M_1) and output (M_5) MOS transistors g_{m1} and g_{m5} and the compensation capacitor C_C and the zero tuning resistor R_Z. C_{P2} is the total parasitic capacitance at the internal node marked (2).

$$\Phi_M = \frac{\pi}{2} - \tan^{-1}\left(\frac{\omega_{CL}}{-\omega_{NDP}}\right). \qquad \text{… Eqn. A.1}$$

$$\beta = \frac{C_1}{C_1 + C_2 + C_{P1}} = \frac{C_{Stg1} \cdot 2^{-n}}{C_{Stg1} + C_{P1}}. \qquad \text{… Eqn. A.2}$$

$$\omega_{CL} = \beta \cdot \omega_{UGB}. \qquad \text{… Eqn. A.3}$$

$$C_L = C_{Stg2} + \beta \cdot \left(\frac{C_{Stg1}}{2^n - 1} + C_{P1}\right) + C_{P3}. \qquad \text{… Eqn. A.4}$$

$$\omega_{UGB} = \frac{g_{m1}}{C_C}. \qquad \text{… Eqn. A.5}$$

$$\omega_{NDP} = -\frac{g_{m5} \cdot C_C}{C_C \cdot (C_L + C_{P2}) + C_L \cdot C_{P2}}. \qquad \text{… Eqn. A.6}$$

$$\omega_Z = -\frac{1}{C_C \cdot \left(\frac{1}{g_{m5}} - R_Z\right)}. \qquad \text{… Eqn. A.7}$$

The design procedure involves starting with an initial design with assumed parasitic capacitance values based on experience and iterating until the desired frequency response is met. Several iterations are usually required to position the non-dominant pole ω_{NDP} correctly since this method does not make use of information about the parasitic capacitance scaling with the scaling of the devices M_1 … M_6. The zero due to Miller compensation is usually positioned to improve the phase margin either by placing this close to the unity gain frequency or often by attempting to cancel the non-dominant pole.

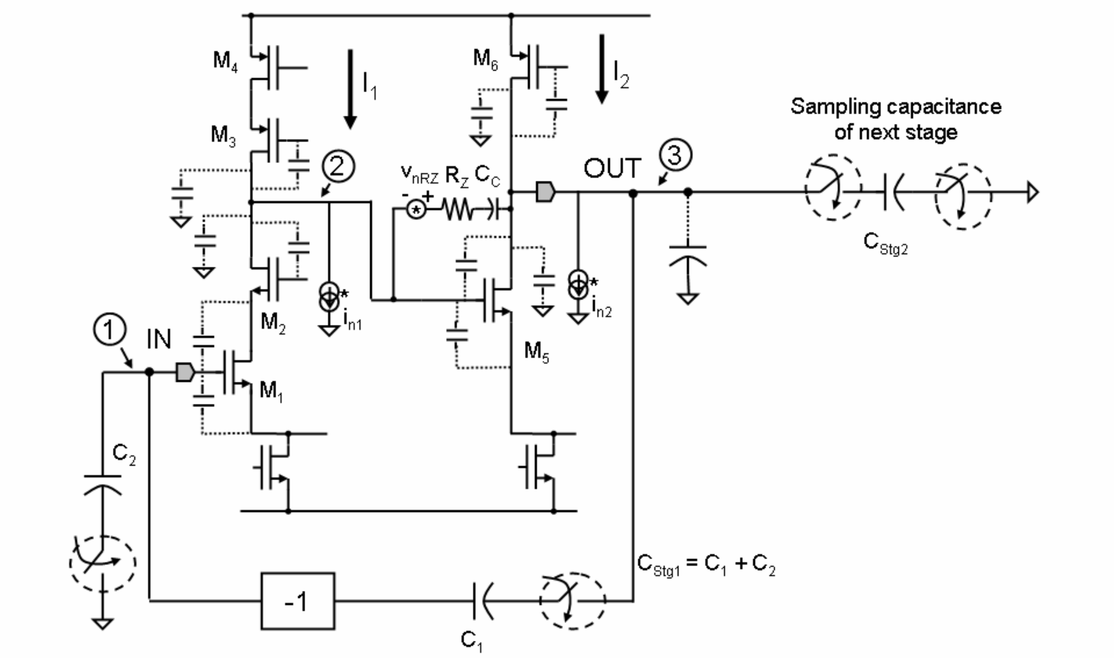


Figure A.1: Unit amplifier design method.

It is however possible to set up the problem in a way that allows the power to be optimized and is aware of the impact of parasitic capacitance scaling with scaling of devices in the amplifier. Such a technique also leads to a closed form solution for the choice of the compensation capacitor. The proposed approach is called the unit amplifier method. In this approach, an initial design of the amplifier is performed in which all the transistors are biased in the desired region of operation and the DC gain is measured. This circuit is also used to verify that the output swing can be achieved with the desired static linearity. This step is typically performed with a choice of an initial current – typically the unit bias current flowing through each current path hence called the unit amplifier. If the bias current in a limb is scaled by a factor *k* and all the transistors are scaled by the same scale factor, then the bias voltages like the overdrive voltage and the drain to source saturation voltages do not change. All transconductances and parasitic capacitances scale by the same factor. The noise voltage/current scales by a factor of square root of *k*. The transconductances and parasitic capacitances of interest are measured from an initial circuit simulation (or can be computed from the device models) for the properly biased unit amplifier and then are normalized by the bias currents I_1 and I_2 in the first and the second stages of the amplifier shown in Fig. A.1. The Miller amplifier equations can be rewritten using these unit amplifier parameters as in Eqn. A.8-A.16.

$$C_{P1} = I_1 \cdot \left(C_{UM1gd} \cdot (g+1) + C_{UM1gs}\right) . \qquad \text{… Eqn. A.8}$$

$$C_{P2} = I_1 \cdot \left(C_{UM2db} + C_{UM2gd} + C_{UM3db} + C_{UM3gd}\right) + I_2 \cdot C_{UM5gs} . \qquad \text{… Eqn. A.9}$$

$$C_{P3} = I_2 \cdot \left(C_{UM5db} + C_{UM6db} + C_{UM6gd}\right) . \qquad \text{… Eqn. A.10}$$

$$C_L = C_{Stg2} + \beta \cdot \left(\frac{C_{Stg1}}{2^n - 1} + C_{P1}\right) + C_{P3} . \qquad \text{… Eqn. A.11}$$

$$\beta = \frac{C_{Stg1} \cdot 2^{-n}}{C_{Stg1} + I_1 \cdot C_{UP1}} . \qquad \text{… Eqn. A.12}$$

$$\omega_{UGB} = \frac{I_1 \cdot g_{Um1}}{C_C} . \qquad \text{… Eqn. A.13}$$

$$\omega_{CL} = \beta \cdot \omega_{UGB} = \frac{C_{Stg1} \cdot 2^{-n}}{C_{Stg1} + C_{P1}} \cdot \frac{I_1 \cdot g_{Um1}}{C_C} . \qquad \text{… Eqn. A.14}$$

$$\omega_{NDP} = -\frac{I_2 \cdot g_{Um5} \cdot C_C}{C_C \cdot (C_L + C_{P2}) + C_L \cdot C_{P2}} . \qquad \text{… Eqn. A.15}$$

$$\omega_Z = -\frac{1}{C_C \cdot \left(\frac{1}{I_2 \cdot g_{Um5}} - R_Z \right)} \quad .$$

… Eqn. A.16

Note that this set of equations is completely specified using unit device parasitic capacitances and conductances parameterized using bias currents I_1 and I_2. Here g is the intrinsic gain of the input transistor M_1. The capacitances and transconductances with the *U* subscript represent the values normalized to the bias currents. This is a closed form of equations, albeit one that is not solved in a straightforward fashion. For a chosen phase margin, this equation set can be minimized for the least power consumption i.e. minimum value of (I_1+I_2). This leads to a choice of C_C that is unique and optimum for the chosen design and is obtained without making any further assumptions. These equations can be drawn as a contour in the I_1-I_2 space for different values of phase margin. The optimum power solution for a chosen phase margin or a chosen SNR is the point where the line (I_1+I_2 = constant) is tangential to the phase margin contour or the SNR contour as shown in Fig. A.2.

$$v_3 = \frac{i_{n1} \cdot N_1(s) + i_{n2} \cdot N_2(s) + v_{nR} \cdot N_{31}(s)}{D(s)} \quad .$$

… A.17

$$\begin{aligned} D(s) = &\left(g_{m1} \cdot g_{m5} \cdot \beta + \frac{1}{r_{o1} \cdot r_{o2}} \right) \\ &+ s \cdot \left(\frac{C_L}{r_{o1}} + \frac{C_{P2}}{r_{o2}} + C_C \cdot \left(\frac{R_Z}{r_{o1} \cdot r_{o2}} + g_{m5} - g_{m1} \cdot \beta + \frac{1}{r_{o1}} + \frac{1}{r_{o2}} + g_{m1} \cdot g_{m5} \cdot \beta \cdot R_Z \right) \right) \\ &+ s^2 \cdot \left(C_L \cdot C_{P2} + C_C \cdot C_{P2} + C_L \cdot C_C + C_C \cdot R_Z \cdot \left(\frac{C_L}{r_{o1}} + \frac{C_{P2}}{r_{o2}} \right) \right) \\ &+ s^3 \cdot C_C \cdot C_{P2} \cdot C_L \cdot R_Z \end{aligned}$$

. … A.18

$$N_1(s) = g_{m5} + s \cdot (g_{m5} \cdot R_Z - 1) \cdot C_C \ . \qquad \ldots \text{A.19}$$

$$N_2(s) = \frac{1}{r_{o1}} + s \cdot \left(\left(\frac{R_Z}{r_{o1}} + 1 \right) \cdot C_C + C_{P2} \right) + s^2 \cdot C_{P2} \cdot C_C \cdot R_Z \ . \qquad \ldots \text{A.20}$$

$$N_3(s) = s \cdot \left(g_{m5} + \frac{1}{r_{o1}} \right) \cdot C_C + s^2 \cdot C_{P2} \cdot C_C \ . \qquad \ldots \text{A.21}$$

The noise voltage at the output of a Miller compensated amplifier can be derived as shown in Eqn. A.17 – A.21. It is important to note that in a Miller compensated amplifier, the capacitance at the output node C_L sets the non-dominant pole and the internal compensation capacitor C_C sets the dominant pole i.e. the noise bandwidth. The input referred noise is obtained by appropriately scaling the integral of the amplitude noise spectrum as shown in Eqn. A.22. The cross integrals of the noise variables is assumed to be zero because the limb noise currents and the noise due to the Miller resistor are uncorrelated random variables.

$$v_{ni} = \frac{\sqrt{\int_0^\infty \left| \left(v_3(s) \Big|_{s=j\cdot\omega} \right)^2 \right| \cdot d\omega}}{2^n} \ . \qquad \ldots \text{A.22}$$

The integral is solved numerically. The circuit noise can be parameterized in terms of the limb currents leading to the SNR contour and the intersection of these curves will lead to a solution with appropriate phase margin and signal to noise ratio as shown in Fig. A.2. The Miller zero is positioned as per the method described in Section 4.2.3. An interesting observation is that for the current values at the optimum solution the input referred noise decreases by 1.3% when the load capacitor is changed by 10% and changes by 2.8% when the compensation capacitor changes by 10%. This clearly shows that the use of the $\gamma.kT/C_L$ (or $\gamma.kT/C_C$) equation for the design of a switched capacitor circuit with a Miller compensated amplifier is not an appropriate model which should have shown a decrease of noise by 5.4% when the load capacitor is changed by 10%.

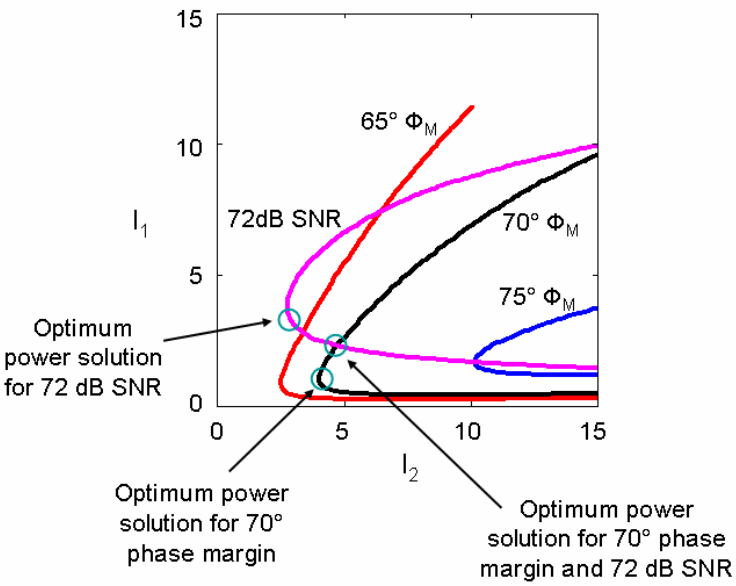


Figure A.2: Optimum solution for a Miller compensated amplifier

The set of equations described in this section are approximate and represent a simplified model of the amplifier as a result the solution is not perfect. However, this leads to a good starting point for the amplifier design and reduces the number of iterations needed in the design process. The design is further tweaked using simulations to achieve the desired transient settling performance.

Bibliography

[1] http://en.wikipedia.org/wiki/History_of_technology

[2] E. A. Lee and D. G. Messerschmitt, "Digital Communication," pp. 14-15, *Kluwer Academic Publishers*, 1988

[3] M. J. M. Pelgrom, A. C. J. Duinmaijer, A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol.24, no.5, pp. 1433-1439, Oct 1989

[4] B. Razavi, B. A. Wooley, "Design techniques for high-speed, high-resolution comparators," *IEEE J. Solid-State Circuits,* vol.27, no.12, pp.1916-1926, Dec 1992

[5] N. Da Dalt, M. Harteneck, C. Sandner, A. Wiesbauer, "On the jitter requirements of the sampling clock for analog-to-digital converters," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.49, no.9, pp. 1354-1360, Sep 2002

[6] S. K. Gupta, V. Fong, "A 64-MHz clock-rate ΣΔ ADC with 88-dB SNDR and -105-dB IM3 distortion at a 1.5-MHz signal frequency," *IEEE J. Solid-State Circuits*, vol.37, no.12, pp. 1653-1661, Dec 2002

[7] A.M. Abo and P.R. Gray, "A 1.5-V, 10-bit, 14.3-MS/s CMOS pipeline analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol.34, no.5 pp.599-606, May 1999

[8] A. R. Bugeja, B.-S. Song, P. L. Rakers, S. F. Gillig, "A 14-b, 100-MS/s CMOS DAC designed for spectral performance," *IEEE J. Solid-State Circuits*, vol.34, no.12, pp.1719-1732, Dec 1999

[9] H. Stark, J. W. Woods, "Probability and Random Processes with Applications to Signal Processing- 3rd Ed.," *Prentice Hall,* 2002

[10] J. Doernberg, H. Lee, D. Hodges, "Full-speed testing of A/D converters*," IEEE J. Solid-State Circuits*, Dec. 1984

[11] "IEEE standard for terminology and test methods for analog-to-digital converters," *IEEE Std. 1241-2000*

[12] C. Enz, Y. Cheng, "MOS transistor modeling for RF IC design," *IEEE J. Solid-State Circuits*, vol.35, no.2, pp.186-201, Feb 2000.

[13] Hui Pan, A. A. Abidi, "Spectral spurs due to quantization in Nyquist ADCs," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol.51, no.8, pp. 1422-1439, Aug. 2004

[14] J. Tsimbinos, K. V. Lever, "Applications of higher-order statistics to modelling, identification and cancellation of nonlinear distortion in high-speed samplers and analogue-to-digital converters using the Volterra and Wiener models," *IEEE Sig. Proc. Workshop on Higher-Order Statistics,* pp.379-383, 1993

[15] D. J. Huber, R. J. Chandler, A. A. Abidi, "A 10b 160MS/s 84mW 1V Subranging ADC in 90nm CMOS," *ISSCC Dig. Tech. Papers.*, pp.454-615, 11-15 Feb. 2007

[16] Tzi-Hsiung Shu, Bang-Sup Song, K. Bacrania, "A 13-b 10-Msample/s ADC digitally calibrated with oversampling delta-sigma converter," *IEEE J. Solid-State Circuits*, vol.30, no.4, pp.443-452, Apr 1995

[17] B. Song, S. Lee, and M. Tompsett, "A 10b 15MHz CMOS recycling two-step A/D converter," *IEEE J. Solid-State Circuits*, vol. SC-25, pp. 1328-1338, Dec. 1990.

[18] R. W. Watson, C. W. Hastings, "Self-checked computation using residue arithmetic," *Proc. IEEE*, vol.54, no.12, pp. 1920-1931, Dec. 1966

[19] S. Lewis and P. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," *IEEE J. Solid-State Circuits*, vol.22, no.6 pp. 954- 961, Dec. 1987.

[20] R. Schreier, "On the use of chaos to reduce idle-channel tones in delta-sigma modulators," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol.41, no.8, pp. 539-547, Aug 1994

[21] S. Ardalan, J. Paulos, "An analysis of nonlinear behavior in delta - sigma modulators," *IEEE Trans. Circuits and Systems*, vol.34, no.6, pp. 593-603, Jun 1987

[22] R. Schreier, "An empirical study of high-order single-bit delta-sigma modulators," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.40, no.8, pp.461-466, Aug 1993

[23] R. T. Baird, T. S. Fiez, "Stability analysis of high-order delta-sigma modulation for ADCs," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.41, no.1, pp.59-62, Jan 1994

[24] F. Lu, J. Min, S. Liu, K. Cameron, C. Jones, O. Lee, J. Li, A. Buchwald, S. Jantzi, C. Ward, K. Choi, J. Searle, H. Samueli, "A single-chip universal burst receiver for cable modem/digital cable-TV applications," *Proc. CICC*, pp.311-314, 2000

[25] D. Ehrhardt, T. Benkner, “Digital TV receiver with a low IF,” *IEEE Trans. Consumer Electronics*, vol.39, no.3, pp.331-339, 8-10 Jun 1993

[26] W. Kluge, L. Dathe, R. Jaehne, S. Ehrenreich, D. Eggert, “A 2.4GHz CMOS transceiver for 802.11b wireless LANs,” *ISSCC Dig. Tech. Papers*, pp. 360-361 vol.1, 2003

[27] A. Zanchi, F. Tsay, “A 16-bit 65-MS/s 3.3-V pipeline ADC core in SiGe BiCMOS with 78-dB SNR and 180-fs jitter,” *IEEE J. Solid-State Circuits*, vol.40, no.6, pp. 1225-1237, June 2005

[28] C. Moreland, F. Murden, M. Elliott, J. Young, M. Hensley, R. Stop, “A 14-bit 100-Msample/s subranging ADC,” *IEEE J. Solid-State Circuits*, vol.35, no.12, pp.1791-1798, Dec 2000

[29] P. Malla, H. Lakdawala, K. Kornegay, K. Soumyanath, “A 28mW Spectrum-Sensing Reconfigurable 20MHz 72dB-SNR 70dB-SNDR DT ΔΣ ADC for 802.11n/WiMAX Receivers”, *ISSCC Dig. Tech. Papers*, pp. 496-497, 2008

[30] T. Christen, T. Burger, Qiuting Huang, “A 0.13µm CMOS EDGE / UMTS / WLAN Tri-Mode ΔΣ ADC with -92dB THD," *ISSCC Dig. Tech. Papers*, pp.240-599, 2007

[31] J. Ming, S. Lewis, “An 8-bit 80-Msample/s pipelined analog-to-digital converter with background calibration,” *IEEE J. Solid-State Circuits*, vol. SC-36, pp. 1489-1497, Oct. 2001.

[32] K. Poulton, R. Neff, B. Setterberg, B. Wuppermann, T. Kopley, R. Jewett, J. Pernillo, C. Tan, A. Montijo, “A 20 GS/s 8 b ADC with a 1 MB memory in 0.18µm CMOS,” *ISSCC Dig. Tech. Papers*, pp. 318-496 vol.1, 2003

[33] R. J. Van De Plassche, “Dynamic element matching for high-accuracy monolithic D/A converters,” *IEEE J. Solid-State Circuits*, vol.11, no.6, pp. 795-800, Dec 1976

[34] I. Galton, “Digital cancellation of D/A converter noise in pipelined A/D converters,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing,* vol.47, no.3, pp.185-196, Mar 2000

[35] R. Aparicio and A. Hajimiri, “Capacity limits and matching properties of integrated capacitors,” *IEEE J. Solid-State Circuits*, vol.37, no.3 pp.384-393, March 2002.

[36] B. Murmann and B. Boser, “A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification,” *IEEE J. Solid-State Circuits*, vol. SC-38, pp. 2040-2050, Dec. 2003.

[37] A. Panigada, I. Galton, “Digital Background Correction of Harmonic Distortion in Pipelined ADCs,” *IEEE Trans. Circuits and Systems I: Regular Papers,* vol.53, no.9, pp. 1885-1895, Sept. 2006

[38] J. P. Keane, P. J. Hurst, S. H. Lewis, “Digital background calibration for memory effects in pipelined analog-to-digital converters,” *IEEE Trans. Circuits and Systems I: Regular Papers,* vol.53, no.3, pp. 511-525, March 2006

[39] D. M. Hummels, F. H. Irons, R. Cook, I. Papantonopoulos, “Characterization of ADCs using a non-iterative procedure,” *Proc. ISCAS*, vol.2, pp.5-8, 1994

[40] D. Moulin, "Real-time equalization of A/D converter nonlinearities," *Proc. ISCAS*, vol.1, pp.262-267, 1989

[41] P. Holloway, M. Norton, “A high yield, second generation 10-bit monolithic DAC,” *ISSCC Dig. Tech. Papers.* vol.XIX, pp. 106-107, Feb 1976

[42] J. R. Naylor, “A complete high-speed voltage output 16-bit monolithic DAC,” *IEEE J. Solid-State Circuits*, vol.18, no.6, pp. 729-735, Dec 1983

[43] B. J. Tesch, J. C. Garcia, “A low glitch 14-b 100-MHz D/A converter,” *IEEE J. Solid-State Circuits*, vol.32, no.9, pp.1465-1469, Sep 1997

[44] Wang Ruoxu, “Design of a high speed 12-bit subranging A/D converter,” *Proc. Int. Conf. Solid-State and Integ. Circuit Tech*, pp.389-392, 1998

[45] B. Song, M. Tompsett, and K. Lakshmikumar, “A 12-Bit 1-MSample/s capacitor error averaging pipelined A/D converter,” *IEEE J. Solid-State Circuits*, vol. SC-23, pp. 1324-1333, Dec. 1988.

[46] Hsin-Shu Chen, Bang-Sup Song, K. Bacrania, “A 14-b 20-Msamples/s CMOS pipelined ADC,” *IEEE J. Solid-State Circuits*, vol.36, no.6, pp.997-1001, Jun 2001

[47] P. C. Yu, Hae-Seung Lee, “A 2.5-V, 12-b, 5-MSample/s pipelined CMOS ADC,” *IEEE J. Solid-State Circuits*, vol.31, no.12, pp.1854-1861, Dec 1996

[48] S. Lee and B. Song, “Digital-domain calibration techniques for multi-step A/D Converters,” *IEEE J. Solid-State Circuits*, vol. SC-27, pp. 1679-1688, Dec. 1992.

[49] A. Karanicolas, H. Lee, and K. Bacrania, “A 15-b 1-MSample/s digitally self-calibrated pipeline ADC,” *IEEE J. Solid-State Circuits*, vol. SC-28, pp. 1207-1215, Dec. 1993.

[50] S. Kwak, B. Song, and K. Bacrania, “A 15b 5Msample/s low-spurious CMOS ADC,” *IEEE J. Solid-State Circuits*, vol. SC-32, pp. 1866-1875, Dec. 1997.

[51] K. Nair and R. Harjani, “A 96dB SFDR 50MS/s digitally enhanced CMOS pipelined A/D converter,” *ISSCC Dig. Tech. Papers*, pp. 456-457, Feb. 2004.

[52] E. Siragusa and I. Galton, “A digitally enhanced 1.8-V 15-bit 40-MSample/s CMOS pipelined ADC,” *IEEE J. Solid-State Circuits*, vol. SC-39, pp. 2126-2138, Dec. 2004.

[53] C. Grace, P. Hurst, and S. Lewis, “A 12-bits 80-Msample/s pipelined ADC with bootstrapped digital calibration,” *IEEE J. Solid-State Circuits*, vol. SC-40, pp. 1038-1046, May 2005.

[54] H. Liu, Z. Lee, and J. Wu, “A 15-b 40-MS/s CMOS pipelined analog-to-digital converter with digital background calibration,” *IEEE J. Solid-State Circuits*, vol. SC-40, pp. 1047-1056, May 2005.

[55] Y. Lin, B. Kim, and P. Gray, “A 13-b 2.5-MHz self-calibrated pipelined A/D converter in 3µm CMOS,” *IEEE J. Solid-State Circuits*, vol. SC-26, pp. 628-636, April 1991.

[56] J. Ingino and B. Wooley, “A continuously calibrated 12-b, 10-MS/s, 3.3-V A/D converter,” *IEEE J. Solid-State Circuits*, vol. SC-33, pp. 1920-1931, Dec. 1998.

[57] S. Ryu, S. Ray, B. Song, G. Cho, and K. Bacrania, “A 14b-linear capacitor self-trimming pipelined ADC,” *IEEE J. Solid-State Circuits*, vol. SC-39, pp. 2046-2051, Nov. 2004.

[58] Y. Chiu, P. Gray, and B. Nikolic, “A 14-b 12-MS/s CMOS pipeline ADC with over 100-dB SFDR,” *IEEE J. Solid-State Circuits*, vol. SC-39, pp. 2139 - 2151, Dec. 2004.

[59] K. Maio, M. Hotta, N. Yokozawa, M. Nagata, K. Kaneko, K. Iwasaki, “An untrimmed DAC with 14b resolution,” *ISSCC Dig. Tech. Papers*, vol.XXIV, Feb 1981

[60] Hae-Seung Lee, D. Hodges, “Self-calibration technique for A/D converters,” *IEEE Trans. Circuits and Systems*, vol.30, no.3, pp. 188-190, Mar 1983

[61] H.-S. Lee, D. A. Hodges, P. R. Gray, “A self-calibrating 15 bit CMOS A/D converter,” *IEEE J. Solid-State Circuits*, vol.19, no.6, pp. 813-819, Dec 1984

[62] Hae-Seung Lee; D. Hodges, “Accuracy considerations in self-calibrating A/D converters,” *IEEE Trans. Circuits and Systems*, vol.32, no.6, pp. 590-597, Jun 1985

[63] J. Goes, J. C. Vital, J. E. Franca, “A CMOS 4-bit MDAC with self-calibrated 14-bit linearity for high-resolution pipelined A/D converters,” *Proc. CICC*, pp.105-108, May 1996

[64] Y. Manoli, “A self-calibration method for fast high-resolution A/D and D/A converters,” *IEEE J. Solid-State Circuits*, vol.24, no.3, pp.603-608, Jun 1989

[65] Z. Boyacigiller, B. Weir, P. Bradshaw, “An error-correcting 14b/20μs CMOS A/D converter,” *ISSCC Dig. Tech. Papers*, vol.XXIV, pp. 62-63, Feb 1981

[66] S.-H. Lee, B.-S. Song, “Simplified digital calibration for multi-stage analog-to-digital converters,” *Proc. ISCAS*, pp.1216-1219, vol.2, 1993

[67] Hae-Seung Lee, “A 12-b 600 ks/s digitally self-calibrated pipelined algorithmic ADC ,” *IEEE J. Solid-State Circuits*, vol.29, no.4, pp.509-515, Apr 1994

[68] D. W. J. Groeneveld, H. J. Schouwenaars, H. A. H. Termeer, C. A. A. Bastiaansen, “A self-calibration technique for monolithic high-resolution D/A converters,” *IEEE J. Solid-State Circuits,*, vol.24, no.6, pp.1517-1522, Dec 1989

[69] P. Benabes, P.Aldebert, A. Yahia, R. Kielbasa, “Influence of the feedback DAC delay on continuous-time bandpass ΣΔ coverter,” *Electronics Letters*, vol.36, no.4, pp.292-294, 17 Feb 2000

[70] K. Falakshahi, C.-K. K. Yang, B. A. Wooley, “A 14-bit, 10-Msamples/s D/A converter using multibit Σ-Δ modulation,” *IEEE J. Solid-State Circuits*, vol.34, no.5, pp. 607-615, May 1999

[71] S. Yan, E. Sanchez-Sinencio, “A continuous-time sigma-delta modulator with 88-dB dynamic range and 1.1-MHz signal bandwidth,” *IEEE J. Solid-State Circuits*, vol.39, no.1, pp. 75-86, Jan. 2004

[72] Li Zhimin, T. S. Fiez, “A 14 Bit Continuous-Time Delta-Sigma A/D Modulator With 2.5 MHz Signal Bandwidth,” *IEEE J. Solid-State Circuits*, vol.42, no.9, pp.1873-1883, Sept. 2007

[73] M. Clara, W. Klatzer, B. Seger,A. Di Giandomenico, L. Gori, “A 1.5V 200MS/s 13b 25mW DAC with Randomized Nested Background Calibration in 0.13μ CMOS,” *ISSCC Dig. Tech. Papers*, pp.250-600, Feb. 2007

[74] R. Jewett, K. Poulton, Kuo-Chiang Hsieh, J. Doernberg, “A 12 b 128 MSample/s ADC with 0.05 LSB DNL,” *ISSCC Dig. Tech. Papers.* pp.138-139, 443, 1997

[75] Yun Chiu, C. W. Tsang, B. Nikolic, P. R. Gray, “Least mean square adaptive digital background calibration of pipelined analog-to-digital converters,” *IEEE Trans. Circuits and Systems I: Regular Papers,* vol.51, no.1, pp. 38-46, Jan. 2004

[76] Myung-Jun Choe, Bang-Sup Song, K. Bacrania, “A 13-b 40-MSamples/s CMOS pipelined folding ADC with background offset trimming,” *IEEE J. Solid-State Circuits*, vol.35, no.12, pp.1781-1790, Dec 2000

[77] K. Nagaraj, “Self-calibration technique for pipe-lined algorithmic A/D converters,” *Proc. ISCAS*, vol.1, pp.696-699, 1995

[78] K. Nagaraj, “Area-efficient self-calibration technique for pipe-lined algorithmic A/D converters,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing,* vol.43, no.7, pp.540-544, Jul 1996

[79] Un-Ku Moon, Bang-Sup Song, “Background digital calibration techniques for pipelined ADCs,” *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.44, no.2, pp.102-109, Feb 1997

[80] G. Cauwenberghs, “Blind on-line digital calibration of multi-stage Nyquist-rate and oversampled A/D converters,” *Proc. ISCAS*, vol.1, pp.508-511, 1998

[81] O. E. Erdogan, P. J. Hurst, S. H. Lewis, “A 12-b digital-background-calibrated algorithmic ADC with -90-dB THD,” *IEEE J. Solid-State Circuits*, vol.34, no.12, pp.1812-1820, Dec 1999

[82] E. B. Blecker, T. M. McDonald, O. E. Erdogan, P. J. Hurst, S. H. Lewis, “Digital background calibration of an algorithmic analog-to-digital converter using a simplified queue,” *IEEE J. Solid-State Circuits*, vol.38, no.6, pp. 1059-1062, June 2003

[83] P. C. Yu, S. Shehata, A. Joharapurkar, P. Chugh, A. R. Bugeja, Xiaohong Du, Sung-Ung Kwak, Y. Papantonopoulous, T. Kuyel, “A 14 b 40 MSample/s pipelined ADC with DFCA,” *ISSCC Dig. Tech. Papers*, pp.136-137, 439, 2001

[84] Dong-Young Chang, Un-Ku Moon, “Radix-based digital calibration technique for multi-stage ADC,” *Proc. ISCAS*, vol.2, pp. II-796-II-799, 2002

[85] Yun-Shiang Shu, Bang-Sup Song, “A 15-bit Linear 20-MS/s Pipelined ADC Digitally Calibrated with Signal-Dependent Dithering,” *IEEE J. Solid-State Circuits*, vol.43, no.2, pp.342-350, Feb. 2008

[86] U. Eduri, F. Maloberti, “Online calibration of a Nyquist-rate analog-to-digital converter using output code-density histograms,” *IEEE Trans. Circuits and Systems I: Regular Papers*, vol.51, no.1, pp. 15-24, Jan. 2004

[87] Xiaoyue Wang, P. J. Hurst, S. H. Lewis, “A 12-bit 20-Msample/s pipelined analog-to-digital converter with nested digital background calibration,” *IEEE J. Solid-State Circuits*, vol.39, no.11, pp. 1799-1808, Nov. 2004

[88] J. McNeill, M. Coln, B. Larivee, "“Split-ADC” architecture for deterministic digital background calibration of a 16-bit 1-MS/s ADC," *IEEE J. Solid-State Circuits*, vol. 40, no. 12, pp. 2437–2445, Dec. 2005.

[89] Jen-Lin Fan, Chung-Yi Wang, Jieh-Tsorng Wu, "A Robust and Fast Digital Background Calibration Technique for Pipelined ADCs," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol.54, no.6, pp.1213-1223, June 2007

[90] R. Serrano-Gotarredona, L. Camuas-Mesa, T. Serrano-Gotarredona, J. A. Leero-Bardallo, B. Linares-Barranco, "The Stochastic I-Pot: A Circuit Block for Programming Bias Currents," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol.54, no.9, pp.760-764, Sept. 2007

[91] R.V. Hogg, J. W. McKean and A. T. Craig, "Introduction to Mathematical Statistics, Sixth Edition," pp. 238-253, *Pearson Prentice Hall,* 2005

[92] H. J. Ryser, "The Corus Mathematical Monographs: Number 14; Combinatorial Mathematics," *The Mathematical Association of America,* 1965

[93] Prof. M. B. Rao, University of Cincinnati, Private Communication

[94] Hui Pan, M. Segami, L.-C. Choi, A. A. Abidi, "A 3.3-V 12-b 50-MS/s A/D converter in 0.6-μm CMOS with over 80-dB SFDR," *IEEE J. Solid-State Circuits*, vol.35, no.12, pp.1769-1780, Dec 2000

[95] A. K. Ong, V. I. Prodanov, M. Tarsia, "A method for reducing the variation in "on" resistance of a MOS sampling switch," *Proc. ISCAS*, pp.437-440, vol.5, 2000

[96] M. Waltari, K. Halonen, "Bootstrapped switch without bulk effect in standard CMOS technology," *Electronics Letters*, vol.38, no.12, pp. 555-557, 6 Jun 2002

[97] D. Johns and K. Martin, "Analog Integrated Circuit Design," pp. 243, *John Wiley and Sons,* 1997

[98] B.Y.T. Kamath, R.G. Meyer and P.R. Gray, "Relationship between frequency response and settling time of operational amplifiers," *IEEE J. Solid-State Circuits*, vol.9, no.6 pp. 347- 352, Dec 1974

[99] S.Y. Robert Li, "Algebraic Switching Theory and Broadband Applications," *Academic Press*, 2001

[100] S.R. Norsworthy et. al., "Delta-Sigma Data Converters: Theory, Design, and Simulation," pp. 253-257, *IEEE Press*, 1997

[101] R. Naiknaware, H. Tang, T. S. Fiez, "Time-referenced single-path multi-bit ΔΣ ADC using a VCO-based quantizer," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.47, no.7, pp.596-602, Jul 2000

[102] Feng Chen, B. H. Leung, "A high resolution multibit sigma-delta modulator with individual level averaging," *IEEE J. Solid-State Circuits*, vol.30, no.4, pp.453-460, Apr 1995

[103] R. T. Baird, T. S. Fiez, "Linearity enhancement of multibit ΔΣ A/D and D/A converters using data weighted averaging," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.42, no.12, pp.753-762, Dec 1995

[104] I. Galton, "Spectral shaping of circuit errors in digital-to-analog converters," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.44, no.10, pp.808-817, Oct 1997

[105] L. R. Carley, J. Kenney, "A 16-bit 4th order noise-shaping D/A converter," *Proc. CICC*, pp.21.7/1-21.7/4, 16-19 May 1988

[106] C. Petrie, M. Miller, "A background calibration technique for multibit delta-sigma modulators," *Proc. ISCAS*, vol.2, pp.29-32 vol.2, 2000

[107] J. W. Fattaruso, S. Kiriaki, M. De Wit, G. Warwar, "Self-calibration techniques for a second-order multibit sigma-delta modulator," *IEEE J. Solid-State Circuits*, vol.28, no.12, pp.1216-1223, Dec 1993

[108] R. Schreier, B. Zhang, "Delta-sigma modulators employing continuous-time circuitry," *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications,* vol.43, no.4, pp.324-332, Apr 1996

[109] O. Shoaei, W. M. Snelgrove, "Design and implementation of a tunable 40 MHz-70 MHz Gm-C bandpass ΔΣ modulator," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol.44, no.7, pp.521-530, Jul 1997

[110] Y.-S. Shu, B.-S. Song, K. Bacrania, "A 65nm CMOS CT ΔΣ Modulator with 81dB DR and 8MHz BW Auto-Tuned by Pulse Injection," *ISSCC Dig. Tech. Papers,* vol. 51, pp. 500-501, Feb 2008

[111] J. A. Cherry, W. M. Snelgrove, "Clock jitter and quantizer metastability in continuous-time delta-sigma modulators," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing,* vol.46, no.6, pp.661-676, Jun 1999

[112] P. Benabes, P. Aldebert, A. Yahia, R. Kielbasa, "Influence of the feedback DAC delay on continuous-time bandpass ΣΔ coverter," *Electronics Letters*, vol.36, no.4, pp.292-294, 17 Feb 2000

[113] R. Mittal, D. J. Allstot, “Low-power high-speed continuous-time Σ-Δ modulators,” *Proc. ISCAS*, vol.1, pp.183-186 vol.1, 30 Apr-3 May 1995

[114] K. Nguyen, R. Adams, K. Sweetland, Huaijin Chen, “A 106-dB SNR hybrid oversampling analog-to-digital converter for digital audio,” *IEEE J. Solid-State Circuits*, vol.40, no.12, pp. 2408-2415, Dec. 2005