NOISE-SHAPING SAR ADCS

by

Jeffrey Alan Fredenburg

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)
in the University of Michigan
2015

Committee:

Professor Michael P. Flynn, Chair
Professor Zhong He
Professor Dave D. Wentzloff
Professor Zhengya Zhang

“Nobody tells this to people who are beginners; I wish someone told me. All of us who do creative work, we get into it because we have good taste. But there is this gap. For the first couple years you make stuff; it’s just not that good. It’s trying to be good, it has potential, but it’s not. But your taste, the thing that got you into the game, is still killer. And your taste is why your work disappoints you. A lot of people never get past this phase. They quit. Most people I know who do interesting, creative work went through years of this. We know our work doesn’t have this special thing that we want it to have. We all go through this. And if you are just starting out or you are still in this phase, you gotta know it's normal and the most important thing you can do is do a lot of work. Put yourself on a deadline so that every week you will finish one story. It is only by going through a volume of work that you will close that gap and your work will be as good as your ambitions. And I took longer to figure out how to do this than anyone I’ve ever met. It’s gonna take a while. It’s normal to take a while. You’ve just gotta fight your way through.”

-Ira Glass

PREFACE

Long gone are the days of VCRs, cassette tapes, floppy disks, camera film, and slide rulers. Cheap digital memory and ubiquitous processing platforms have completely revolutionized the way we interact with world around us. Indeed, the sheer amount of personal computing recourses available to human beings today should absolutely astonish anyone humbled enough to remember that familiar clunk of rotary dial phones and that helpful assistance of human telephone operators. Even the most modest personal electronic devices we carry in our pockets today – and sometimes toss aside like yesterday's old toys, easily dwarf the capabilities of electronics from decades past. In particular, look at the Apollo guidance computer from 1969. That simple computer hardly rivals even some of the calculators we have available today, but that brilliant device landed people on the Moon. From sticks and stones, to fire, to the wheel, to the compass, to the telephone, to the light bulb, to penicillin, into space, and through the internet, the rate at which technology is advancing is unparalleled by any other stage in human history. Digital memory and digital computing are no exception, and perhaps even one day, in the spirit of Richard Feynman [1], we might eventually encode the entire history of human knowledge into the surface of a single atom...

In all seriousness, we as human beings are only limited by the ingenuity of our imagination and that curiously unique drive to reach outwards beyond what we know – and of course, the laws of physics! Already, we construct circuits with physical dimension on the order of a hundred atoms, and as amazing as that is, what we've yet to discover and what we've yet to do

with those discoveries is only going to get more amazing. The ride is far from over yet, and whatever fantastic marvels the future holds, digital memory and digital processing will certainly play an important role in those wonderful things to come. Consider even for a moment the discovery of the Higgs Boson, or even other particles from the Standard Model – or the entire Large Hadron Collider[1] itself and every particle accelerator before. Without the mind-boggling amount of digital memory and computing resources available today, none of that would have ever happened. Sometimes we as academics, and as scientists and engineers, get so immersed in subtleties of our specialized fields that we often forget how life-changing what we do really can be… "If I have seen further it is by standing on the shoulders of giants [2]."

The prevalence of digital electronics will continue to spread throughout the many facets of our lives, and as access to cheaper computing resources become more commonplace, digital devices will need to become more autonomous and more interactive with the world around us. Digital devices, however, face a serious problem when interacting with the outside world, for that tiny digital world of '1's and '0's they hold so dear – even with it's perfect language of expressively complete logic, hardly describes anything beyond an abstraction of an abstraction of the physical world. Just flickers from shadows on a cave wall [3]. The real world isn't digital, it's analog – wonderfully continuous, and as often times seems, logically defiant right to its core. Sometimes it's as sharp as it is smooth, as coarse as it is fine, and twice two makes four – sometimes five [4], but still a great deal removed from that tiny world of '1's of '0's.

We, as human beings, perceive light and sound and heat and touch as continuous quantities, which is quite different from how digital electronics abstract the world. In order to reach out and

[1] Estimates suggest that the Large Hadron Collider generates 20 petabytes of data per year, 20×10^{15} bytes, and it's international computing grid is one of largest know clusters in the world.

interact with the physical world the same way as us, digital electronics require bilingual interpreters to explain the world – namely analog-to-digital converters. As *digital* as they say the world is becoming, *analog* is far from out the door and certainly here to stay.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER I

Introduction

1.1 Overview

This work investigates hybrid analog-to-digital converters (ADCs) that combine the phenomenal energy efficiency of successive-approximation (SAR) ADCs with the resolution enhancement strategies used by noise-shaping converters. Because charge-redistribution SAR ADCs contain few active components and rely on highly digital controllers, SAR ADCs demonstrate the best energy efficiencies of all low bandwidth, moderate resolution converters (~10 bits).

SAR ADCs achieve remarkable power efficiency at low resolution, but as the resolution of the SAR ADC increases, the specifications for input-referred comparator noise become more stringent and total DAC capacitance becomes too large, which degrades both power efficiency and bandwidth. For these reasons, lower resolution, lower bandwidth applications tend to favor traditional SAR ADC architectures, while higher bandwidth, higher resolution applications tend to favor pipeline-SARs. Although the use of amplifiers in pipeline-assisted SARs relaxes the comparator noise requirements and improves bandwidth, amplifier design becomes more of a challenge in highly scaled processes with reduced supply voltages.

In this work, we explore the use of feedback and noise-shaping to enhance the resolution of SAR ADCs. Unlike pipeline-SARs, which require high-gain, linear amplifiers, noise-shaping

SARs can be constructed using passive FIR filter structures. Furthermore, the use of feedback and noise-shaping reduces the impact of thermal *kT/C* noise and comparator noise. This work details and explores a new class of noise-shaping SARs.

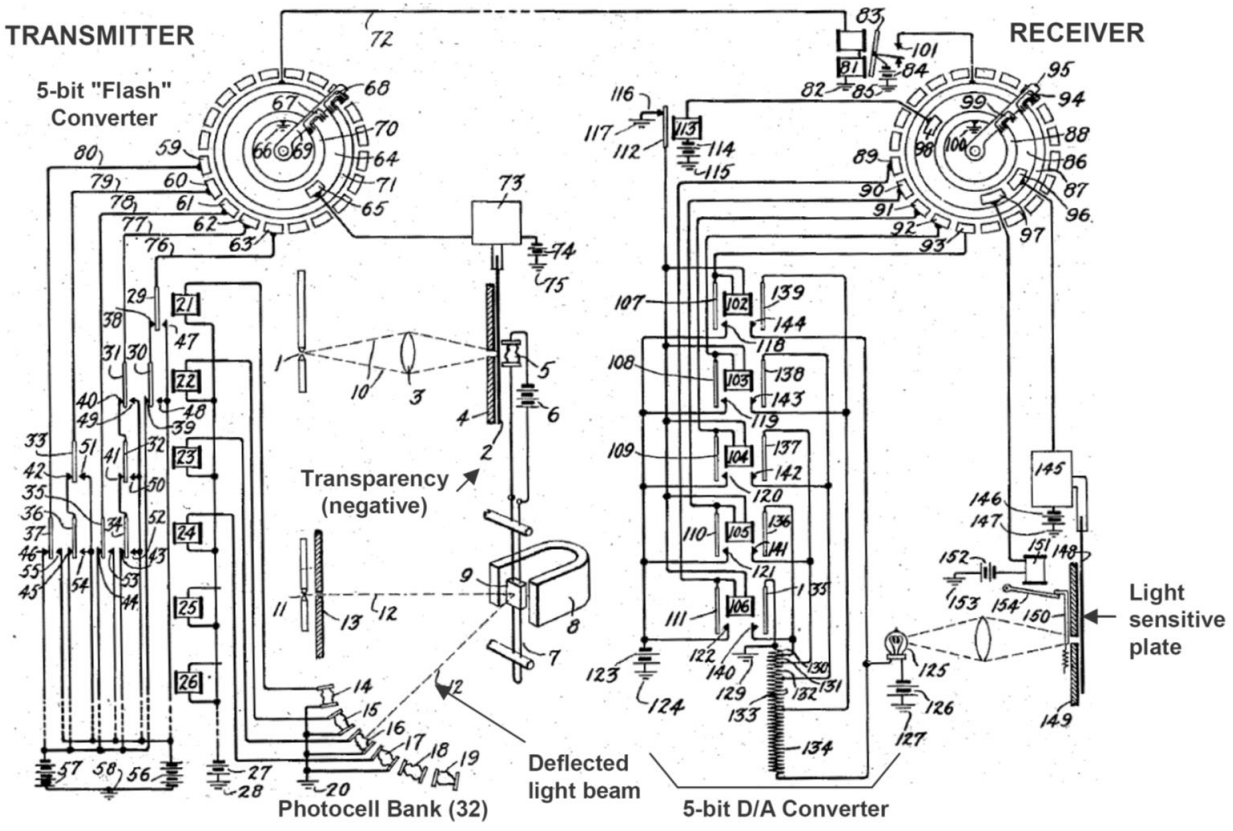

Figure 1-1: The First Disclosure of PCM: Paul M. Rainey, "Facsimile Telegraph System," *U.S. Patent 1,608,527, Filed July 20, 1921, Issued November 30, 1926* [5][2]

[2] Figure and caption quoted directly form Figure 1-5 in [1].

1.2 Comparison of Architectures

In this section we compare different ADC architectures and explore some of the fundamental ADC design tradeoffs. Figure 1-2 highlights the primary division between ADC architectures with respect to resolution and bandwidth. As sketched in Figure 1-2, ADCs exhibit a fundamental tradeoff between bandwidth and resolution across architecture types. At low bandwidths, delta-sigma ADCs dominate higher resolution applications, followed by successive-approximation, pipeline, and flash ADCs. Similar to the fundamental gain-bandwidth tradeoff for general purpose amplifiers, ADCs achieve the highest resolutions at lower bandwidths and accommodate higher bandwidths at the expense of lower resolutions.

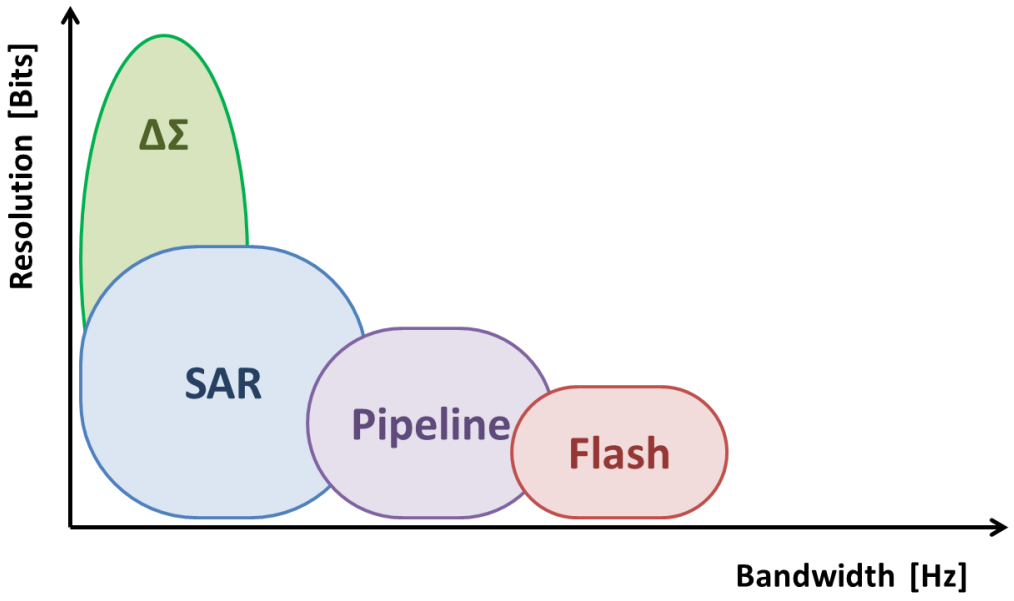

Figure 1-2: Overview of conventional ADC architectures

The tradeoff between resolution and bandwidth is further quantified in Figure 1-3, which surveys measured performance data from scholarly published ADCs [6]. As shown in Figure 1-3, ADC resolution, expressed in signal-to-noise-and-distortion (SNDR), decreases at higher signal bandwidths. Similar to the sketch of architectures provided in Figure 1-2, delta-sigma ADCs dominate high resolution applications and flash ADCs dominate high bandwidth applications. When examining the limits to this resolution-bandwidth tradeoff, an interesting trend arises. The solid and dashed trendlines which envelope the data points in Figure 1-3 represent surfaces of constant jitter. Empirically, sampling jitter can be shown to sets the upper limit on the maximum bandwidth achievable for a given resolution [7]-[10].

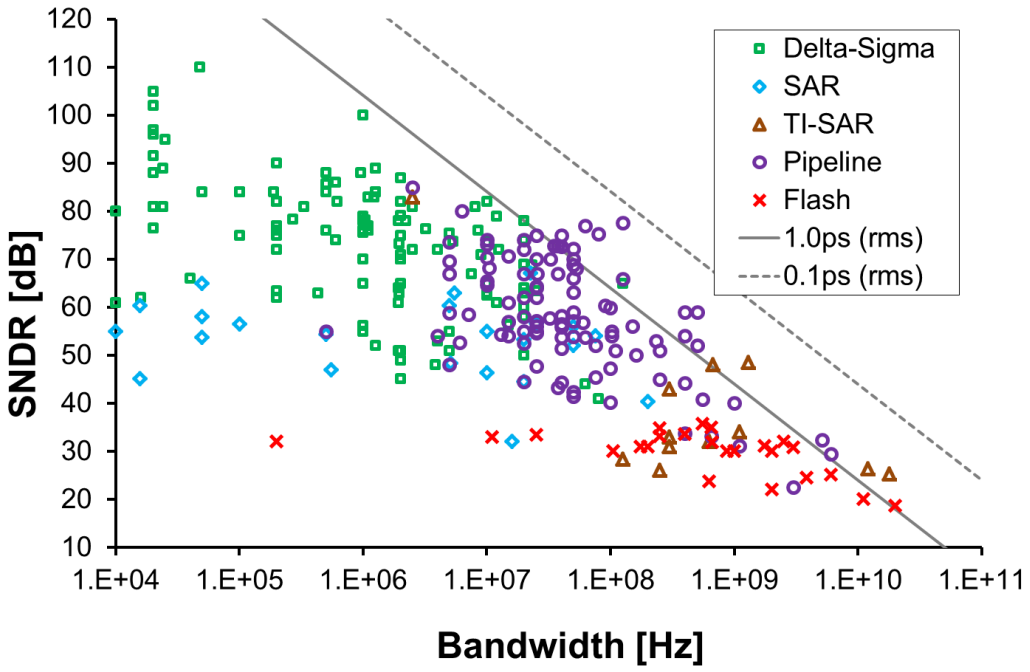

Figure 1-3: Survey of resolution versus bandwidth for various ADC architectures with data compiled from 2012 online survey [6].

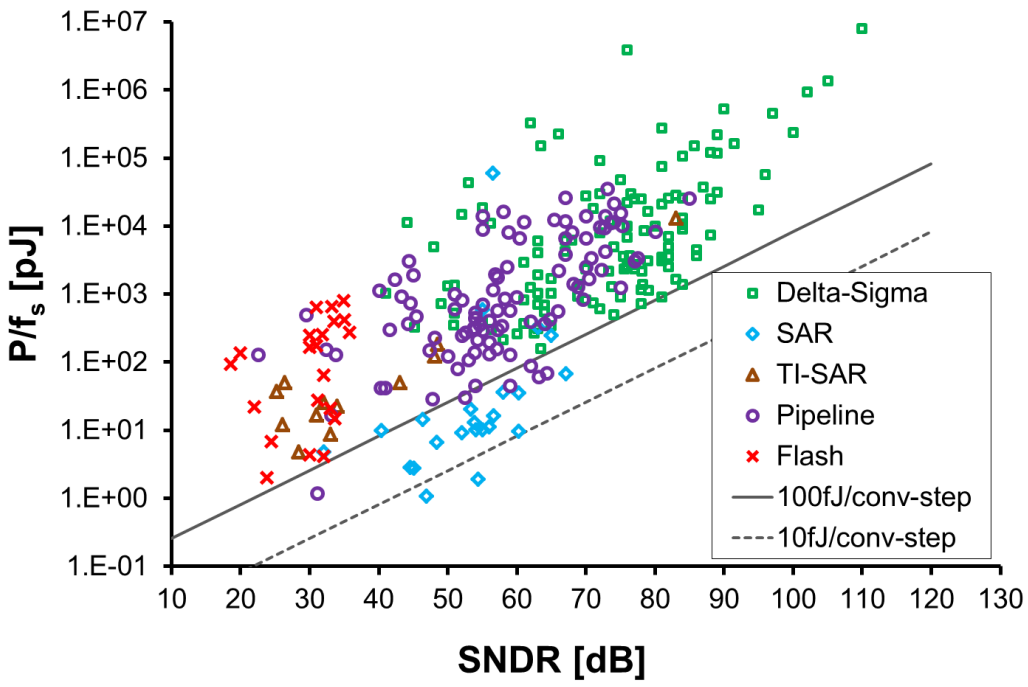

Figure 1-4: Survey of energy versus resolution for various ADC architectures with data compiled from 2012 online survey [6].

While the connection between the ADC resolution-bandwidth tradeoff and jitter can be understood as a limitation resulting from the sampling operation of ADCs, the tradeoff between ADC energy consumption and SNDR is less clear. Figure 1-4 plots measured performance data relating energy consumption and SNDR. As shown in Figure 1-4, the energy consumption of ADCs tends to increase as the resolution increases. At higher resolutions, delta-sigma ADCs typically consume the highest energies, and at the lowest resolutions, flash ADCs typically consume the lowest energy. For most of the architectures, Figure 1-4 shows a consistent tradeoff between energy and SNDR at a lower limit envelope around 100fJ/conv-step.[3] SAR ADCs, on

[3] Walden figure-of-merit expressed as Power/Bandwidth/2^{BITS} [7].

the other hand, seem to break this trend and follow and FOM envelope closer to 10fJ/conv-step. In terms of Walden figure-of-merit (FOM), SAR ADCs exhibit a much better energy performance than all other ADC architectures.

For completeness, ADC power consumption is compared with ADC bandwidth in Figure 1-5. Except for delta-sigma ADCs, the ADCs reported in the survey are roughly bounded by a surface of constant energy at about 1.0 pJ. Delta-sigma ADCs, however, appear to be roughly bounded at about 100 pJ, which is two orders of magnitude higher than the other ADC architectures. For delta-sigma ADCs, this higher energy bound most likely results from the way in which delta-sigma ADCs logarithmically trade bandwidth for resolution through oversampling.

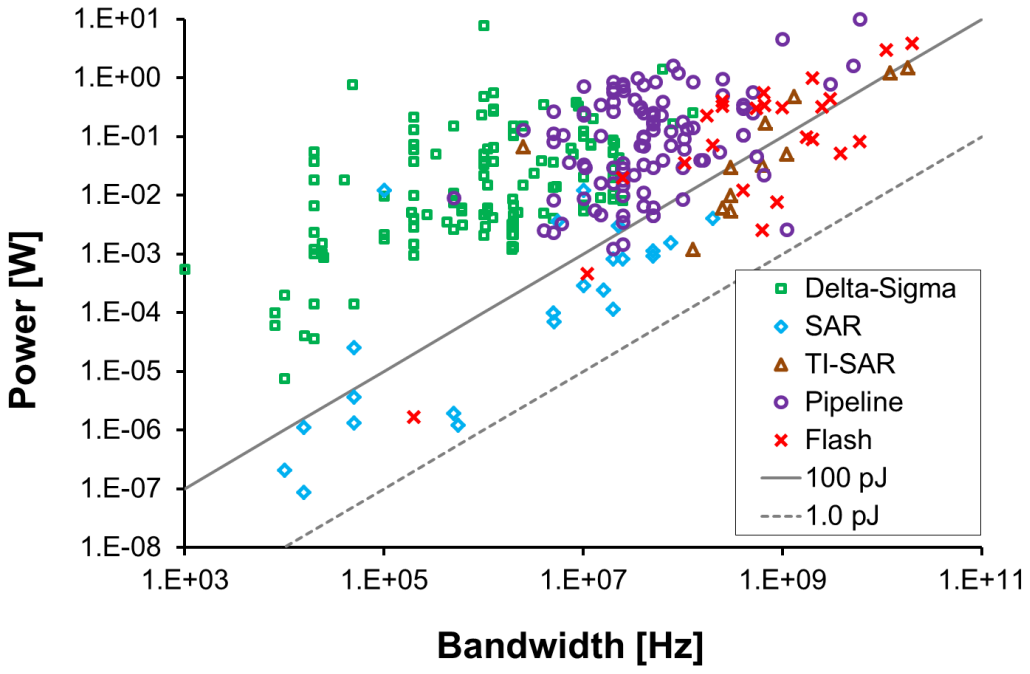

Figure 1-5: Survey of power versus bandwidth for various ADC architectures with data compiled from 2012 online survey [6].

Based on the surfaces of constant energy described in Figure 1-5, we should expect the energy consumption of ADCs to remain fairly flat across bandwidth. The picture becomes less clear, however, when we directly compare ADC energy to bandwidth – where energy is expressed as power normalized by bandwidth. Figure 1-6 provides a plot of ADC energy versus bandwidth. Except for SAR ADCs, which seem to reveal the expected independence between energy and bandwidth, the other architectures reveal a trend of higher energies at lower bandwidths. Typically one would expect lower bandwidth application to require less energy, so the design tradeoff implied by Figure 1-6 is probably not completely decoupled from other ADC specifications, such as SNDR.

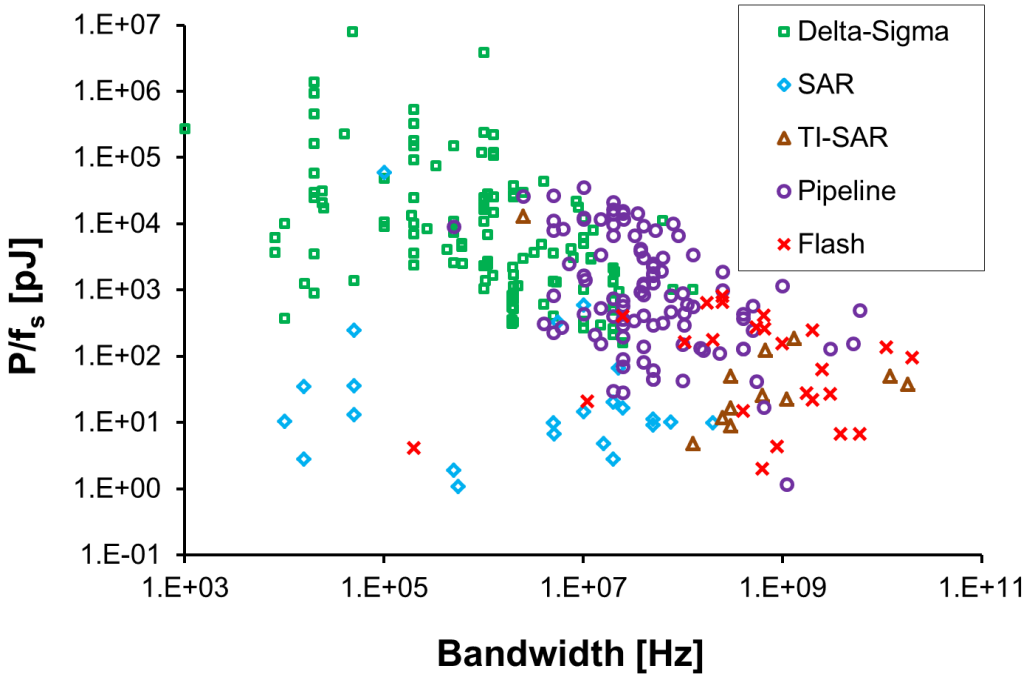

Figure 1-6: Survey of energy versus bandwidth for various ADC architectures with data compiled from 2012 online survey [6].

1.3 Time Interleaving

Traditionally, high speed applications required flash ADCs. Other architectures, such as SAR, were limited to lower bandwidth applications. In 1980, however, time-interleaving ADC architectures were introduced [11]-[12]. With time-interleaving, multiple low bandwidth ADCs are multiplexed together to achieve a higher effective converter bandwidth. As shown in Figure 1-7, time-interleaving allows SAR and pipeline ADCs to achieve bandwidths that were typically restricted to flash ADCs. Although time-interleaving has allowed architectures such as SAR to achieve overall bandwidths in excessive of 20 GHz, time-interleaving is a technique restricted to Nyquist ADCs. Oversampling converters, like delta-sigma ADCs, do not leverage the same bandwidth performance benefit form time-interleaving. This point is further discussed in further detail in Chapter 5.

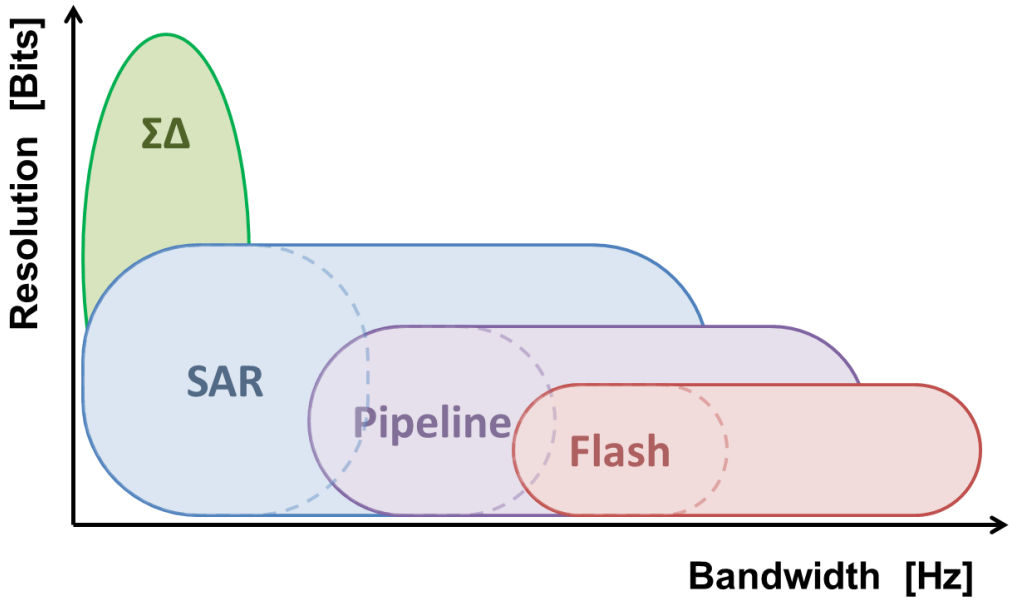

Figure 1-7: Overview of time-interleaved ADC architectures

In Figure 1-7, we see a gap for resolutions above time-interleaved SAR ADCs and for bandwidths beyond delta-sigma ADCs. To address this ADC architecture gap, this work explores hybrid ADC architectures that combine delta-sigma ADCs and SAR ADCs. As described in the rest of this work, these hybrid noise-shaping SAR ADCs leverage the high energy efficiency of the SAR ADC architecture and exploit the oversampling techniques of traditional delta-sigma ADCs to produce high-performance, energy efficient ADCs.

1.4 Outline

The rest of the work is divided as follows. In Chapter 2, the energy efficiency of charge-redistribution SAR ADCs is examined. Although an energy analysis of the digital SAR controller is omitted form the analysis, a detailed look at the energy consumption of the SAR capacitor DAC and comparator is presented in terms of the resolution and bandwidth constraints.

In Chapter 3, the effects of capacitor mismatch on ADC resolution and yield are examined. The analysis on capacitor mismatch is quite intensive, but the analysis is the first to relate resolution, yield, and mismatch for SAR ADCs with a complete closed-form statistical model.

In Chapter 4, we describe a hybrid noise-shaping SAR ADC which combines a SAR ADC with a switch cap FIR filter to produce a low energy, moderate resolution oversampling ADC.

In Chapter 5, we describe an extension to the noise-shaping SAR ADC described in Chapter 4, and present a noise-shaping ADC structure that combines a time-interleaved SAR ADC with a delta-sigma MASH ADC to produce a time-interleaved oversampling converter. Although the time-interleaved MASH ADC described in Chapter 5 does not solve all the issues for time-interleaving general delta-sigma ADCs, the time-interleaved MASH ADC leverages time-

interleaving to expand the bandwidth of noise-shaping SAR ADCs, which helps to fill some of that gap highlighted in Figure 1-7.

CHAPTER II

SAR ADC Energy Analysis

2.1 Capacitor DAC Switching Energy

This chapter derives an expression for the DAC switching energy of a SAR ADC. The derivation begins by analyzing the energy consumption of a binary weighted capacitor array during a single switching event and continues by summing these energy contributions across complete conversion cycles for a uniformly distributed input.

Figure 2-1 presents the model used to calculate the DAC switching energy. In this model, the differential DAC consists of two single ended capacitors arrays each comprised of 2^{N-1} unit capacitors – where N is the differential DAC resolution in bits. Furthermore, each half of the array uses both positive and negative reference voltages to permit addition and subtraction of voltage at the DAC output during conversion. SAR ADCs using similar DACs are found in [13]-[15].

In terms of implementation, this dual reference switching scheme is more complex than single reference switching schemes [15], but from an energy perspective, this switching operation is more efficient since each capacitor is charged by a reference voltage only once – whereas a DAC using a single reference voltage may need to switch a capacitor twice: once during a trial bit test and again when setting the final bit decision. Although recent literature shows a variety of other energy efficient DAC implementations, the energy efficiency of those

implementations are comparable to this DAC [15]. Furthermore, the settling times analysis for those more complicated DACs switching schemes is less straightforward.

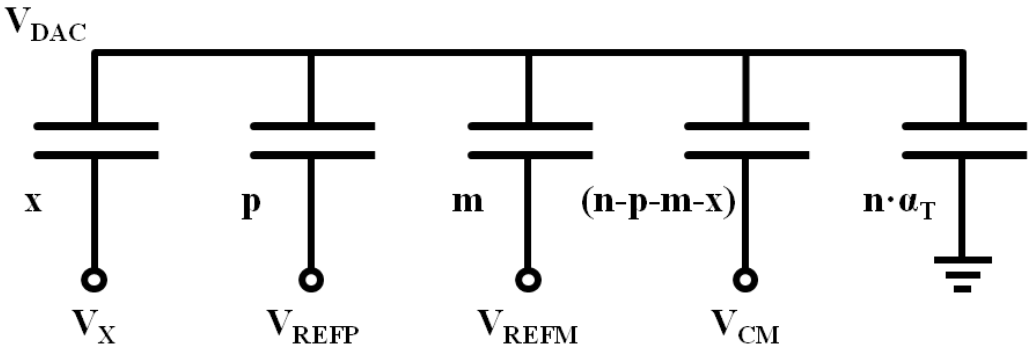

Figure 2-1: Single-ended circuit model used to calculate the DAC switching energy due to switching *x* capacitors from V_{CM} to either V_{REFP} or V_{REFM}. V_X describes final the switching reference voltage (either V_{REFP} or V_{REFM}), the quantities *x*, *p*, *m*, and *(n-p-m-x)* describe the number of unit capacitors at particular voltages, *n* is the total number of capacitors, and α_T describes the top plate parasitic capacitance normalized to the array capacitance.

We first calculate how the DAC output voltage responds to single switching events. As shown in Figure 2-1, the change in the DAC output voltage due to switching *x* capacitors through a voltage difference of ΔV_X is calculated by applying conservation of charge at the top plate of the DAC. Equating the charge on the top plate before and after switching the reference voltage for the *x* capacitors to ΔV_X, we derive (2.1).

$$\Delta V_{DAC} = \frac{x}{n(1+\alpha_T)} \Delta V_x \tag{2.1}$$

Equation (2.1) describes the change in the DAC output voltage after a single switching event – where ΔV_{DAC} describes the change in the DAC voltage, *n* is the total number of unit

capacitors, x is the number switched by a voltage difference of ΔV_X, and α_T is fractional top plate parasitic capacitance as normalized by the total array capacitance. These quantities are shown in Figure 2-1.

Next, we calculate the energy consumed by the reference voltages during switching events. During the switching of the x capacitors, transient currents flow from the reference voltages at each node and shuffle charge between the array capacitors in order to equalize the DAC top plate potential. The energy consumption of the DAC is calculated by integrating the power associated with each these currents over time.

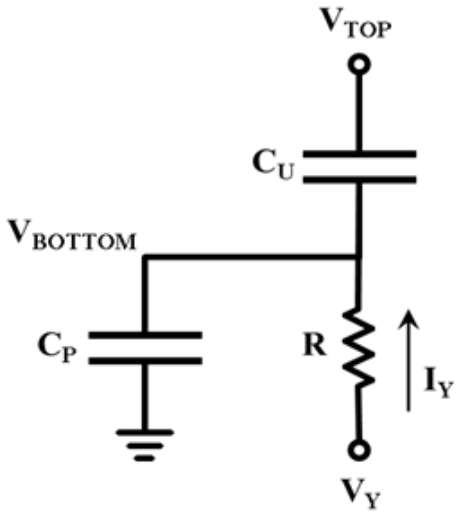

Figure 2-2: Circuit model used to calculate the switching energy of an arbitrary capacitor from the array. C_u is the unit capacitance, C_P is the bottom plate parasitic capacitance, R is the switch resistance, V_{TOP} is the top plate, V_{BOTTOM} is the bottom plate, V_Y is the node reference voltage (either V_{CM}, V_{REFP}, or V_{REFM}), and I_Y is the current delivered by V_Y.

Figure 2-2 presents an abstract model for calculating the switching energy contributed by each node in the array. Assuming the bottom plate of an isolated unit capacitor is connected to

some arbitrary voltage V_Y through a switch resistance R, the calculation of the energy contribution from this node is shown in (2.2).

$$\Delta E_Y = V_Y \int_0^\infty I_Y(t)\, dt \tag{2.2}$$

Since the time integral of current is charge, ΔE_Y is the product of V_Y and ΔQ_Y – where ΔQ_Y is the change in charge at the bottom plate node. Equation (2.3) describes the change in energy associated with switching the reference voltage to V_Y through a voltage difference of ΔV_Y. By convention, a positive ΔE_Y indicates energy consumption, and a negative ΔE_Y indicates energy recovery. Note that the sign of the energy in (2.3) can be either positive or negative depending on the signs of ΔV_Y and ΔV_{TOP}.

$$\Delta E_Y = [C_u(\Delta V_Y - \Delta V_{TOP}) + C_P \Delta V_Y] V_Y \tag{2.3}$$

Next, we iterate the results from (2.3) across each of the DAC nodes and calculate the energy for the entire differential DAC. During the switching of x capacitors, as described in Figure 2-1, the energy contributions at each node are calculated and summed. Assuming a differential array structure, which switches the references voltages oppositely, the energy contributions calculated from (2.3) are summed across every node for both halves of the differential array. The ΔV_{TOP} term in the resulting expression is simplified using (2.1) since ΔV_{TOP} is equal to ΔV_{DAC}.

The simplified result of this calculation is shown in (2.4), which describes the change in energy for the differential DAC during the switching of x capacitors in each array through a differential voltage of $\Delta V_{X,DIFF}$ – where $|\Delta V_{X,DIFF}| = \frac{1}{2} V_{FS}$, $V_{REFP} - V_{REFM} = \frac{1}{2} V_{FS}$, α_P is the bottom

plate parasitic capacitance normalized to a unit capacitance, α_T is the top plate parasitic capacitance normalized to the array capacitance, n is total number of capacitors in each half of the array, and the quantities p and m represent the number of capacitors connected to the positive and negative reference voltages as shown in Figure 2-1.

$$\Delta E = \Delta E_{Independent} + \Delta E_{Dependent}$$

$$\Delta E_{Independent} = \frac{C_u {V_{FS}}^2}{8}\left[x(1+\alpha_P) - \frac{x^2}{n(1+\alpha_T)}\right] \tag{2.4}$$

$$\Delta E_{Dependent} = \frac{C_u {V_{FS}}^2}{8}\left[\frac{x(m-p)}{n(1+\alpha_T)}\right] \cdot sign\left(\Delta V_{X,diff}\right)$$

Equation (2.4) is decomposed into two parts: a state independent energy term and a state dependent energy term. Since the energy dissipated by the reference voltages during a single switching event depends on the state of the DAC switches, the power dissipation of the DAC during a complete conversion cycle varies as a function of the sampled input voltage. The independent term describes the constant amount of energy required to switch x capacitors – independent of the DAC switch arrangement. The dependent term describes the excess switching energy that is either expended or recovered during a switching event as a function of the DAC switch configuration.

Figure 2-3 graphs the normalized energy contributions from the state independent and dependent terms summed throughout the switching events of a complete conversion cycle. As shown in Figure 2-3, the average state dependent energy contribution across the ADC codes is zero. Hence, the state dependent energy contribution for a uniformly distributed input signal is zero and can be neglected. Note that the average state dependent energy contribution for a

sinusoidally distributed input signal is negative since its probability density is higher at the outer codes than the middle codes. Therefore, the assumption of a uniformly distributed inputs result in a slightly pessimistic energy approximation when considering sinusoidally distributed inputs.

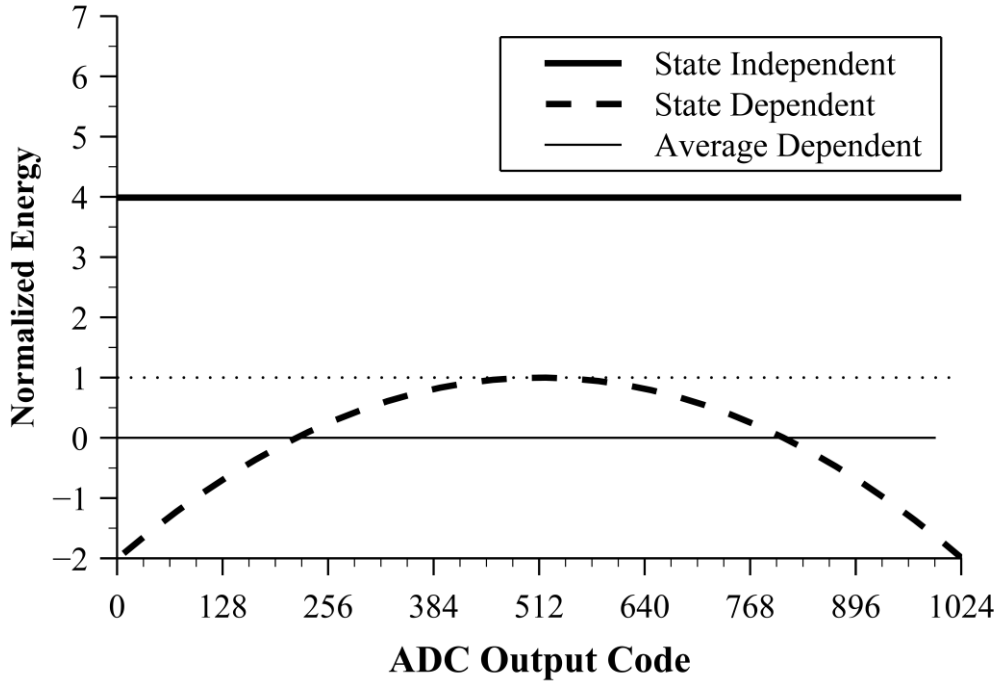

Figure 2-3: Normalized energy contributions versus 10 bit ADC output code for both the state dependent and state independent terms from (2.4) summed throughout the switching events of a complete conversion cycle. For comparison purposes, both, α_T and α_P are set to zero

Finally, we derive the average switching energy of the DAC. By summing each of the state independent switching energy contributions described by (2.4) throughout a complete conversion cycle, we calculate the average energy. Equation (2.5) describes the average energy consumption after completing the summation – where *N* is the number of bits. The parameter *x* from (2.4) is

summed in a descending binary fashion from 2^{N-2} down to 2^0 in order to capture the binary weighting of the DAC capacitors. Note that the summation proceeds from 2^{N-2} since the MSB capacitor in each half of the differential array consists of 2^{N-2} unit capacitors and only *N-1* switching events are needed during a complete conversion cycle.

$$\Delta E_{DAC} = \frac{C_u {V_{FS}}^2}{8}\left[(1+\alpha_P)(2^{N-1}-1) - \frac{2^{N-1}-2^{1-N}}{3(1+\alpha_T)}\right] \tag{2.5}$$

In the limit of large *N*, we can approximate (2.5) as (2.6),

$$\Delta E_{DAC} = \frac{2^N(1+\alpha_P)C_u {V_{FS}}^2}{16}\left[1 - \frac{1}{3(1+\alpha_T)(1+\alpha_P)}\right] \tag{2.6}$$

Equation (2.6) is further simplified by relating the DAC capacitance to the *kT/C* noise incurred during sampling. When a signal is sampled through bottom plate sampling, each unit capacitor acquires some noise voltage. The total sampling noise of the DAC is calculated by summing the noise power contributions from all the unit caps of the differential array, which yields (2.7) – where $\sigma_{sampling}$ is expressed as a fraction of an *LSB*. The noise power expression in (2.7) is just the kT/C noise of the sampled voltage.

$${\sigma_{sampling}}^2 = \frac{1}{LSB^2} \cdot \frac{kT}{2^N C_u} \tag{2.7}$$

Substituting the sampling noise expression from (2.7) into the DAC switching energy expression from (2.6) yields (2.8) – which describes the DAC energy in terms of the sampling noise.

$$\Delta E_{DAC} \cong \frac{kT}{16}\left(\frac{2^N}{\sigma_{sampling}}\right)^2 (1+\alpha_P)\left[1-\frac{1}{3(1+\alpha_T)(1+\alpha_P)}\right] \tag{2.8}$$

Equation (2.8) describes the average DAC switching energy across a full SAR conversion cycle for a uniformly distributed input. As shown in (2.8), the energy consumed by the DAC increases as the resolution, *N*, of the array increases. We can, however, express (2.8) in a more convenient form by expressing the energy in terms of the signal-to-noise ratio of the sampled DAC voltage. The definition of the signal-to-noise ratio is given by (2.9).

$$snr = \frac{Signal\ Power}{Noise\ Power} \tag{2.9}$$

If we express the signal power and noise power from (2.9) in terms a full scale sine-wave input signal and the sampled *kT/C* noise, we arrive at (2.10) – where V_{FS} is the full-scale voltage range of the DAC.

$$snr_{DAC} = \frac{\left(\frac{V_{FS}}{2\sqrt{2}}\right)^2}{\left(LSB \cdot \sigma_{sampling}\right)^2} \tag{2.10}$$

Since V_{FS} is equal to $2^N \cdot LSB$, we can further simplify this expression. The simplified expression is provided by (2.11).

$$snr_{DAC} = \frac{1}{8}\left(\frac{2^N}{\sigma_{sampling}}\right)^2 \tag{2.11}$$

Equation (2.11) expresses the signal-to-noise ratio of a sampled voltage on the capacitor DAC in the presence of *kT/C* noise. If we substitute (2.11) into (2.8), we can express the energy of capacitor DAC as function of the sampled signal-to-noise ratio and the parasitic capacitances for the top plate, α_T, and the bottom plate, α_P.

$$\Delta E_{DAC} \cong \frac{kT}{2} \cdot snr_{DAC} \cdot \left[(1+\alpha_P) - \frac{1}{3(1+\alpha_T)}\right] \tag{2.12}$$

As shown by equation (2.12) expresses, the DAC energy consumption increases linearly with both the available signal-to-noise ratio of the DAC and the bottom plate parasitic capacitance, α_P. Although the DAC energy increases as the top plate parasitic capacitance, α_T, increases as well, the increase in energy associated with α_T is a much weaker function than the signal-to-noise ratio or the bottom plate parasitic, α_P, and the increase in energy eventually approaches an asymptotic limit.

2.2 Capacitor DAC Settling Time

In this section, we calculate the DAC settling time and estimate the maximum sampling frequency of a SAR ADC. Using the circuit model presented in Figure 2-4, we first derive the time domain solution of the DAC output voltage during switching events. We then calculate the DAC settling time from this time domain expression and estimate the conversion time of a complete SAR cycle.

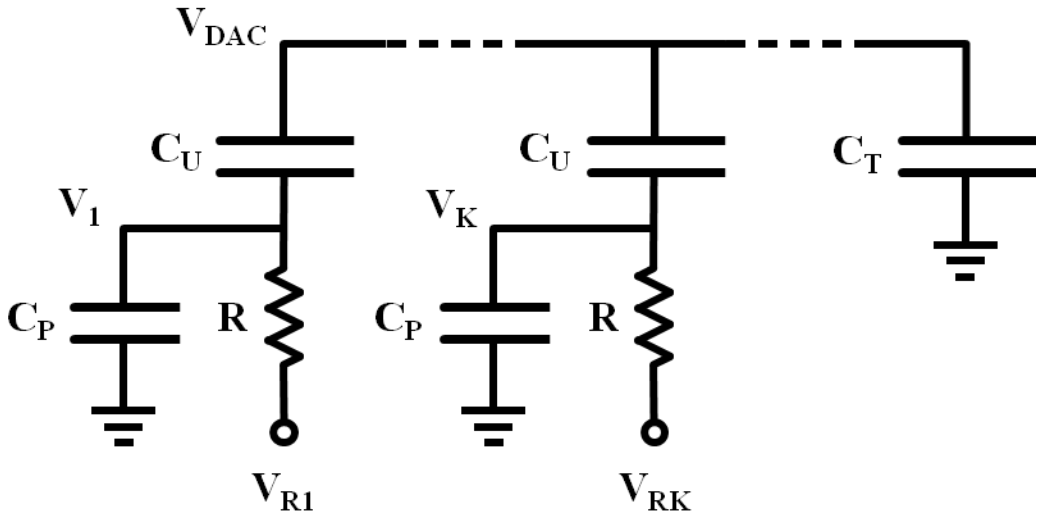

Figure 2-4: Circuit model used to calculate the DAC settling time. The bottom plate of each unit capacitor is modeled by a switch resistance, *R*, connected to a reference voltage in parallel with a bottom plate parasitic capacitance, C_P. For the *k-th* node, V_K is the bottom plate voltage, and V_{RK} is the node reference voltage. C_T is the top plate parasitic capacitance

Using the circuit model from Figure 2-4, we apply KCL at the DAC output and derive (2.13) – where C_T is the top plate parasitic capacitance, V_K is the bottom plate voltage of the k^{th} capacitor, and the summation shown in (2.13) is carried over each of the *n* nodes within the array.

$$\sum_{k=1}^{n} C_u\left(\dot{V}_{DAC} - \dot{V}_k\right) = -C_T\dot{V}_{DAC} \tag{2.13}$$

Solving (2.13) for $\dot{V}_K$, and substituting $C_T = n{\cdot}\alpha_T{\cdot}C_U$, we arrive at (2.14) – where α_T is the fractional top-plate parasitic capacitance of the array.

$$\sum_{k=1}^{n} \dot{V}_k = \mathrm{n}(1 + \alpha_T)\dot{V}_{DAC} \tag{2.14}$$

Equation (2.15) is the integral of (2.14) with *M* representing a constant of integration.

$$\sum_{k=1}^{n} V_k = \mathrm{n}(1+\alpha_T)V_{DAC} - M \tag{2.15}$$

Next, applying KCL at the bottom plate of the k^{th} node, we derive (2.16).

$$RC_u\left(\dot{V}_{DAC} - \dot{V}_K\right) = RC_P\dot{V}_K + (V_K - V_{RK}) \tag{2.16}$$

In order to obtain an expression for the V_{DAC}, we need to substitute the V_K terms from (2.14) and (2.15) into (2.16). A direct substitution, however, is not possible due to the summation, so to facilitate this substitution, we sum (2.16) over each of the *n* nodes in the array and substitute $C_P = \alpha_P \cdot C_{U\_}$ where αP is the fractional bottom-plate parasitic capacitance. The summation of (2.16) over *n* is given by (2.17).

$$\dot{V}_{DAC} = \frac{1}{n}\sum_{k=1}^{n}\left[(1+\alpha_P)\dot{V}_K + \frac{V_K - V_{RK}}{RC_u}\right] \tag{2.17}$$

Substituting the summations from (2.14) and (2.15) into the summation of (2.17), we obtain a first order differential equation describing the DAC output voltage as shown in (2.18).

$$\begin{gathered}\tau\dot{V}_{DAC} + V_{DAC} = \frac{1}{n(1+\alpha_T)}\left[M + \sum_{k=1}^{n} V_{RK}\right] \\ \tau = RC_u\left[\alpha_P + \frac{\alpha_T}{1+\alpha_T}\right]\end{gathered} \tag{2.18}$$

Assuming *x* of the node reference voltages switch by common voltage difference of ΔV_X at time, t = 0, and assuming that the final DAC voltage is determined by the ratio of capacitors as

described in (2.1) by *x* and *n*, we obtain the time domain solution for the DAC voltage shown in (2.19) – where ΔV_X is the change in the reference voltage, *x* is the number capacitors switched, *n* is the total number of unit capacitors in the array, α_P is the bottom plate capacitance normalized to the unit capacitance, and α_T is the top plate capacitance normalized to the array capacitance. Note that the constant of integration, *M*, cancels from the summation term when applying the two initial conditions described above.

$$\Delta V_{DAC}(t) = \frac{x}{n(1+\alpha_T)}\left[1 - e^{-t/\tau_{DAC}}\right]\Delta V_X$$
$$\tau = RC_u\left[\alpha_P + \frac{\alpha_T}{1+\alpha_T}\right] \tag{2.19}$$

The DAC transient response is described by Equation (2.19). For a large top plate capacitance – as is the case with extremely large top plate sampling switches or large comparator input transistors, the contribution from the α_T term approaches unity and the *RC* time constant becomes dominated by the unit capacitance and switch resistance. For a large bottom plate capacitance, the settling time is dominated by the *RC* time constant of the bottom plate parasitic and the switch resistance. Since α_P represents the bottom plate capacitance normalized to the unit capacitance and α_T represents the top plate capacitance normalized to the capacitance of the entire array, the bottom plate parasitic capacitance dominants the DAC settling time dynamics.

Next, we calculate the time required for the DAC voltage to settle to within ½*LSB* of the final output value using (2.19). The inequality describing this condition is shown in (2.20). Note that the effective *LSB* of the DAC is attenuated by the top plate capacitance since input voltages acquired through bottom plate sampling will also experience this attenuation.

$$\Delta V_{DAC}(\infty) - \Delta V_{DAC}(t_{settle}) \leq \frac{V_{FS}}{2n(1+\alpha_T)} \tag{2.20}$$

Assuming $\Delta V_X = \pm \frac{1}{2} V_{FS}$, we solve for t_{settle} as shown in (2.21) – where x is the number of capacitors switched in each half of the array.

$$t_{settle} \geq RC_u \left[\alpha_P + \frac{\alpha_T}{1+\alpha_T}\right] \ln x \tag{2.21}$$

The *½LSB* settling time is plotted in Figure 2-5. We see that the worst-case settling time occurs during the first switching transition when the number of switched capacitors in each array is largest – that is, $x = 2^{N-2}$.

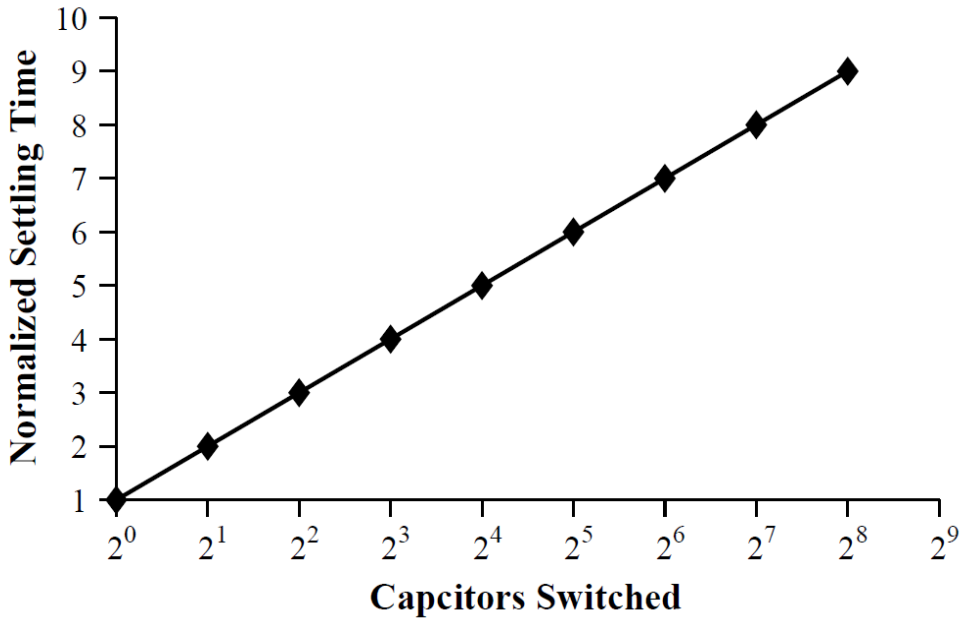

Figure 2-5: Normalized *½LSB* settling time from (2.21) versus the number of capacitors switched in each array for a 10 bit SAR ADC. The required settling time increases linearly on log scale with the number of caps switched.

The maximum settling time of the DAC is given in (2.22) – where N is the resolution of the ADC in bits.

$$t_{settle,min} = RC_u \left[\alpha_P + \frac{\alpha_T}{1+\alpha_T}\right](N-2)\ln 2 \tag{2.22}$$

The maximum sampling frequency of the ADC is approximated using the settling time described by (2.22). During a SAR conversion cycle, each bit is processed through a sequence of comparator operations and DAC switching events. If we assume the SAR cycle proceeds synchronously through each of the N bit trials with $t_{settle,min}$ allocated to complete each comparator decision and another $t_{settle,min}$ for each DAC settling event, the time required for a complete SAR conversion cycle is approximately $2N \cdot t_{settle,min}$. From this, the maximum sampling frequency is calculated as given by (2.23). Since the input sampling time is not included, the sampling frequency shown in (2.23) is optimistic.

$$\begin{aligned} {F_S}^{-1} &= 2N(N-2)\tau_{DAC}\ln 2 \\ \tau_{DAC} &= RC_u\left[\alpha_P + \frac{\alpha_T}{1+\alpha_T}\right] \end{aligned} \tag{2.23}$$

Equation (2.23) describes the maximum sampling frequency of the ADC as function of the ADC resolution and the DAC settling time constant – where F_S is the sampling frequency, N is the resolution of the ADC in bits, R is the switch resistance, C_U is the DAC unit capacitance, α_T is the top-plate parasitic capacitance normalized to the entire array, and α_P is the bottom-plate parasitic capacitance normalized to the unit capacitance C_U. These quantities are shown in Figure 2-4.

2.3 Comparator Energy Model

In this section, we derive an expression for the minimum energy consumption of a regenerative comparator. The goal of this derivation is to estimate the comparator energy consumption in a general manner such that the final expression does not depend on technology parameters such as threshold voltage, transition frequency, transconductance, etc. In this analysis, we estimate the lower bound of the comparator energy consumption in terms of the ADC bit resolution and the comparator input referred noise.

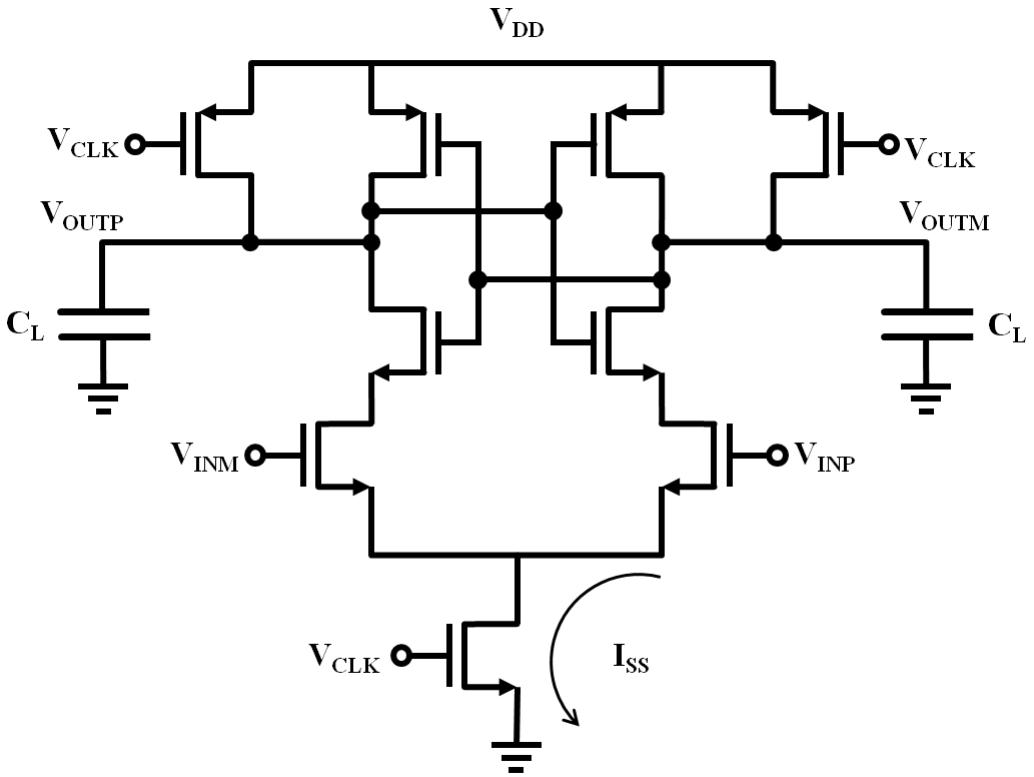

Figure 2-6: Simplified dynamic comparator schematic from [16]. The key features of this comparator are a latched output and a clocked reset.

Figure 2-6 shows a general comparator schematic that we use as a starting point for this analysis. Although the comparator is a simplified version of the dynamic comparator [16], the particular circuit topology is not important. For our purposes, the key features of this comparator are a capacitive load, a clocked reset, and a latching output.

The comparator in Figure 2-6 has two distinct phases of operation: a reset phase and an amplification stage. In the reset phase, the load capacitors are pre-charged to V_{DD}, and in the amplification phase, the input voltage difference is amplified by the positive feedback latching structure. During each of these operations, charge is transferred from the supply to the load capacitors. By estimating the energy consumption of these charge transfers, we estimate the overall energy consumption of the comparator.

We first calculate the energy associated with the reset phase. At the beginning of the reset phase, we assume that one of the output capacitors is fully charged to V_{DD} and other is completely discharged. Therefore, during reset, charge is only transferred to one capacitor. The energy consumed by the supply during this charging process is shown by (2.24).

$$\Delta \mathrm{E}_{RESET} = C_L {V_{DD}}^2 \tag{2.24}$$

Next, we calculate the energy consumed during the amplification phase. Since the positive feedback latch structure at the output dominates the comparator behavior during amplification, we simplify the energy calculation by estimating only the energy consumption of a latch. Figure 2-7 presents the latch schematic used in this calculation. Figure 2-8 presents the latch small signal model.

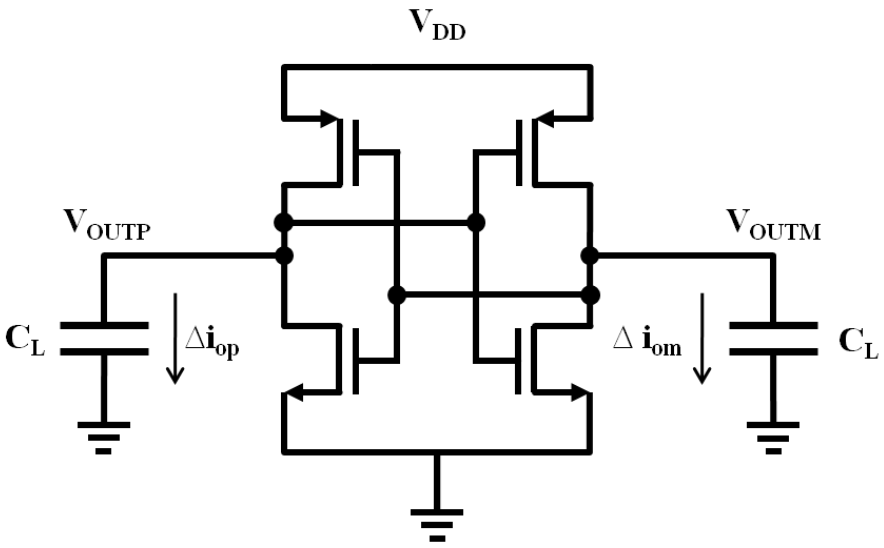

Figure 2-7: Simplified model of the comparator during the amplification phase. Assuming the latch outputs are initially balanced at $V_{DD}/2$, Δi_{OP} and Δi_{OM} describe the initial conditions generated from the input voltage.

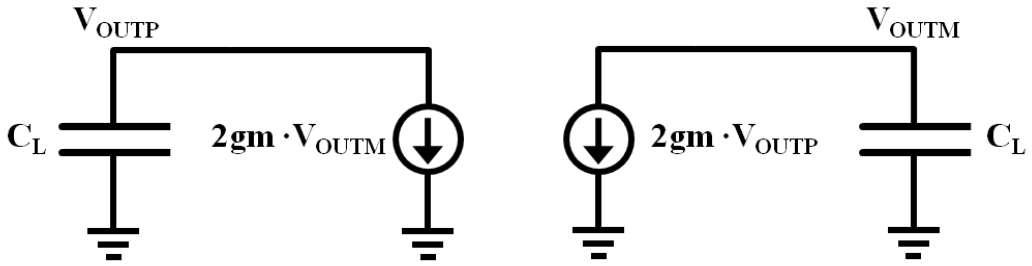

Figure 2-8: Small signal model of the latch from Figure 2-7. For simplicity, output resistances are neglected and all transconductances are assumed equal

The energy consumed by the latch is described by (2.25) – where $I_{DD,MAX}$ is the peak supply current and Δt is time interval of the latching operation.

$$\Delta \mathrm{E}_{LATCH} \leq I_{DD,MAX} V_{DD} \Delta t \tag{2.25}$$

When the latch outputs in Figure 2-7 are held at mid-rail, the overdrive voltages of the internal gates are largest and $I_{DD,MAX}$ is sourced from the supply. Summing the currents flowing through the top devices at this bias point, we approximate the peak supply as shown in (2.26) – where V_{GS} is approximated as $V_{DD}/2$ and we simply neglect the threshold voltage, V_{TH}.

$$I_{DD,MAX} \cong g_m |V_{GS} - V_{TH}| < \frac{1}{2} g_m V_{DD} \tag{2.26}$$

To estimate the time interval over which latching occurs, we derive a time domain expressions for the latch output and calculate the time until the outputs saturate. Applying KCL to the small signal model in Figure 2-8, we derive the coupled system of differential equations shown in (2.27).

$$\begin{aligned} \dot{V}_{OUTP} + \frac{2 \cdot g_m}{C_L} V_{OUTM} &= 0 \\ \dot{V}_{OUTM} + \frac{2 \cdot g_m}{C_L} V_{OUTP} &= 0 \end{aligned} \tag{2.27}$$

We decouple the system in (2.27) by substituting the derivative of each equation into the other. The decoupled system is shown in (2.28).

$$\ddot{V}_{OUTP} - \left(\frac{4 \cdot g_m}{C_L}\right)^2 V_{OUTP} = 0$$
$$\ddot{V}_{OUTM} - \left(\frac{4 \cdot g_m}{C_L}\right)^2 V_{OUTM} = 0 \tag{2.28}$$

Because (2.28) involves second order differential equations, the solution requires two sets of initial conditions. For the first set, we assume the initial output voltages are both zero, and for the second set, we assume that derivatives of the output voltages are established by initial currents flowing through the load capacitors. As shown in Figure 2-7, the initial currents are defined as Δi_{op} and Δi_{om}. Using these initial conditions, the solution to (2.28) is provided in (2.29).

$$V_{OUTP} = \left(\frac{\Delta i_{OP}}{2 \cdot g_m}\right) \sinh\left(\frac{2 \cdot g_m}{C_L} t\right)$$
$$V_{OUTM} = \left(\frac{\Delta i_{OM}}{2 \cdot g_m}\right) \sinh\left(\frac{2 \cdot g_m}{C_L} t\right) \tag{2.29}$$

In a general sense, the system of equations in (2.29) expresses the latch output voltages in terms of the initial capacitor currents. To relate this latch analysis back to a comparator, we assume that the initial currents are established by the comparator input voltage. Assuming the input signal is connected through differential input pair as shown in Figure 2-6, the initial capacitor currents are related to the input signal through the input pair transconductances as in (2.30). For convenience, we assume that the transconductances of the differential input pair transistors equal the transconductances of the latch transistors.

$$\Delta i_{OP} = +\frac{1}{2} g_m \Delta V_{IN}$$
$$\Delta i_{OM} = -\frac{1}{2} g_m \Delta V_{IN} \tag{2.30}$$

Finally, we obtain an expression for the differential output by substituting (2.30) into (2.29) and subtracting the resulting expressions. Equation (2.31) describes the differential output of the latch.

$$\Delta \mathrm{V}_{OUT} = \frac{1}{2} \sinh\left(\frac{2 \cdot g_m}{C_L} t\right) \Delta V_{IN} \tag{2.31}$$

Since the latch saturates when the differential output voltage equals to $\pm V_{DD,}$ we set the output voltage in (2.31) to V_{DD} and solve for the latching time Δt as described in (2.32).

$$\Delta \mathrm{t} = \frac{C_L}{2 \cdot g_m} \operatorname{arcsinh}\left(\frac{2 \cdot V_{DD}}{\Delta V_{IN}}\right) \tag{2.32}$$

Finally, we obtain an estimate of the latch energy consumption by substituting both the latching time interval (2.32) and the $I_{DD,MAX}$ expression from (2.26) into the energy expression (2.25). Equation (2.33) describes the latch energy consumption as function of the comparator input voltage magnitude.

$$\Delta E_{LATCH}(\Delta \mathrm{V}_{IN}) = \frac{1}{4} C_L {V_{DD}}^2 \operatorname{arcsinh}\left(\frac{2 \cdot V_{DD}}{\Delta V_{IN}}\right) \tag{2.33}$$

Assuming the magnitude of input signal to the latch is uniformly distributed between 0 and $V_{DD}/2^m$ – where m defines some arbitrary binary weighted scaling factor, we calculate the

average energy consumption as a function of *m* by averaging (2.33) across the input signal range as shown in (2.34). During a complete SAR conversion cycle, the magnitude of the comparator input range is halved after each comparison. Therefore, the input range is defined using 2^m in order to accommodate the binary scaling of the input signal magnitude during a conversion cycle.

$$\Delta E_{LATCH}(m) = \frac{2^m C_L {V_{DD}}^2}{4} \int_0^{2^{-m}V_{DD}} \operatorname{arcsinh}\left(\frac{2 \cdot V_{DD}}{\Delta V_{IN}}\right) d(\Delta V_{IN}) \qquad (2.34)$$

The integral in (2.34) is easier to compute numerically than to solve explicitly; therefore, we substitute the unitless parameter *u* for the argument of the *arcsinh* in the integrand. The result of this substitution is shown in (2.35).

$$\Delta E_{LATCH}(m) = \frac{2^m C_L {V_{DD}}^2}{4} \int_0^{2^{-m}} \operatorname{arcsinh}\left(\frac{2}{u}\right) du \qquad (2.35)$$

Equation (2.35) describes the average energy consumption of the latch as a function of a binary scaled input range. To calculate the average latch energy consumed after *N* comparisons, we sum the energy contributions across the binary scaling parameter *m* from 0 to *N-1* as shown in (2.36).

$$\Delta E_{LATCH} = \frac{C_L {V_{DD}}^2}{4} \sum_{m=0}^{N-1} 2^m \int_0^{2^{-m}} \operatorname{arcsinh}\left(\frac{2}{u}\right) du \qquad (2.36)$$

When deriving the latch energy expression in (2.36), we assumed that the latch was initially biased at mid-rail. During a comparison, however, the output voltages must discharge from the

V_{DD} reset value to the latch common mode voltage before the positive feedback amplification can occurs. As the current ramps up from zero to $I_{DD,MAX}$, this discharging process will consume energy.

If we consider the summation term in (2.36) as a measure of how efficiently the latch transfers charge to the load capacitance during transitions from the peak supply current down to zero current, we can approximate the energy efficiency of this initial discharging process during a transition from zero current to peak supply current as roughly equal to (2.36). Therefore the total energy consumed by the comparator after N comparisons is approximately the sum of N reset energy contributions (2.24) and twice the latch energy contribution form (2.36). Equation (2.37) describes the total energy consumption of the comparator.

$$\Delta E_{COMP} = \Gamma_N \cdot \mathrm{C}_L \cdot {\mathrm{V}_{DD}}^2$$
$$\Gamma_N = N + \frac{1}{2}\sum_{m=0}^{N-1} 2^m \int_0^{2^{-m}} \operatorname{arcsinh}\left(\frac{2}{u}\right) du \tag{2.37}$$

The Γ_N scaling parameter from (2.37) includes a complicated integral. Although an analytical solution exists for this integral, a simpler approximation is easier to manipulate and interpret. Equation (2.38) presents a quadratic approximation using the coefficients for a least-squares fit rounded to convenient fractions.

$$\Gamma_N \cong \frac{4}{23}[N^2 + 12N + 2] \tag{2.38}$$

Figure 2-9 compares the accuracy of the numerical approximation to Γ_N. As shown in Figure 2-9, the approximation in (2.38) slightly overestimates Γ_N.

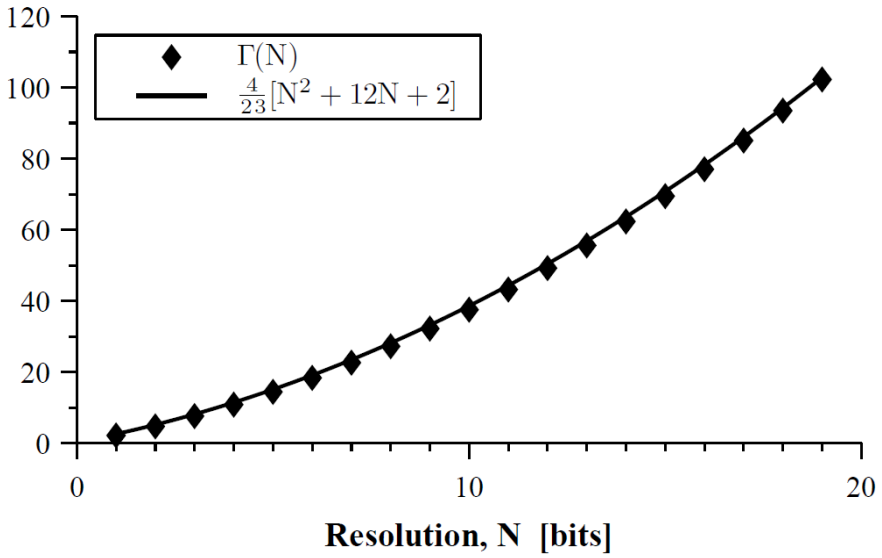

Figure 2-9: Plot of Γ_N (2.37) and the quadratic approximation (2.38). The coefficients of the quadratic approximation are the values obtained from a least-squares fit rounded to convenient whole numbers. As shown above, the quadratic approximation slightly overestimates Γ_N over the range of resolution.

Lastly, we remove the C_L dependency from (2.37) by relating C_L to input referred noise of the comparator. A comprehensive transient, noise analysis of the dynamic comparator can be found in [16], which describes the input referred comparator as scaled *kT/C* noise. Although less accurate than [16], we will approximate the input-referred noise as the simply the sum of the two *kT/C* noise powers contributed from each latch output. An estimate for the input referred comparator noise normalized to LSB^2 is shown in (2.39) – where σ_{COMP} is in bits, N is the resolution, and the differential input range, V_{FS}, is approximated as $2V_{DD}$.

$$\sigma_{COMP}{}^2 \leq \frac{2kT}{C_L}\left(\frac{2^N}{2 \cdot V_{DD}}\right)^2 \tag{2.39}$$

Substituting the input referred noise expression (2.39) and the numeric approximation (2.38) into (2.37), the comparator energy is expressed as a function of the ADC resolution and the comparator input referred noise.

$$\Delta \mathrm{E}_{COMP} \cong \frac{2kT}{23}\left(\frac{2^N}{\sigma_{COMP}}\right)^2 [N^2 + 12N + 2] \tag{2.40}$$

Equation (2.40) describes the energy consumed by the comparator as a function of the ADC resolution and the input referred comparator noise. Similar to the derivation for (2.12) from the DAC energy section, we can relate 2^N and σ_{COMP} to the available signal-to-noise ratio from the comparator and express (2.40) in the more convenient form given by (2.40).

$$\Delta E_{COMP} \cong \frac{2kT}{3} \cdot snr_{COMP} \cdot \ [N^2 + 12N + 2] \tag{2.41}$$

As shown in (2.41), the energy required to operate the comparator across all bit trials is linearly dependent on the available signal-to-noise ratio of the comparator and is a quadratic function of the resolution, *N*, as expressed in bits. The linear dependence between the comparator energy and the available signal-to-noise ratio from the comparator derives, to the first order, from the load capacitance, C_L – see Figure 2-6 and Figure 2-7.[4]

[4] The noise at the output is inversely proportional to C_L, kT/C, and the signal-to-noise ratio is inversely proportional to noise, hence the linear dependence of the signal-to-noise ratio to C_L.

CHAPTER III

Capacitor Mismatch[5]

3.1 Introduction

SAR ADCs offer an attractive solution in low power applications. Due to the inherent energy efficiency of charge redistribution DACs and the leveraged benefits of scaling [17], SAR ADCs can provide power efficient analog to digital conversion in systems that require moderate resolution and speed. However, specific applications have specific needs, and to ensure those needs are met, it is important for designers to have complete understanding of the design tradeoffs in the key building blocks of SAR ADCs, such as the capacitor DAC, the comparator, and the successive approximation registers.

It is well established that mismatch degrades the overall performance of ADCs, and various techniques have been proposed to overcome this degradation [18]-[24]. However, a precise formulation of the relationship between mismatch, the effective number of bits (ENOB), and yield is still lacking. In practice, an ADC designer may need to target a particular ENOB specification, but when estimating the yield, only indirect metrics such as integral nonlinearity (INL) or differential nonlinearity (DNL) are available. Although ENOB, INL, and DNL are important indicators of ADC performance, ENOB is a better indicator of the *overall* system level

[5] The material in this chapter on capacitor mismatch was first presented in [55].

performance, and with the yield expressions derived in this paper, ADC designers can easily target system level performance objectives.

The use of INL as a yield metric for data converters is prevalent in literature, but has limited utility in *overall* system design. Although the bulk of the analytic work regarding yield has focused on developing INL yield models for current-steering DACs [25]-[29] in the presence of transistor drain current mismatch [30], the major results of these works are also generally applicable to ADCs. According to [29], the analytical development of INL as a yield metric begins with [25], where the maximum deviation of the INL is introduced as a measure for distinguishing between good and bad current-steering DACs. Later, in [26]-[29], we see a progression of refinements aimed towards improving the statistical accuracy of INL based yield estimates. However, none of these works [25]-[29] offer a detailed comparison between INL yield measurements and other performance metrics such as signal-to-noise-plus-distortion ratio (SNDR), and it is unclear how to precisely interpret INL based yield estimates when targeting a specific ENOB yield for ADCs and DACs.

Examples of analyses relating INL/DNL to ENOB can be found in [31]-[34], but these works do not contain a detailed statistical treatment relating ENOB and yield. In [31], DNL is related to signal-to-noise ratio (SNR) by considering DNL errors as an additive noise in flash ADCs. In [32], SNDR is related to INL errors as a function of the input signal probability density function (PDF). In [33], ENOB is related to INL through a harmonic analysis for thermometer-coded structures, and in [34] an approximate relationship between ENOB and INL is given for resistor strings based on analysis in [35]. Although these works provide a convenient sketch relating ENOB and DNL/INL, it is unclear how to extract accurate ENOB yield information.

In this chapter, we develop an alternative statistical model using ENOB as a yield metric. First, we examine the effects of mismatch in a binary weighted, charge redistribution SAR ADC. We then derive an exact algebraic formulation relating capacitor mismatch to the average noise power of the ADC output, and from this algebraic formulation, we derive ENOB as a function of capacitor mismatch. Next, we explore the statistics of this ENOB expression and develop a statistical expression that predicts yield in terms of ENOB and mismatch. Finally, we generalize the results of this work by presenting a compact design equation, which accurately relates resolution, mismatch, and ENOB to yield for all binary weighted ratiometric converters. The design equation offered is accurate to within ±0.17 bits for yield values between 0.5% and 99.5% and is consistent with standard test methodology.

Section 3.2 analyzes the effects of mismatch and derives an expression for ENOB as a function of capacitor mismatch. Section 3.3 explores the statistics of this ENOB expression, and Section 3.4 formulates an expression for yield. Section 3.5 develops a compact design equation for yield, ENOB and mismatch, which generalizes the results of this work.

3.2 Analytical ENOB Derivation

In this section, we derive an analytic expression for the ENOB of a binary weighted SAR ADC in terms of capacitor mismatch. Although we derive this expression from the perspective of a capacitor DAC, our results are equally valid form the perspective of the ADC. We begin this derivation by relating the INL errors of a DAC to its average noise power. Next, we formulate an expression for the INL in terms of capacitor mismatch parameters, and use this relationship to express the mismatch induced noise power as a function of the capacitor mismatch. Finally, we

translate the mismatch induced noise expression into an analytical expression for ENOB which supports differential sinusoidal signals and is consistent with standard ADC test methodology.

3.2.A Mismatch Induced Noise Power

The relationship between the mismatch induced noise power and INL can be derived in a manner similar to the calculation of ideal quantization noise power. By including INL errors into this calculation, we can capture the noise power contributed from INL errors.[6] Figure 3-1 shows the transfer function of a DAC and its corresponding noise voltage with and without INL errors.

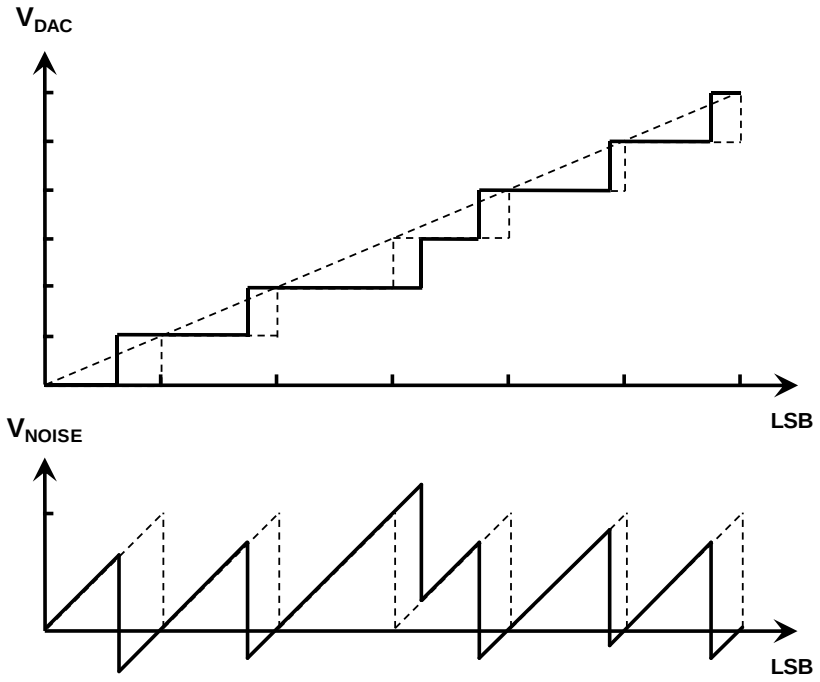

Figure 3-1: Transfer function and residual noise voltage of a capacitor DAC with mismatch (solid) and without mismatch (dashed). Without mismatch, the code transitions and DAC outputs occur in regular LSB intervals.

[6] A related result in [26] expresses the average noise power of a flash ADC to its DNL errors.

For an ideal single-ended DAC without mismatch, the established $LSB^2/12$ quantization noise expression represents the mean-squared value of the output noise voltage [33]. Assuming the DAC output codes are uniformly distributed, we can calculate this quantity as shown in (3.1) and (3.2) – where N is the DAC resolution in bits, Λ is the LSB, u is the output noise voltage, and u_0 is the mean output noise voltage.

$$V_{noise}{}^2 = \frac{1}{2^N\Lambda}\sum_{i=0}^{2^N-1}\int_0^{\Lambda}(u-u_0)^2 du = \frac{\Lambda^2}{12} \tag{3.1}$$

$$u_o = \frac{1}{2^N\Lambda}\sum_{i=0}^{2^N-1}\int_0^{\Lambda} u\, du = \frac{\Lambda}{2} \tag{3.2}$$

We incorporate mismatch into this expression by modifying the limits of integration in (3.1) to include the INL errors of the DAC. Since the i-th code transition voltage of a mismatched DAC is offset from the ideal transition voltage by the INL error of that code, we offset the integration limits in (3.1) by the INL error as shown in (3.3) – where Φ_i is the INL error of the i-th code expressed in *LSB*.

$$V_{noise}{}^2 = \frac{1}{2^N\Lambda}\sum_{i=0}^{2^N-1}\int_{\Lambda\Phi_i}^{\Lambda(1+\Phi_{i+1})}\left(u-\frac{\Lambda}{2}\right)^2 du \tag{3.3}$$

Evaluating the integral in (3.3) and simplifying the resulting expression, we obtain an expression for the noise power in terms of the INL, which is given by (3.4).[7] A more intuitive formulation of (3.4) is also presented in (3.5).

7 The DC power contributed by the INL errors is not removed from (3.4).

$$V_{noise}{}^2 = \frac{\Lambda^2}{12} + \frac{\Lambda^2}{2^N}\sum_{i=0}^{2^N-1}\Phi_i{}^2 \tag{3.4}$$

$$V_{noise}{}^2 = \frac{\Lambda^2}{12} + \Lambda^2\, Mean(\Phi^2) \tag{3.5}$$

In the limit of a large *N*, the contribution from the mismatch induced noise power can be approximated as the variance of the INL as shown in (3.6).[8]

$$V_{noise}{}^2 = \frac{\Lambda^2}{12} + \Lambda^2\, Var(INL) \tag{3.6}$$

Expressions (3.4)-(3.6) describe the average noise power of a single-ended DAC as the sum of the ideal quantization noise and the mean square of the INL errors. These results are generally applicable to all ADCs and DACs with both fixed quantization levels and uniformly distributed DAC outputs and indicate that nonlinearities in the quantization levels manifest as an additive noise. This conclusion is also suggested in [31] for DNL errors.

3.2.B Analytic Formulation of DNL and INL

We continue by formulating an expression for the noise power contributed by the INL errors as a function of capacitor mismatch. To this end, we first introduce a capacitor mismatch model and then derive expressions for DAC DNL errors in terms of this model. Finally, we convert these DNL expressions into INL expressions and solve for the mean squared INL in terms of the capacitor mismatch parameters.

8 A similar result to (3.6) is derived in [27] using a probabilistic approach.

We model the mismatch of capacitors within the array as an additive random error – i.e., $C = C_{nom}+\Delta C$, where C_{nom} is the nominal design capacitance and ΔC is a normally distributed random error with zero mean with σ_c^2 variance. Furthermore, we define a mismatch parameter γ to describe the fractional error of each binary weighted capacitor group from its ideal value. Assuming that the capacitors are carefully arranged, we neglect pathological errors and effects from spatial gradients.[9] [36]

The mismatch model is provided in (3.7) – where 2^N is the total number of capacitors in the array, C_u is the average unit capacitance of the array, C_i is the capacitance of the i-th binary weighted capacitor group, and γ_i is the associated fractional mismatch of the i-th group. In addition, we let $i = N$ represent the MSB, $i = 1$ the LSB, and $i = 0$ the termination capacitor. Note that the effective unit capacitance, C_u, is distinct from the nominal design capacitance, C_{nom}.

$$C_i = 2^{i-1}C_u(1+\gamma_i)$$
$$C_u = \frac{1}{2^N}\sum_{j=1}^{2^N}\left(C_{nom} - \Delta C_j\right) \tag{3.7}$$

Since the sum of the binary weighted capacitors, defined in (3.7) as C_i, must equal the total capacitance of the array, the weighted sum of the fractional mismatch parameters γ_i must sum to zero. This condition is enforced by (3.8).[10]

$$\gamma_0 + \sum_{i=1}^{N} 2^{i-1}\gamma_i = 0 \tag{3.8}$$

[9] An analysis relating INL errors to spatial gradients is found in [31].

[10] Although based on the physical construction of the capacitor array, the constraint on γ given by (3.8) also ensures that gain errors in the transfer curve are not counted as distortion since (3.8) imposes a unity gain for the DAC transfer curve.

Using the capacitor mismatch model defined in (3.7) and (3.8), we now relate the DNL errors of the DAC to the fractional mismatch parameter γ_i. The DNL error of a DAC can be expressed by (3.9) – where ΔV_{DAC} is the difference between successive DAC output voltages [37].

$$DNL = \frac{\Delta \mathrm{V}_{DAC} - LSB}{LSB} \tag{3.9}$$

Furthermore, we can express the DAC output voltages in terms of the binary weighted capacitors as shown in (3.10) – where N is the resolution in bits, Λ is the LSB, C_i is the i-th binary weighted capacitor group, and $b_i \in \{0,1\}$ represents the digital bits in the DAC code.

$$V_{DAC} = \sum_{i=1}^{N} \frac{b_i C_i}{C_u} \Lambda \tag{3.10}$$

Substituting the expression for C_i from (3.7) into (3.10), we relate the DAC output voltage to the fractional mismatch parameter γ_i as in (3.11).

$$V_{DAC} = \sum_{i=1}^{N} b_i 2^{i-1} (1 + \gamma_i) \Lambda \tag{3.11}$$

Using the DAC output voltage expression in (11) and the definition for DNL given in (3.9), we calculate the DNL errors for each of the 2^N DAC codes. For an N bit, single-ended, binary weighted capacitor DAC, however, the DNL errors are uniquely determined by N distinct DNL values, and these N values represent the DNL error at the major code transitions – specifically, codes 2^{i-1} where $i \in \{1,\ldots,N\}$.

Intuitively, we can understand why the DAC has only N unique DNL by examining the odd numbered codes. Since all the odd numbered codes have a binary representation ending in one, the difference in the DAC output voltage between these codes and one code less is determined solely by the DAC LSB capacitor. Therefore, the DNL error for every odd code is the same and is equal to the DNL error for code 2^0, which is an odd code. Using similar examples, we can show through induction that only N unique values are needed to describe the entire DNL of the DAC and these unique values are equal to the DNL at the major code transitions.

We now calculate the DNL errors at the major code transitions by substituting (3.11) into (3.9) – where ΔV_{DAC} from (3.9) is the difference in the DAC output voltages between codes 2^{i-1} and 2^{i-1}-1. An expression for the N unique DNL values is provided in (3.12) – where d_i represents the DNL error at code 2^{i-1}, and $i \in \{1,..,N\}$.

$$d_i = 2^{i-1}\gamma_i - \sum_{j=1}^{i-1} 2^{j-1}\gamma_j \tag{3.12}$$

The distribution of the DNL values given in (3.12) across each of the DAC codes can be described by the recursively ordered set shown in (3.13) – where D_N is ordered set of DNL values, and d_N, as described by (3.12), represents the DNL at the most significant code in the level of hierarchy. The arrangement of DNL values given by (3.13) describes a sequence in which the N unique DNL values are distributed across the DAC codes in an "x modulo 2^{N-1}" manner.

$$D_N = \{D_{N-1} \quad d_N \quad D_{N-1}\} \tag{3.13}$$

As an example of how (3.13) describes the DNL distribution, we consider a 3 bit DAC. For N = 3, the arrangement of the DNL errors for this DAC is shown in (3.14) – where d_i is again described by (3.12).

$$D_3 = \{d_1 \quad d_2 \quad d_1 \quad d_3 \quad d_1 \quad d_2 \quad d_1\} \tag{3.14}$$

With both the DNL values and their arrangement calculated, we now relate the INL to the DNL and work towards expressing the mean-squared INL in terms of the mismatch parameter γ_i. The relationship between INL and DNL is shown in (3.15) [37] – where Φ_i is the INL error at code i, and δ_j is the DNL error at code j. Furthermore, δ_j assumes one of the values described by (3.12) in an order determined by (3.13).

$$\Phi_i = \sum_{j=1}^{i} \delta_j \tag{3.15}$$

Substituting the DNL expression from (3.12) into (3.15) and simplifying the resulting summation by exploiting the inherent folding symmetry of (3.13), we derive the mean-squared INL in terms of the mismatch parameter γ_i. The simplified result is shown in (3.16).

$$Mean(\Phi^2) = \frac{1}{4}{\gamma_0}^2 + \frac{1}{4}\sum_{i=1}^{N}\left(2^{i-1}\gamma_i\right)^2 \tag{3.16}$$

Substituting this expression for the INL noise power from (3.16) into the noise power expression from (3.5), we obtain an explicit expression for the average noise power of an N bit single-ended DAC with capacitor mismatch, which is shown in (3.17) – where Λ is the LSB, and

γ_i is the fractional mismatch of the *i*-th capacitor group as defined in (3.7). Furthermore, we let $i = N$ represent the MSB, $i = 1$ the LSB, and $i = 0$ the termination capacitor.

$$V_{noise}{}^2 = \frac{\Lambda^2}{12} + \frac{\Lambda^2}{4}\left[\gamma_0{}^2 + \sum_{i=1}^{N}\left(2^{i-1}\gamma_i\right)^2\right] \tag{3.17}$$

The expression given in (3.17) describes the average noise power of a binary weighted DAC as the sum of the ideal quantization noise and a linear combination of the γ_i mismatch parameters squared. Similar to INL and DNL, the mismatch parameter γ_i manifests as additive noise.

3.2.C Differential Conversion

Since most high-performance SAR ADCs process differential signals, we now convert the noise power expression given by (3.17) from a single-ended result into a differential result. If we imagine constructing an *N* bit, differential DAC using two *N-1* bit, single-ended DACs, each with identical mismatch and opposite polarity[11], the average noise power of this composite differential DAC is the average of the two single-ended DAC noise powers. Using the results from (3.17) to describe the noise powers of the two *N-1* bit single-ended DACs and averaging, we obtain the noise power of an *N* bit, differential, binary weighted DAC as given in (3.18) – where Λ now describes the differential LSB, γ_i is the composite fractional mismatch of the *i*-th capacitor groups, and $\gamma_{i,p}$ and $\gamma_{i,m}$ are the individual mismatch parameters from the positive and negative arrays.

11 This DAC structure represents a generic sign/magnitude encoded structure utilizing a fixed common-mode output.

$$V_{noise}{}^2 = \frac{\Lambda^2}{12} + \frac{\Lambda^2}{4}\left[\gamma_0{}^2 + \sum_{i=1}^{N}\left(2^{i-1}\gamma_i\right)^2\right]$$
$$\gamma_i = \frac{1}{2}\left(\gamma_{i,p} + \gamma_{i,m}\right) \quad i \in \{0, \cdots, N-1\} \tag{3.18}$$

Equation (3.18) presents an exact algebraic solution for the average noise power of a binary weighted *N* bit, differential DAC with uniformly distributed INL errors. Furthermore, since the differential DAC output voltages are perfectly symmetrical about the origin, the noise power given by (3.18) is zero mean. Additionally, the constraint on γ_i given by (3.8) properly accounts for gain errors throughout the development of (3.18).

3.2.D Analytic Formulation of ENOB

We now formulate an expression for ENOB in terms of the mismatch parameter γ_i. For a perfectly matched DAC, only the quantization errors contribute noise and the average noise power is LSB^2/12, as shown in (3.1). If we define an effective LSB size, which generates an average noise power equivalent to the noise power of a mismatched DAC, we can explicitly relate ENOB to the average noise power of the mismatched DAC as is done in (3.19) – where V_{FS} is the differential full scale range of the DAC output voltage, and Λ_{eff} is the effective LSB size.

$$V_{noise}{}^2 = \frac{\Lambda_{eff}{}^2}{12}$$
$$\Lambda_{eff} = V_{FS} \cdot 2^{-ENOB} \tag{3.19}$$

Substituting the differential noise expression from (3.18) into (3.19), we can relate the ENOB of the DAC to the mismatch parameters γ_i. Solving this resulting expression for ENOB, we obtain (3.20).

$$ENOB = N - \log_4\left[1 + 3{\gamma_0}^2 + 3\sum_{i=1}^{N}\left(2^{i-1}\gamma_i\right)^2\right]$$
$$\gamma_i = \frac{1}{2}\left(\gamma_{i,p} + \gamma_{i,m}\right) \quad i \in \{0, \cdots, N-1\} \tag{3.20}$$

Equation (3.20) offers an exact analytic expression relating the ENOB of an N bit, differential, binary weighted capacitor DAC to capacitor mismatch for uniformly distributed signals. Although we derived (3.20) from the perspective of a SAR ADC, the result provided in (3.20) is applicable to all binary weighted ratiometric converters.[12]

3.2.E Correction for Sinusoidal Distributions

The ENOB expression given in (3.20) assumes that the DAC codes are uniformly distributed. In practice, however, the ENOB of an ADC is typically measured using a sinusoidal input signal, not a uniformly distributed signal. With a full-scale, uniformly distributed signal, all of the INL errors across the entire code range each contribute equally to noise. On the other hand, since sinusoidal signals tend to dwell more near their peaks than their mean, the INL errors at the outer codes contribute a larger fraction of the noise than the INL errors near the center codes. Therefore, the noise power contributed by INL errors depends on the probability distribution of the signal.[13]

To reconcile the ENOB expression in (3.20) with this preferred sinusoidal testing method, we introduce the scalar correction factor, α, to convert the ENOB expression given by (3.20) into an equivalent expression describing the ENOB of a sinusoidally distributed input signal. The

12 For binary weighted ratiometric converter without an explicit termination element, γ_0 is still defined as in (3.8), but should instead be interpreted as either the mean of the single-ended INL errors or a description of the INL induced gain error of the converter transfer function.

13 If the INL error were constant across the code range, the INL induced noise power would be independent of the signal distribution. This is why the quantization noise does not need to be scaled. By definition, however, the INL errors across an extended code range must sum to zero and therefore cannot remain constant across the codes.

modified ENOB expression is given in (3.21)– where γ_i represents the composite fractional mismatch parameter of the binary weighted capacitor groups as defined in (3.18) and (3.7), and α is approximated as the ratio between the INL noise contributions from a sinusoidal distribution and a uniform distribution.[14] A derivation for the estimated value of α used in (3.21) is offered in APPENDIX A.

$$ENOB = N - \log_4\left[1 + 3\alpha{\gamma_0}^2 + 3\alpha\sum_{i=1}^{N}\left(2^{i-1}\gamma_i\right)^2\right]$$
$$\alpha = \frac{3(4-\pi)}{\pi} \cong 0.8197 \tag{3.21}$$

Equation (3.21) provides an accurate estimate for the ENOB of N bit, differential, binary weighted ratiometric converters, which is consistent with the standard sinusoidal testing of the ADCs and DACs. Had we not introduced the correction factor α, the ENOB expression would overestimate the mismatch induced noise power by 18%.[15] Using (3.21), we can now accurately estimate the ENOB of a sinusoidal distribution over a wide range of γ_i values and compare results with standard ADC and DAC test measurements.

3.3 Statistical ENOB Derivation

Section 3.2 provides an analytic expression relating ENOB and mismatch (3.21), and in this section, we examine the statistics of this ENOB expression. First, we derive the probability density functions (PDF) for the single-ended mismatch parameters $\gamma_{i,p}$ and $\gamma_{i,m}$. Next, we use

[14] Alternatively, the ENOB of sinusoidally distributed DAC codes can be derived by replacing the "averaging" in (3.3) with the probability mass function (PMF) of a sinusoidal distribution, but it is unclear whether a tractable ENOB expression can be obtained due to the complexity of the sinusoidal PMF.

15 Since α linearly scales only the mismatch induced noise power, the 18% overestimation can be approximated by 1-α.

these PDFs for $\gamma_{i,p}$ and $\gamma_{i,m}$ to derive the PDF for the differential, composite parameter γ_i, and subsequently, the PDF for the square of γ_i. Finally, we combine these results with the ENOB expression given by (3.21) and obtain a statistical expression for ENOB. Finally, we compare this expression to results from numerical ADC simulations.

3.3.A PDF for the Single-Ended Mismatch Parameter γ

In the capacitor mismatch model presented in (3.7), each capacitor is modeled as $C=C_{nom}+\Delta C$, where C_{nom} is the nominal design capacitance and ΔC is a normally distributed error with zero mean with σ_c^2 variance. The PDF for C is shown in (3.22) – where the PDF is expressed using the notation $f_C(c)$.

$$f_C(c) = \frac{1}{\sqrt{2\pi\sigma_c{}^2}} exp\left[\frac{-(c - C_{nom})^2}{2\sigma_c{}^2}\right] \tag{3.22}$$

Furthermore, both the binary weighted capacitor groups and the total array capacitance can be represented as sums of the individual capacitors. Since the sum of independent normal random variables is itself normal with a mean and variance equal to the sum of the constituent means and variances, we obtain the marginal PDFs for the binary weighted capacitors directly from (3.22) as given in (3.23) – where Ns is the single-ended resolution, and X_i is the capacitance of the i-th binary weighted capacitor group in one of the single-ended arrays. To avoid parametric equations, we will omit the PDF of the termination capacitor and note that the distribution for the termination capacitor, X_0, follows the same distribution as the LSB capacitor, X_1.

$$f_{X_i}(x) = \frac{1}{\sqrt{2\pi{\sigma_i}^2}} exp\left[\frac{-(x-\mu_i)^2}{2{\sigma_i}^2}\right]$$
$$\mu_i = 2^{i-1}C_{nom} \quad i \in \{1, \cdots, N_S\} \tag{3.23}$$
$${\sigma_i}^2 = 2^{i-1}{\sigma_c}^2$$

Similarly, we derive the PDF for the total single-ended array capacitance from (22) as shown in (3.24) – where *Ns* is the single-ended resolution, and *W* is the total capacitance for one of the single-ended arrays.

$$f_W(w) = \frac{1}{\sqrt{2\pi{\sigma_W}^2}} exp\left[\frac{-(w-\mu_W)^2}{2{\sigma_W}^2}\right]$$
$$\mu_W = 2^{N_S}C_{nom} \tag{3.24}$$
$${\sigma_W}^2 = 2^{N_S}{\sigma_c}^2$$

Using the definition of γ_i from the mismatch model given in (3.7), we next reformulate γ_i in terms of the new variables X_i and W as shown in (3.25). For convenience, we will denote the single-ended fractional mismatch parameter with γ_i. When we derive the composite mismatch parameter, we will clarify the notation with $\gamma_{i,p}$ and $\gamma_{i,m}$.

$$\gamma_i = 2^{(N_S+1-i)}\frac{X_i}{W} - 1 \tag{3.25}$$

As shown in (3.25), the PDF for γ_i is determined by ratio of two *dependent* normal variables, X_i and W, which results in a prohibitively complicated expression for the PDF.[16] In order to simplify this PDF into a form amenable to further analysis, we will therefore expand (3.25) and approximate the capacitance of the array, W, as a constant in the denominator. The expansion of (3.25) is given by (3.26) with W approximated as $2^{N_S}C_u$ in the denominator[17] – where C_u is the mean capacitance of the array as defined in equation (3.7).

$$\gamma_i = \frac{2^{(N_S+1-i)}X_i - W}{2^{N_S}C_u} \tag{3.26}$$

Using the PDFs for X_i and W from (3.23) and (3.24), we now derive an approximation of the marginal PDF for γ_i through the expansion given by (3.26). The simplified PDF for γ_i is provided in (3.27) – note that the correlation between X_i and W in the numerator has not been neglected.

$$\begin{aligned} f_{\Gamma_i}(\gamma) &\cong \frac{1}{\sqrt{2\pi{\sigma_i}^2}} exp\left[\frac{-\gamma^2}{2{\sigma_i}^2}\right] \\ {\sigma_i}^2 &\cong \left(2^{1-i} - 2^{-N_S}\right)\left(\frac{\sigma_c}{C_{nom}}\right)^2 \quad i \in \{1, \cdots, N_S\} \end{aligned} \tag{3.27}$$

We now calculate the PDF for the composite mismatch factor. Using (3.27) to describe the distributions for γi,p and γi,m, we obtain the PDF for composite mismatch factor using the relationship for the mismatch factors given by (3.20), which states that the composite mismatch factor is the average of the single-ended mismatch factors. The PDF for the composite mismatch

16 An exact formulation of this PDF is derived in [30] to analyze nonlinearities in resistor strings.

17 Approximating W as $2^{N_S}C_u$ follows from the weak law of large numbers and is equivalent to assuming that σ_c/C_{nom} is well approximated by σ_c/Cu when the number of capacitors is large.

factor is provided in (3.28) – where N is the differential resolution and is related to single-ended resolution[18], Ns, by $N=Ns+1$.

$$f_{\Gamma_i}(\gamma) \cong \frac{1}{\sqrt{2\pi{\sigma_i}^2}} exp\left[\frac{-\gamma^2}{2{\sigma_i}^2}\right]$$
$$\sigma_i{}^2 \cong \left(2^{-i} - 2^{-N}\right)\left(\frac{\sigma_c}{C_{nom}}\right)^2 \quad i \in \{1, \cdots, N-1\} \tag{3.28}$$

Equation (3.28) provides an analytic expression for the PDF of the composite mismatch parameter γ_i of an N bit differential DAC – where γ_0 follows the same distribution as γ_1.

3.3.B Statistical ENOB Expression

Since the ENOB expression in (3.21) depends on a linear combination of $\gamma_i{}^2$, we now derive the PDF for the square of the composite mismatch factor from the PDF of the composite mismatch factor. Letting $\eta_i = \beta_i\gamma_i{}^2$ – where β_i represents the scalar coefficients from the ENOB expression given by (3.21), the PDF of η_i follows a Chi-Squared distribution [38]. Using the PDF described from (3.28) and replacing the scalars β_i with the appropriate values from (3.21), we calculate the distribution for η_i, which is shown in (3.29)[19] – where the distributions for η_0 is described by the distribution for η_1.

$$f_{\mathrm{H}_i}(\eta) \cong \frac{1}{\sqrt{2\pi{\sigma_i}^2}} \eta^{-1/2} exp\left[\frac{-\eta}{2{\sigma_i}^2}\right]$$
$$\sigma_i{}^2 \cong \frac{9(4-\pi)}{4\pi}\left(2^i - 2^{2i-N}\right)\left(\frac{\sigma_c}{C_{nom}}\right)^2 \quad i \in \{1, \cdots, N-1\} \tag{3.29}$$

18 This DAC structure represents the generic sign/magnitude encoded structure described in II.C which utilizes a fixed common-mode output.
19 This PDF is an approximation for the marginal PDF for η

Substituting η_i into the ENOB expression (3.21), we express the ENOB in terms of η_i as shown in (3.30) – where the distributions for η_i are described in (3.29).

$$ENOB = N - \log_4\left(1 + \sum_{i=0}^{N-1} \eta_i\right) \tag{3.30}$$

Equation (3.30) provides an analytic model describing the statistics for the ENOB of an N bit, binary weighted, differential SAR ADC with a normally distributed capacitor mismatch. Furthermore, this model includes a sinusoidal correction factor, so this statistical model is valid for sinusoidally distributed signals and is thus compatible with standard ADC test methods.

3.3.C Expected Value and Variance

We verify the validity of (3.30) by comparing analytical expressions for the expected value and variance of ENOB to numerical simulations of randomly generated SAR ADCs. Because the ENOB expression in (3.30) contains a logarithmic term, we will estimate the expected value and variance using a Taylor series expansion.

Letting X represent the sum of η_i in (3.30), the Taylor series expansion for the ENOB centered at E[X] is shown in (3.31).

$$ENOB = N - \log_4(1 + \mathrm{E}[X]) - \sum_{k=1}^{\infty} \frac{(-1)^k}{k! \ln 4}\left(\frac{X - E[X]}{1 + E[X]}\right)^k \tag{3.31}$$

Taking the expected value of (3.31) and dropping higher order terms, we obtain the approximation for the expected ENOB shown in (3.32).

$$E[ENOB] = N - \log_4(1 + \mathrm{E}[X]) + \frac{Var[X]}{2 \ln 4 \, (1 + E[X])^2} \tag{3.32}$$

Due to the complexity of including correlations between each η_i in later analysis, we will neglect all correlations.[20] Therefore, treating the η_i from (3.30) as independent variables, we can approximate the expected value and variance of X as the sum of the expected values and variances of η_i. Figure 3-2 and Figure 3-3 offer a comparison between the calculated and simulated values for the expected ENOB of a SAR ADC. As shown in Figure 3-2, the calculated ENOB values track the simulated values reasonably well, and in Figure 3-3, we see that the analytic expected value is within 1.0% of the simulated value over a wide range of resolution and mismatch.[21]

Next, we obtain an expression for the ENOB variance. Taking the variance of (3.31) and dropping higher order terms, we derive (3.33).

$$Var[ENOB] = \frac{Var[X]}{[\ln 4 \, (1 + E[X])]^2} \tag{3.33}$$

[20] A comparison between the first four moments of the ENOB expression given in (3.30) and the moments calculated from simulation data showed reasonable similarity, which included correlations, and the moments derived from (3.30) with η_i treated as independent random variables.

[21] In Figure 3-3, however, the error in the expected ENOB is non-monotonic with respect to resolution, we attribute this to the fixed 1024 point FFTs used to generate the simulation data. With a fixed 1024 FFT, only a subset of the output codes is measured for resolutions beyond 11 bits.

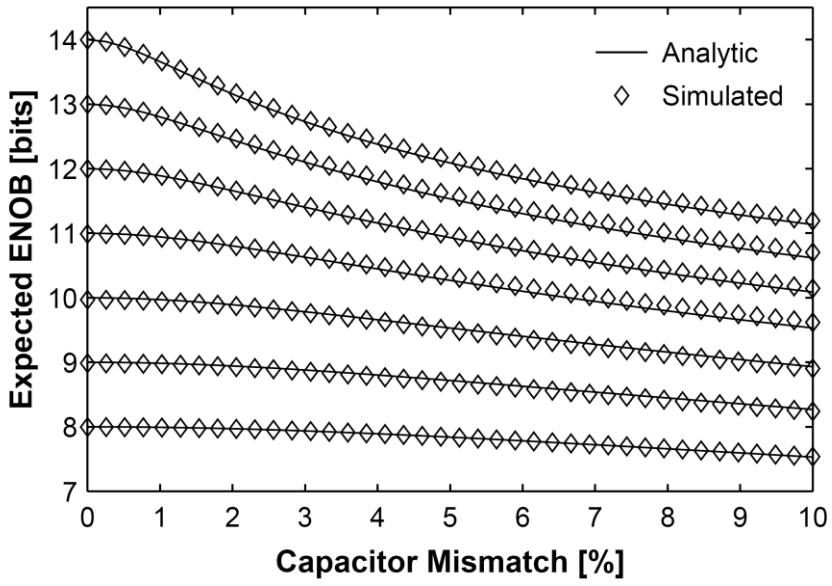

Figure 3-2: Comparison between simulated and calculated expected values (3.32) for ENOB across various resolutions. The numerical simulation results are obtained using a 1024 point FFT of 300,000 randomly mismatched ADCs at each resolution and each standard deviation of capacitor mismatch.

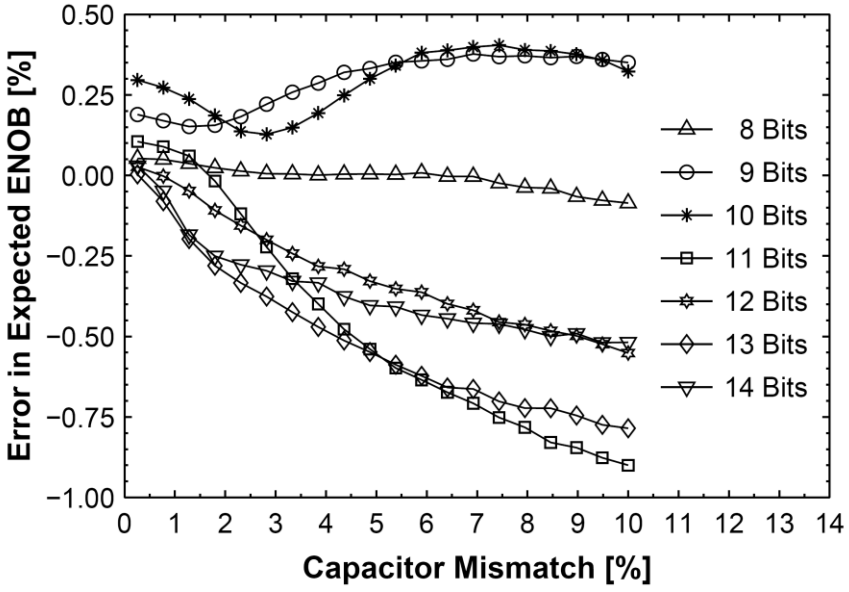

Figure 3-3: Comparison between simulated and calculated expected values (3.32) for ENOB across various resolutions expressed as percent error. The analytic expected ENOB values are within ±1.0% of simulated values.

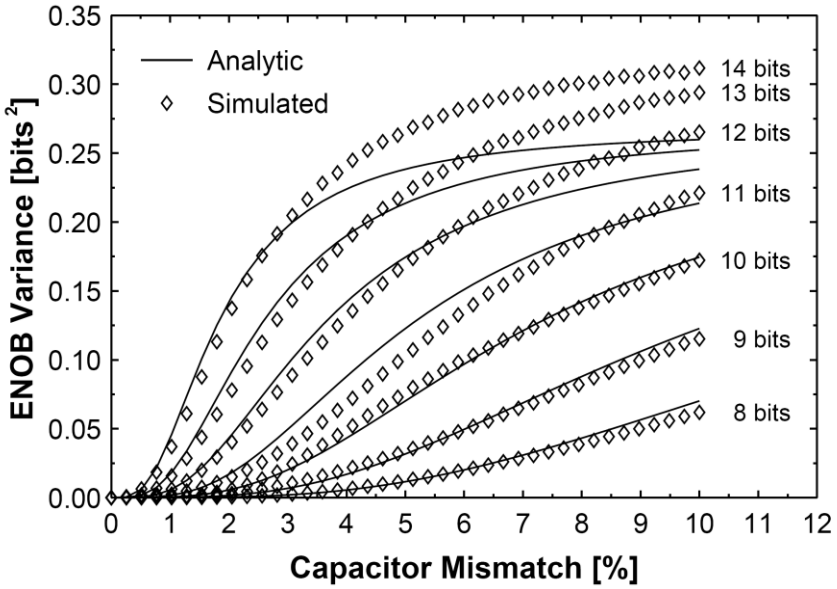

Figure 3-4: Comparison between the simulated and calculated ENOB variances (3.33) across various resolutions. The numerical simulation results are obtained using a 1024 point FFT of 300,000 randomly mismatched ADCs at each resolution and each standard deviation of capacitor mismatch.

Similar to the expected value calculation, we treat the η_i from (3.30) as independent variables and approximate the variance of X as the sum of the η_i variances. Figure 3-4 compares the calculated and simulated values for the ENOB variance.

As shown in Figure 3-4, the calculated variances compress at higher resolutions. This compression indicates a nonlinear relationship between the calculated and simulated variances. Since the inclusion of higher order terms up to the fourth moment of X in the Taylor series expansion did not reduce this error, we attribute the causes of this discrepancy to the scalar correction factor, α, and the assumption that the η_i are independent. While the correction factor α correctly scales the expected ENOB to approximate a sinusoidal distribution, α does not properly scale the higher moments. Furthermore, the η_i are not independent since the γ_i mismatch

parameters are correlated, which is evidenced by (3.8). Nevertheless, the magnitude of the error between the calculated and simulated variances is small compared to the resolution of the ADC.

3.4 Yield Analysis

We complete the statistical analysis of ENOB with an examination of the ENOB yield for an *N* bit, binary weighted, differential SAR ADC. Using the ENOB expression given in (3.30), we can express the probability of achieving some minimal ENOB in terms of the probabilities for η_i as shown in (3.34) – where $ENOB_{MIN}$ is the minimal desired ENOB, and *N* is the ADC resolution in bits.

$$P(ENOB > ENOB_{MIN}) = P(X < 4^{N-ENOB_{MIN}} - 1)$$
$$X = \sum_{i=0}^{N-1} \eta_i \tag{3.34}$$

3.4.A Full Yield Approximation

We next derive an approximate ENOB yield expression in terms of the cumulative distribution function (CDF) for *X* from (3.34). The details of this derivation are provided in APPENDIX B.

When N is even number of bits, the CDF of *X* can be approximated as in (35) – where $F_X(x)$ denotes the CDF of *X*, σ_{2i-2} is the *i*-th even σ from (29) including σ_0, σ_{2i-1} is *i*-th odd σ from (29), and σ_{N-1} denotes the value in the sequence.

$$F_X(x) = \int_0^x \int_0^{\frac{\pi}{2}} \cdots \int_0^{\frac{\pi}{2}} \sum_{i=1}^{N/2} B_i e^{-\lambda_i t}\, d\theta_1 \cdots d\theta_{N/2}\, dt$$

$$B_i = \left[(\lambda_i - s) \prod_{j=1}^{N/2} \left(\frac{A_j}{\lambda_j - s} \right) \right]_{s \to \lambda_i} \tag{3.35}$$

$$A_i = \frac{1}{\pi \cdot \sigma_{2i-2} \cdot \sigma_{2i-1}} \qquad \lambda_i = \frac{\cos^2 \theta_i}{2 \cdot {\sigma_{2i-2}}^2} + \frac{\sin^2 \theta_i}{2 \cdot {\sigma_{2i-1}}^2}$$

. When N is odd number of bits, the CDF of X can be approximated as in (3.36) – where, again, where $F_X(x)$ denotes the CDF of *X*, σ_{2i-2} is the *i*-th even σ from (3.29) including σ_0, σ_{2i-1} is *i*-th odd σ from (3.29), and σ_{N-1} denotes the last value from (3.29).

$$F_X(x) = \int_0^x \int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \cdots \int_0^{\frac{\pi}{2}} \sum_{i=1}^{N/2} C_i e^{-\omega_i t}\, d\theta_1 \cdots d\theta_{N/2}\, d\phi\, dt$$

$$B_i = \left[(\lambda_i - s) \prod_{j=1}^{N/2} \left(\frac{A_j}{\lambda_j - s} \right) \right]_{s \to \lambda_i}$$

$$A_i = \frac{1}{\pi \cdot \sigma_{2i-2} \cdot \sigma_{2i-1}} \qquad \lambda_i = \frac{\cos^2 \theta_i}{2 \cdot {\sigma_{2i-2}}^2} + \frac{\sin^2 \theta_i}{2 \cdot {\sigma_{2i-1}}^2} \tag{3.36}$$

$$C_i = B_i \sin\phi \sqrt{\frac{2t}{\pi \cdot {\sigma_{N-1}}^2}} \qquad \omega_i = \lambda_i \sin\phi + \frac{\cos^2 \phi}{2 \cdot {\sigma_{N-1}}^2}$$

Using the ENOB relationship given by (3.34) along with the CDFs provided by (3.35) and (3.36), we can now calculate the ENOB yield for an *N* bit, binary weighted, differential SAR ADC, but due to the complexity of these equations, however, we provide a more convenient approximation in Section 3.5.

3.4.B Comparison with Simulation Results

We now compare the ENOB yields predicted by the analytic expression for the CDF of X provided in (3.35) and (3.36) to simulated ENOB values. The simulated ENOB yield values are obtained using a 1024 point FFT with a sample of 300,000 randomly mismatched SAR ADCs generated at each resolution and each standard deviation of mismatch. Furthermore, the yield values are extracted from histograms of simulated ENOB values over uniformly distributed bins.

In Figures 3-5 and 3-6, we compare the analytic and simulated ENOB yield curves for capacitor mismatch standard deviations of 1% and 10%. At 1% mismatch, Figure 3-5, we see excellent agreement between the analytic and simulated yield curves, but at 10% mismatch, Figure 3-6, we notice some difference between the analytical and simulated yield curves. Although the 8-10 bits yield curves from Figure 3-6 match well, the 11-14 bit curves display a larger divergence at lower yield values.

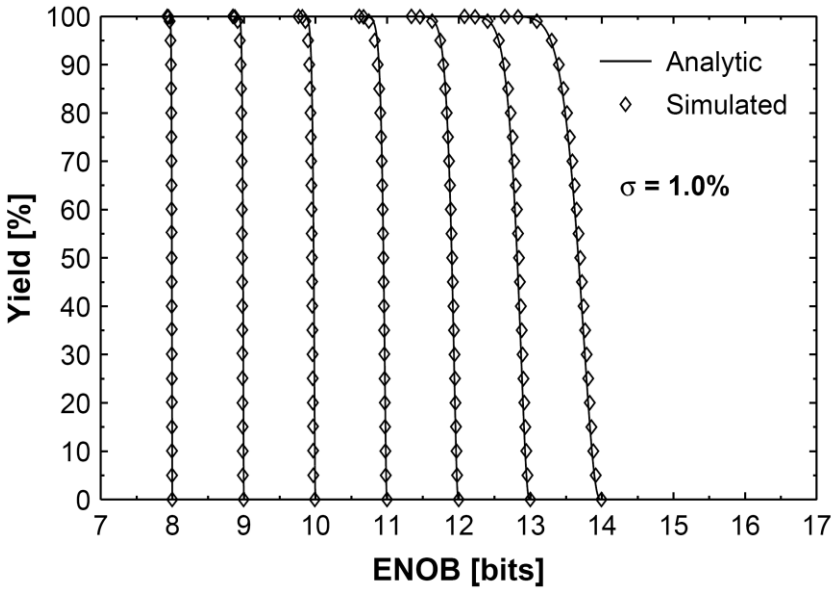

Figure 3-5: Comparison between the simulated and analytic ENOB yields with a standard deviation of 1.0% capacitor mismatch

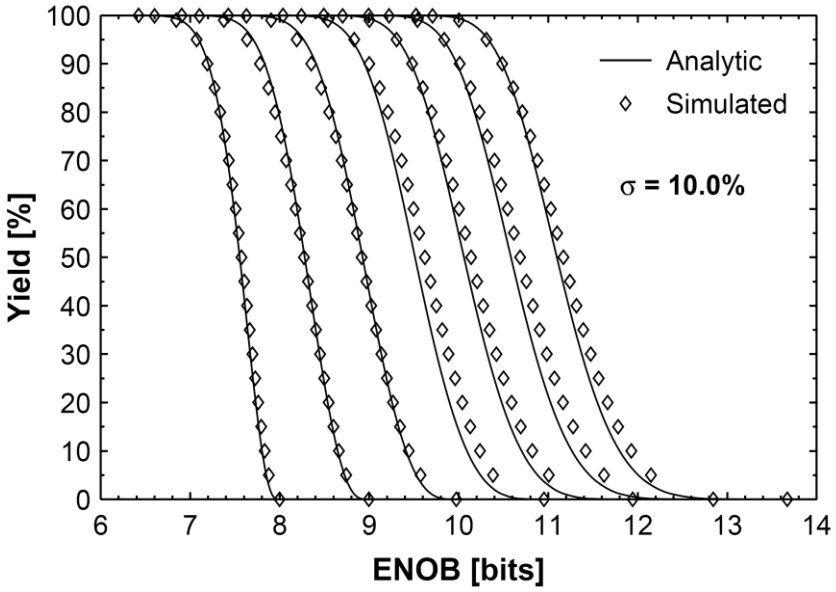

Figure 3-6: Comparison between the simulated and analytic ENOB yields with a standard deviation of 10% capacitor mismatch.

Figures 3-7 and 3-8 offer a more detailed comparison between the analytic and simulated yield curves provided in Figures 3-5 and 3-6. As shown in Figure 3-7, the error between the simulated and analytic ENOB values at 1% mismatch is within ±0.08 bits for 8-14 bits of resolution across the range of yields between 0.5%-99.5%. In Figure 3-8, the error in the ENOB at 10% mismatch is within ±0.17 bits across the range of yields between 0.5%-99.5%. Therefore, at a particular yield value, we see an error in the predicted ENOB less than ±0.08 bits at 1% mismatch and less than ±0.17 bits at 10% mismatch.

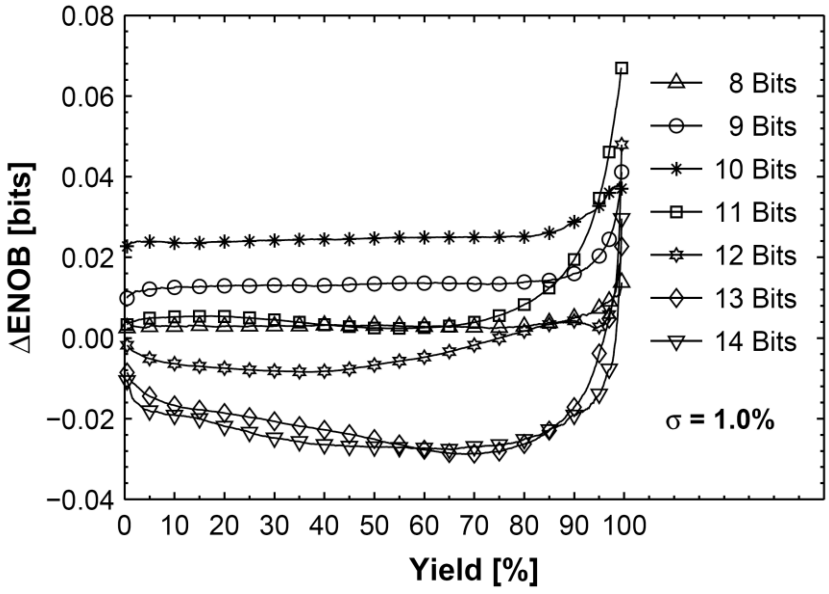

Figure 3-7: Error between the simulated and analytical ENOB values as a function of the yield for a 1% standard deviation of capacitor mismatch. The absolute error in the ENOB is within ±0.08 bits over the range of yields from 0.5% to 99.5%.

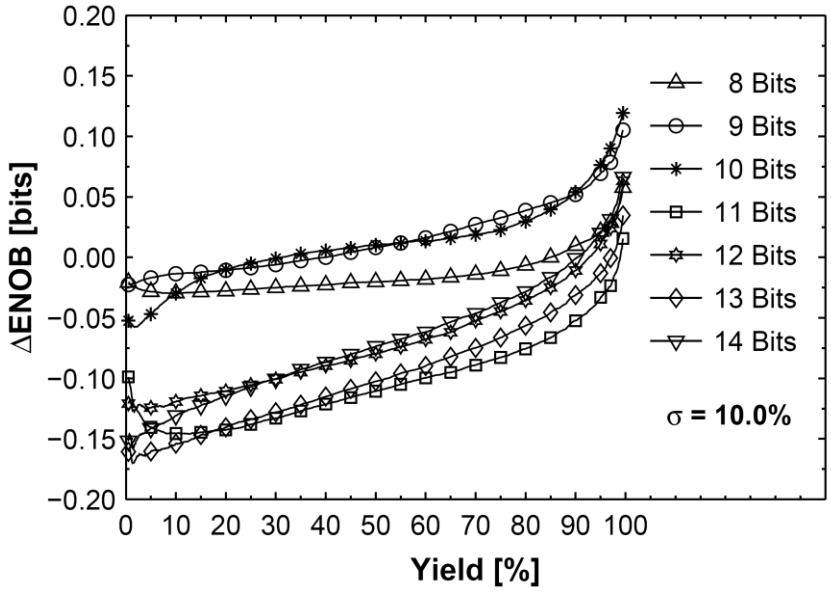

Figure 3-8: Error between the simulated and analytical ENOB values as a function of the yield for a 10% standard deviation of capacitor mismatch. The absolute error in the ENOB is within ±0.17 bits over the range of yields from 0.5% to 99.5%.

In Figures 3-7 and 3-8, we see that the largest errors in the ENOB occur as the yield approaches 0% and 100%. We attribute the source of this error to correlations between η_i values. Since the γ_i mismatch parameters are correlated, which is shown in (3.8), the η_i values are correlated as well. By neglecting these correlations in our statistical model, the frequency of outliers at the tails of the CDF curves are underestimated. Nevertheless, the error in the predicted ENOB is relatively small and the expressions for the CDF of X given in (3.35) and (3.36) offer a reasonably accurate estimate for the ENOB yield of SAR ADCs and, in general, all binary weighted ratiometric converters.

3.5 Simplified Yield Expression

The ENOB yield model provided by (3.34)-(3.36) from Section 3.4 expresses the yield as a function of sigma mismatch and bit resolution. These equations, however, are computationally expensive and cannot be inverted to calculate mismatch as a function of the yield. In this section, we therefore offer an accurate, yet simple, approximation for these expressions which are invertible and allow both the bit resolution and capacitor mismatch to be represented as functions of the ENOB yield.

In what follows, we first develop a single design equation that relates the yield, capacitor mismatch, ENOB, and bit resolution in a more convenient form than (3.35) and (3.36). We then present a sample calculation showing how to extract desired parameters from this new yield expression. Finally, we conclude this section by comparing this compact yield expression derived in this section to both simulation results and the full expressions derived in Section 3.4.

3.5.A Formulation of Simplified Yield Expression

In (3.34), we express the probability of maintaining some minimal ENOB as a function *X*, where *X* is defined as a sum of Chi-Squared random variables with marginal PDFs described by (3.29). Since the sum of independent and identically distributed (iid) Chi-squared random variables follows a Gamma distribution, we standardize *X* in terms of mismatch and bit resolution and approximate its standardized CDF with a normalized incomplete Gamma function, which is the analytic form of the CDF for sums of iid Chi-squared variables. The standardization of *X* is provided in (3.37) and the form of our approximation for the standardized CDF is given in (3.38) – where *Z* represents our standardized variable, and $F_Z(z)$ is the CDF of *Z* expressed as an incomplete Gamma function. Furthermore, we denote Γ as the Gamma function and let *k* and *b* represent the shape and scale parameters of $F_Z(z)$.

$$Z = X \cdot 2^{-N} \left(\frac{C_{nom}}{\sigma_C} \right)^2 \tag{3.37}$$

$$F_Z(Z) = \frac{1}{\Gamma(k)} \int_0^{b\sqrt{z}} t^{k-1} e^{-t} dt \tag{3.38}$$

Using numerical optimization, we calculate values for *k* and *b* which minimize the error between the CDF given in (3.38) and standardized forms of the full CDFs given in (3.35) and (3.36) across the entire 8-14 bit resolution range. A complete formulation of our simplified yield approximation is given in (3.39) – where $ENOB_{MIN}$ is the minimum desired ENOB, *N* is the resolution in bits, σ_c/C_{nom} is the standard deviation of the fractional mismatch, $F_Z(z)$ is the CDF

of Z as described by (3.38)[22], Γ is the Gamma function, and both k and b are empirical fitting parameters.

$$
\begin{aligned}
&P(ENOB > \ ENOB_{MIN}) = P(X < 4^{N-ENOB_{MIN}} - 1) \\
&Z = X \cdot 2^{-N} \left(\frac{C_{nom}}{\sigma_C}\right)^2 \\
&F_Z(Z) = \frac{1}{\Gamma(k)} \int_0^{b\sqrt{z}} t^{k-1} e^{-t} dt \\
&k = 7.944 \quad b = 13.146
\end{aligned}
\tag{3.39}
$$

Equation (3.39) relates yield, mismatch, ENOB, and resolution in a single closed form expression. For simplicity, we offer MATLAB® code in Figure 3-9 as an example of how to interpret (3.39) – where we have implemented $F_Z(z)$ using the standard function provided by the software. This code in Figure 3-9 calculates both yield as a function of resolution and mismatch and calculates mismatch as a function of yield and resolution. When this code is executed, the yield calculation returns 95% for *YIELD* and the sigma calculation returns 0.1 for *SIGMA*. We omit a resolution calculation since resolution is easily derived from the sigma calculation by rearranging the terms.

[22] Both the expressions for $F_Z(z)$ and its inverse are standard functions in most commercial math programs

```
% Parameter Values
N = 9;  SIGMA = 0.1;  ENOB_MIN = 7.7;
YIELD = 0.95;  k = 7.944;  b = 13.146;

% Yield Calculation
X=4^(N-ENOB_MIN)-1;
Z=X/2^N/SIGMA^2;
YIELD=gammainc(b*sqrt(Z),k)

% Sigma Calculation
X=4^(N-ENOB_MIN)-1;
Z=(gammaincinv(YIELD,k)/b)^2;
SIMGA=sqrt(X/Z/2^N)
```

Figure 3-9: Example MATLAB® code for implementing the yield equation provided in (3.39). This code calculates the yield as function of resolution and mismatch and calculates mismatch as a function of yield and resolution.

3.5.B Comparison of Yield Expression

We now compare the yield expression from (3.39) to both the full expressions from Section 3.4 and simulation results. In Figure 3-10, we plot the difference between yield values calculated using the approximation given in (3.39) and analytic values calculated using (3.35) and (3.36) as standardized to Z through (3.37). As shown in Figure 3-10, the error in the yield values, expressed as a difference in percentages, is within ±0.16% over the range of resolutions between 8-14 bits. This shows that the simplified expression provided by (3.39) is a good approximation of the full expressions from Section 3.4.

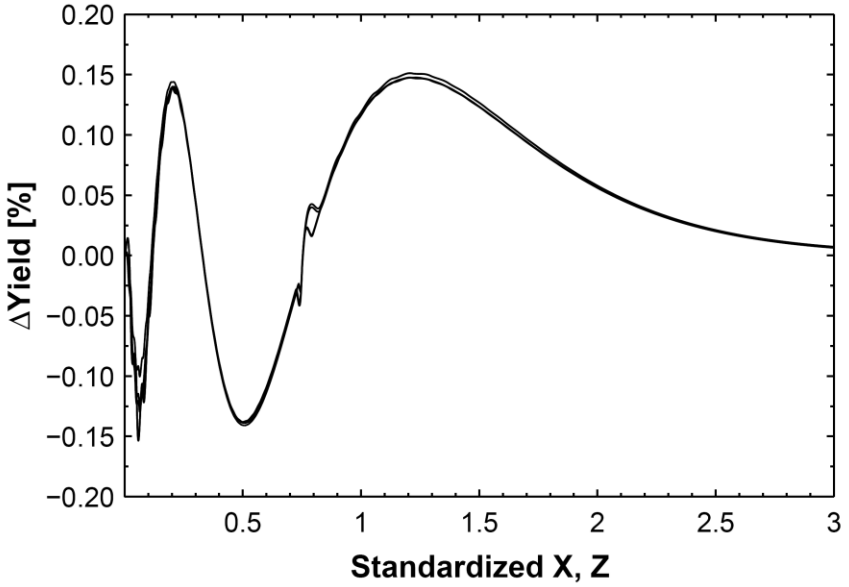

Figure 3-10: Error in yield values between (3.39) and the full expression given by (3.35) and (3.36) as standardized to Z through (3.37) at each resolution from 8-14 bits. The error is expressed as difference in percentages. Since the differences in yields associated with each of the 8-14 bit curves resemble one another so closely, we do not distinguish between the 7 individual curves. As shown, the absolute error between the yield values is within ±0.16% which indicates that the Gamma Distribution approximation from (3.39) matches the full expressions very well.

In Figures 3-11 and 3-12, we compare ENOB values calculated using (3.39) to simulated values at a constant yield of 95%. The simulated yield values are obtained using a 1024 point FFT with a sample of 300,000 randomly mismatched SAR ADCs generated at each resolution and each standard deviation of mismatch. As shown in Figure 3-11, the analytic ENOB values obtained from (3.39) agree with the simulated values, and as shown in Figure 3-12, these analytic ENOB values match within ±0.12 bits at a constant 95% yield.

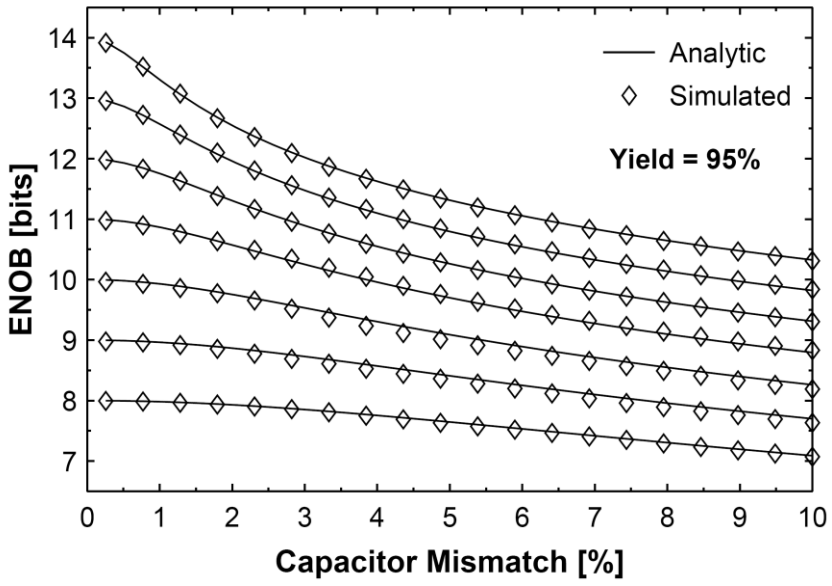

Figure 3-11: Comparison between simulated and the analytically calculated ENOBs using the approximation from (3.39) for a constant yield of 95%. The plot shows the minimum value of ENOB allowed for a good ADC to achieve a yield of 95%. As shown, the simulated values of ENOB match the analytic curves.

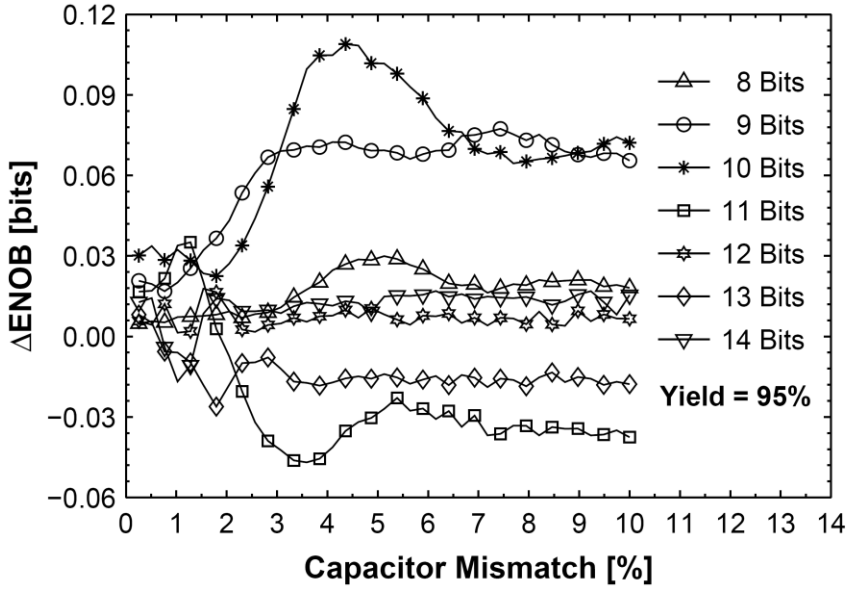

Figure 3-12: Error between simulated and analytically calculated ENOBs using the approximation from (3.39) with a constant yield of 95%. The error between the analytic and simulated ENOBs is less than ±0.12 bit.

Equation (3.39) represents a simple and accurate design equation for calculating yield. In (3.39), the CDF of X is standardized and related to a normalized incomplete Gamma function. Since both the incomplete gamma function and its inverse are standard functions in most numerical software packages, (3.39) provides a convenient design approximation which relates mismatch, ENOB, and yield for binary weighted ratiometric converters.

Although we have neglected losses in ENOB that occur from comparator noise, *kT/C* noise, and sampling jitter, (3.39) can accommodate a more comprehensive yield analysis using these additional noise sources. Assuming these additional noise sources are independent, we can normalize each of their powers by $\Lambda^2/4$ and add them to X in (3.34). Once included in X, we can we can calculate a refined ENOB yield equation by convolving the PDFs of the additional noise power terms with the PDF for X.

3.6 Conclusions

In this chapter, we develop a yield model for binary weighted SAR ADCs based on ENOB which is applicable to all binary weighted ratiometric converters, and we present the results as an accurate and easily implementable design equation. In addition, we derive an exact analytical expression relating mismatch and resolution to ENOB for uniformly distributed signals, and also offer an accurate expression relating mismatch and resolution to ENOB for sinusoidal signals. This work presents the first mathematical expression relating resolution, mismatch, ENOB, and yield. From this work, the mismatch required to achieve a certain ENOB with a particular yield can be calculated, and the fundamental limit on accuracy for binary weighted ratiometric converters can be estimated in terms of component matching.

CHAPTER IV

Noise-Shaping SAR[23]

4.1 Introduction

In recent years, charge-redistribution Successive Approximation (SAR) ADCs have exhibited the highest conversion efficiencies for ADCs with moderate resolution and bandwidth [39]-[41]. For effective resolutions beyond 10 bits or so, however, the accuracy of the SAR circuit blocks limits the overall energy efficiency of the converter. At high resolutions, for instance, the DAC voltages become small compared to the input-referred noise of a dynamic comparator, necessitating an additional power-hungry, low-noise pre-amplifier to drive the comparator. To improve the effective resolution of SAR ADCs, this work introduces a technique that decouples the accuracy of the comparator from the resolution of the ADC.

In this paper, we introduce a low Oversampling-ratio (OSR) noise-shaping SAR ADC that leverages noise shaping to increase the resolution of a conventional SAR ADC. The prototype converter uses an 8-bit capacitor DAC and achieves an ENOB of 10.0 bits over a signal bandwidth of 11MHz with an extremely low OSR of 4. Through noise-shaping and oversampling, we mitigate some of the losses from mismatch, *kT/C* noise, and comparator noise by trading bandwidth for accuracy, which allows us to achieve higher resolutions using lower

[23] The material in this chapter on Noise-Shaping SARs was first presented in [59] and [60].

resolution and lower accuracy circuit blocks. Significantly, the input referred noise of the comparator is noise-shaped along with the quantization noise so the comparator no longer requires the full accuracy of the converter. The noise-shaping technique presented in this paper provides a means to enhance the resolution of SAR ADCs without a significant modification to the basic SAR ADC structure.

While SAR ADCs are very efficient at moderate resolutions, fundamental and related second-order effects significantly reduce the efficiency of SAR ADCs at higher resolutions. As with all ADCs, *kT/C* noise limits sampling accuracy. For moderate resolution ADCs, the minimum capacitance to achieve adequate sampling noise is greater that the required capacitance needed to achieve adequate matching. In addition, a large DAC array capacitance leads to second-order effects that also limit performance. These include the signal dependent resistance of the input switch and slow settling due to parasitic capacitances. Although techniques such as switch gate boosting [42] and redundancy [43]-[44] can alleviate these second order effects, these techniques invariably lead to higher power consumption. A significant advantage of oversampling is that it attenuates *kT/C* noise, but without noise shaping, oversampling is usually unattractive and until this work, noise shaping has not been efficiently demonstrated in SAR ADCs.[24] Although noise-shaping ADCs have previously employed SAR ADC structures as a multi-bit quantizer in delta-sigma ADCs [45] this work embeds noise-shaping into the SAR ADC topology while maintaining the power efficient operation of the SAR ADC.

The input referred noise of the comparator in a SAR ADC is a fundamental limitation to performance, which we alleviate by noise shaping. In a straight, binary SAR ADC, all trial

[24] A similar noise-shaping system for SAR ADCs is described by [40], but the work involves only simulated data, and a practical implementation of [40] requires a highly linear, power-hungry opamp to drive the entire DAC array of the SAR ADC.

comparisons must be made at the full accuracy of the overall converter. This requirement determines the maximum input noise of the comparator and in turn the power consumption of the comparator [46]. Moreover, a preamp is often required at higher resolutions due to noise constraints. The large input devices, needed for low noise comparator operation, increase the parasitic capacitance on the critical top-plate SAR residue nodes. Redundancy schemes can reduce the accuracy needed for the earlier trials, but as noted earlier, redundancy substantially increases the complexity of the ADC. In any case, the later bit trials must still be made to the full ADC accuracy. In this work, a noise-shaping scheme shapes both quantization noise and comparator noise so that comparator noise is decoupled from the ADC resolution.

Noise shaping reduces the number of capacitors in the DAC array, simplifying the practical implementation of the DAC array. It is clear that the number of capacitors in a binary weighted capacitor DAC array grows exponentially with resolution. While by itself, this is not a fundamental limit to performance, it does present practical impediments to performance. Routing is necessarily complicated in high resolution SAR ADCs. Furthermore, the very large ratio between the smallest and largest capacitances can become problematic since the finite minimum value of capacitance can lead to a large total capacitance at high resolutions. The use of a sub-DAC alleviates this problem [47] but the use of a sub-DAC often requires careful calibration of the coupling capacitor [48]. Noise shaping reduces the DAC resolution, and therefore, substantially reduces the number of capacitors in the DAC array.

4.2 SAR ADC Review

In this section, we review the basic operation of the SAR ADC architecture used in this work, Figure 4-1. During the first phase of operation, an input voltage is bottom-plate sampled

onto a binary weighted capacitor array through a bootstrapped switch. Following this sampling operation, each of the capacitor array bottom plates is initialized to a common mode voltage, and then the ADC performs a binary search under control of the SAR logic.

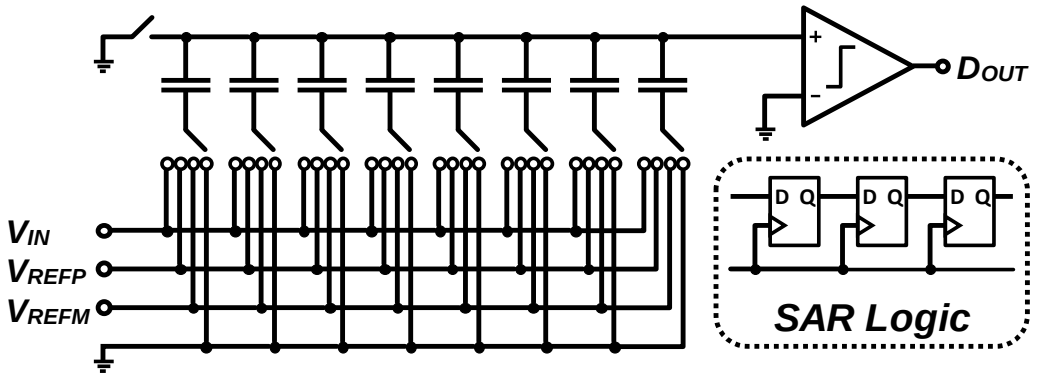

Figure 4-1: Basic operation of the SAR ADC

In this design, the SAR algorithm performs sign-magnitude encoding of the sampled input voltage, and the capacitor DAC uses bipolar reference voltages during the binary search. Therefore, after the DAC references are initialized to the common mode reference voltage, the comparator tests the sign of the sampled voltage and this sign decision is fed back to the bottom plate switches of the MSB capacitor in the DAC. When the subtraction of voltage is required, only the MSB switches move from the common mode reference voltage to a lower reference voltage, and when addition of voltage is required, only the MSB switches move from the common mode reference voltage to a higher reference voltage. The switches for the rest of array are left at the common mode reference voltage after this first decision. Throughout the rest of the conversion algorithm, the comparator is enabled and the decision is fed to the appropriate binary weighted caps in the array.

4.3 Noise Shaping in a SAR ADC

4.3.A Residue generation

We first introduce an efficient and almost seamless technique for measurement of the quantization error. Efficient capture of the quantization error is vital for efficient noise shaping. The quantization error, *Q*, is simply defined by equation (4.1).

$$D_{OUT} = V_{IN} + Q \tag{4.1}$$

In a conventional SAR ADC, the final residue information produced by the SAR DAC at the end of the conversion is discarded when a new input voltage is sampled onto the array for the next analog-to-digital conversion. As discussed in Section 4.2, after each bit trial, the DAC references, V_{REFM} and V_{REFP}, are connected to capacitor bottom plates so that the comparator input represents the un-digitized residue. However, since the analog-to-digital conversion is complete when the comparator determines the least significant bit, the last decision is not fed back to the DAC array. In other words, when the SAR ADC conversion is complete for an *N*-bit ADC, the magnitude of the residue voltage produced at the top plate of the DAC represents the difference between the sampled input and a digital estimate constructed from the first N-1^{th} decision. This is shown in the example of an 8-bit ADC in Figure 4-2.

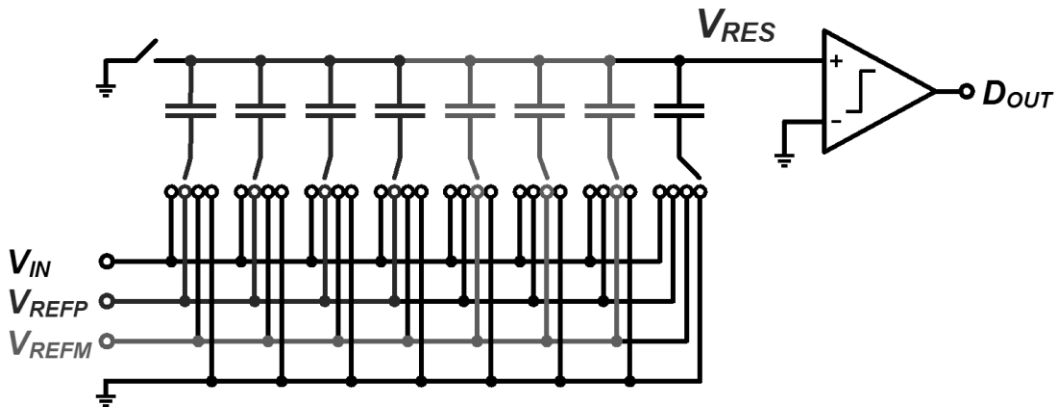

Figure 4-2: The residue voltage produced on the DAC after conversion by an 8-bit SAR ADC is the difference between the sampled input and a 7-bit digital estimate.

The residue voltage produced on the DAC after completing the digital conversion by the 8-bit SAR ADC is the difference between the sampled input and a 7-bit digital estimate. The final residue is therefore not based on the full resolution of the digital estimate. This inequality is expressed by (4.2) – where D_{OUT} is the 7-bit estimate of an 8-bit conversion.

$$V_{RES} \neq D_{OUT} - V_{IN} \tag{4.2}$$

In Figure 4-3, we make one extra switching of the DAC array based on the final comparator decision so that the final residue is:

$$V_{RES} = D_{OUT} - V_{IN} \tag{4.3}$$

Significantly, this final residue voltage also contains information about the input-referred comparator noise, $V_{N,COMP}$. As indicated in Figure 4-4, this final residue also captures the comparator noise for the N^{th} comparison. As in (4.4), V_{RES} can be expressed as a function of D_{OUT}, V_{IN}, and $V_{N,COMP}$.

$$V_{RES} = D_{OUT} - V_{IN} + V_{N,COMP} \tag{4.4}$$

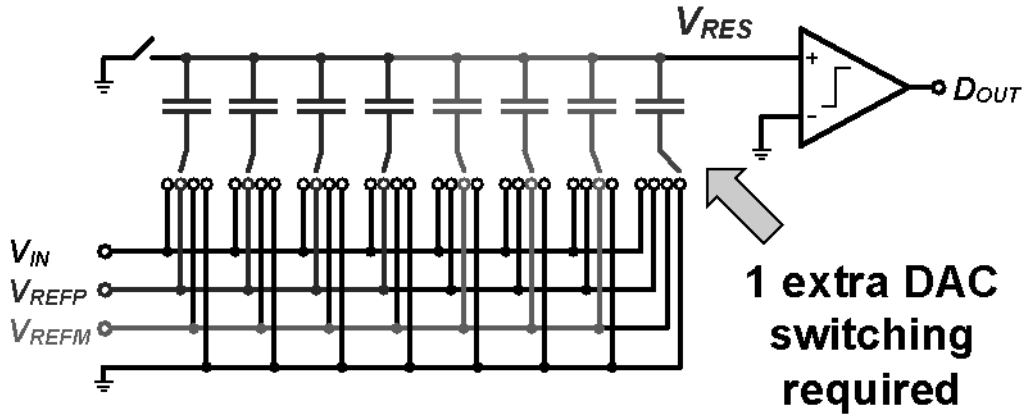

Figure 4-3: One extra switching of the DAC array based to generate resume

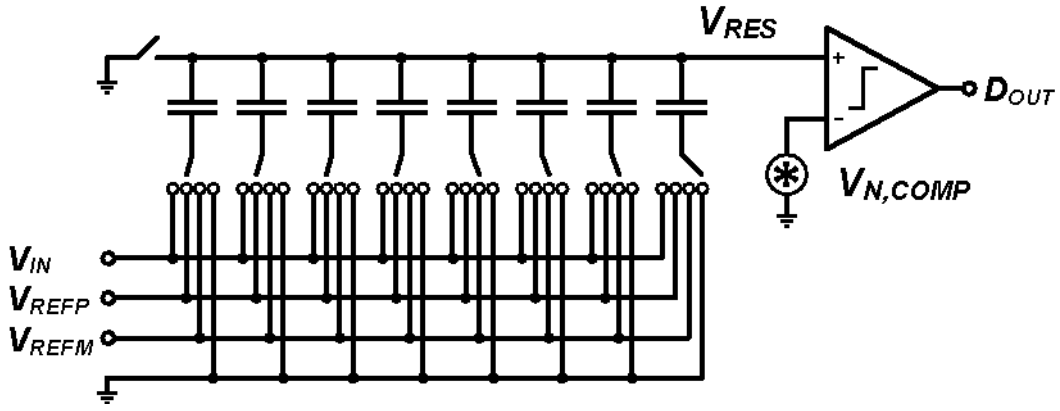

Figure 4-4: Final residue also captures the comparator noise for the N^{th} comparison.

4.3.B Simple Noise Shaping

For clarity of explanation, we begin with the simple SAR ADC noise shaping technique introduced in Figure 4-5.[25]

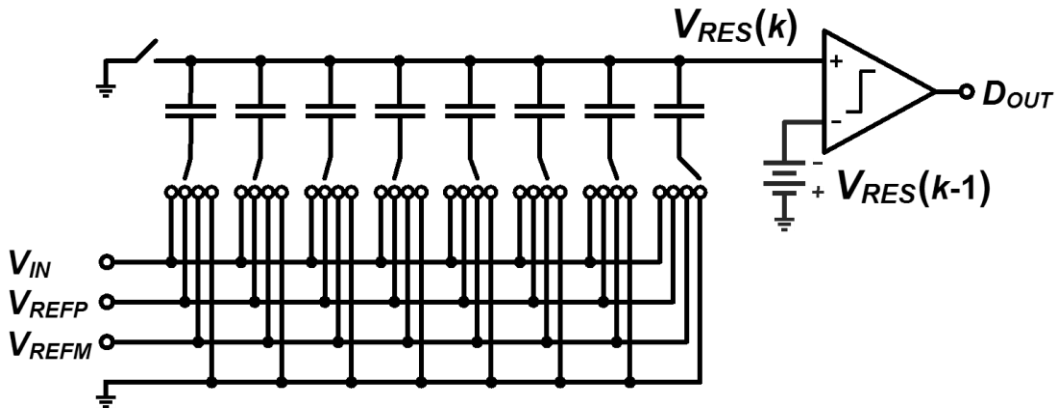

Figure 4-5: Simple SAR ADC noise shaping technique.

In this technique, the residue, $V_{RES}(k\text{-}1)$, from the conversion of the previous ADC sample, *k-1*, is applied to the negative input of the comparator during the conversion of current sample, *k*. Including comparator noise, $V_{N,COMP}(k)$, D_{OUT} is expressed by (4.5).

$$D_{OUT}(k) = V_{IN}(k) + Q(k) + V_{N,COMP}(k) - V_{RES}(k-1) \qquad (4.5)$$

Considering that the residue voltage generated by the DAC represents the difference between D_{OUT} and V_{IN}, V_{RES} can be expressed as follows,

[25] It is important to note that the technique employed in this subsection is for illustration purposes only and does not produce a stable noise-shaping system.

$$V_{RES}(k) = D_{OUT}(k) - V_{IN}(k) \tag{4.6}$$

Substituting (4.6) into (4.5) and performing a *z*-transform, we obtain the following system transfer function,

$$D_{OUT}(z) = V_{IN}(z) + \frac{1}{1+z^{-1}}[Q(z) + V_{N,COMP}(z)] \tag{4.7}$$

This equation indicates an all-pass signal-transfer function (STF) and a high-pass noise-transfer function (NTF), which shapes both the quantization noise and comparator noise, thereby attenuating both the quantization noise and comparator noise at lower frequencies. At high frequencies, however, the system becomes unstable since the NTF contains a pole at Nyquist.

In practice, the battery that applies $V_{RES}(k-1)$ to the negative input of the comparator can be implemented as a capacitor, C_{COMP}. At the end a the SAR conversion, and after the last comparator decision is fed back into the array, the battery capacitor, C_{COMP}, is charged-shared with the DAC top plate voltage. Since C_{COMP} is much smaller than the total DAC capacitance, the residue voltage that is sampled onto C_{COMP} is almost identical to the actual residue voltage. Because C_{COMP} is small compared to the DAC, memory effects can also be ignored.

With a small C_{COMP} capacitance, sampling of the SAR ADC residue on C_{COMP} introduces an additional *kT/C* noise contribution in the ADC operation, which depends on the capacitance of C_{COMP}. This *kT/C* noise contribution, however, presents itself to the comparator in series with the input-referred comparator noise, and thus, this additional *kT/C* noise experiences the same noise transfer function as the quantization noise and the input-referred comparator noise. Therefore, noise shaping and oversampling and the eventual digital filtering of the overall ADC output will greatly reduce the effective contribution of this *kT/C* noise from C_{COMP}.

Figure 4-6 shows a functional representation and the equivalent signal flow diagram of this simple noise shaping SAR ADC**.** Thanks to the additional DAC switching, the DAC array generates the quantization residue. Effectively, this ADC architecture feeds forward the ADC input to the quantizer, where a delayed version of this residue is summed with the input and is then fed to the quantizer.

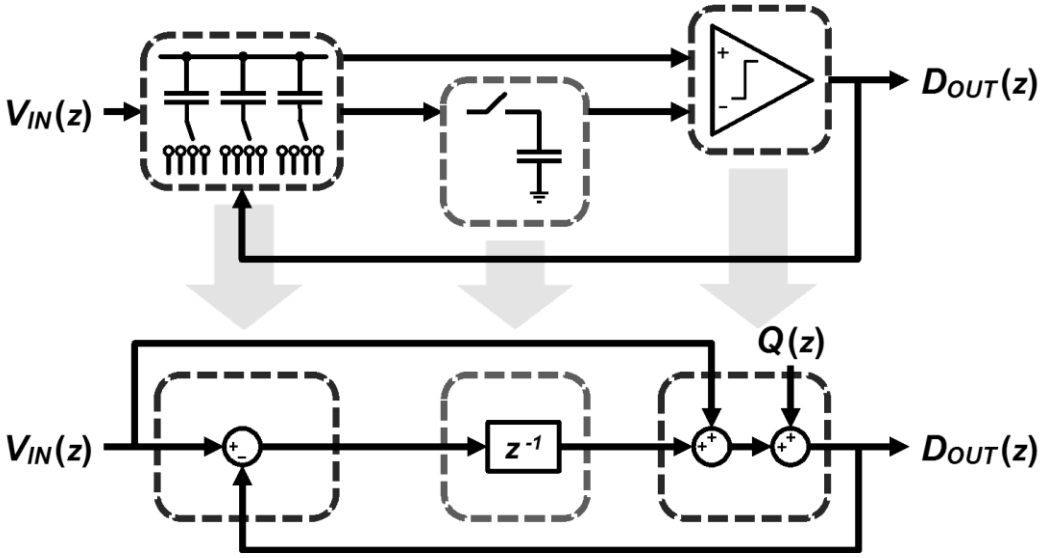

Figure 4-6: Functional representation and the equivalent signal flow diagram of the simple noise shaping SAR ADC.

Through this simplified noise shaping implementation, both the quantization noise and the input-referred comparator noise of the ADC can be reduced at the expense of bandwidth. The noise transfer function that is associated with this simple noise shaping is shown in Figure 4-7. Although the architecture illustrates the advantage of shaping both quantization noise and the comparator noise, the NTF indicates only a flat 6dB of attenuation of low frequency. For this reason, the effective improvement in resolution offered by this technique is small. The resolution

improvement from this simplified implementation does not trade resolution and bandwidth equally, so in terms of figure of merit, the resolution improvement is not an energy efficient design tradeoff. More importantly, however, this noise shaping technique does not produce a stable NTF, and is thus, of little use in practice.

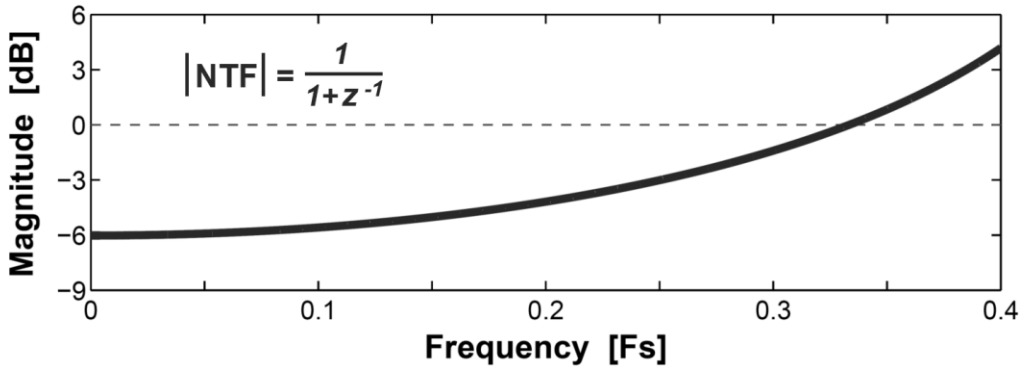

Figure 4-7: Noise transfer function associated with this simple noise shaping.

4.3.C Improved Noise Shaping

The simplified noise shaping system described in Section 4.3.B illustrates an implementation that uses the DAC residue voltage produced at the end of the SAR conversion to perform noise shaping. We improve the resolution gain of the simplified implementation and stabilize the NTF described in the previous section by inserting an integrator between the passive sampling network and the inverting terminal of the comparator. With ideal sampling of the DAC residue voltage and an ideal integrator, this system behaves exactly like a first order delta-sigma modulator.

Figure 4-8 shows a functional representation and the equivalent signal flow diagram of the improved noise-shaping SAR ADC, which now includes a integration filter after the sampling of the final DAC residue. As shown in the signal flow diagram, the sum of the input signal and the integrated residue is fed to the quantizer.

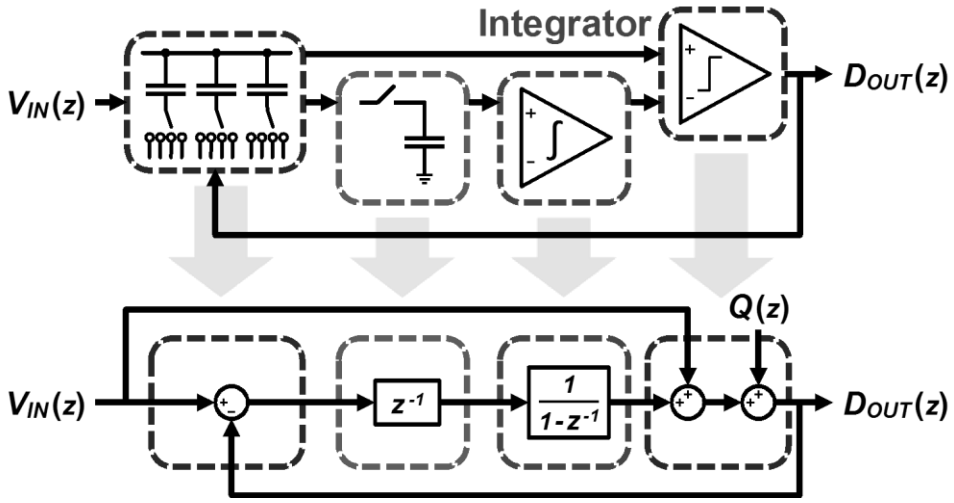

Figure 4-8: Functional representation and the equivalent signal flow diagram of the improved noise-shaping SAR ADC.

The transfer function for this improved noise-shaping system is:

$$D_{OUT}(z) = V_{IN}(z) + (1 - z^{-1})Q(z) \tag{4.8}$$

As before, the STF is all-pass. However, the NTF, $1\text{-}z^{-1}$, is now identical to the NTF of a first order delta-sigma modulator. As expected, a plot of noise transfer function, as shown in Figure 4-9, indicates significant attention of quantization noise at lower frequencies.

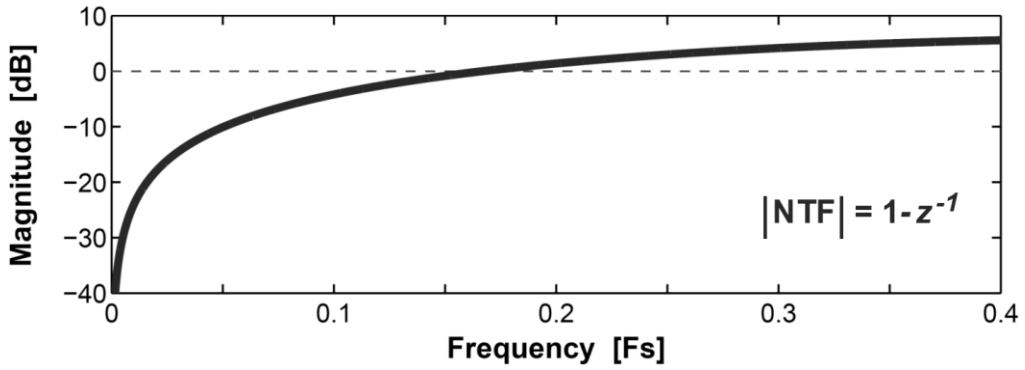

Figure 4-9: Plot of noise transfer function for the improved noise shaping SAR ADC. Frequency is plotted on a linear scale in order to include frequencies near Nyquist, but on a log-log scale, the NTF will show the traditional 20dB/decade slope at low frequencies.

To understand the limitations of this improved architecture, we consider the simplified depiction of this system shown in Figure 4-10. The DAC residue is sampled by residue-sampling capacitor, C_R, and this sampled charge is then transferred to an integrator, which is formed by an OTA with a feedback capacitor, C_F – where C_P represents parasitic capacitances which include contributions from the switches and the OTA input. As with the simple noise-shaping scheme in Section 4.3.B, C_R introduces an additional *kT/C* noise contribution. Unlike the simple noise-shaping scheme from Section 4.3.B, however, this *kT/C* noise contribution from C_R is not noise-shaped, but is still digitally filtered. Practically speaking, the inclusion of the integrator prevents noise-shaping of this *kT/C* noise because the integrator provides gain through the loop and this noise contribution occurs before the integrator and cannot be input-referred though the integration. The residue-sampling capacitor, C_R, must therefore be sized in accordance with the desired resolution of the ADC in order to keep the *kT/C* noise negligible, and at high resolutions, charge sharing between the ADC and a large C_R will set the minimum size of the DAC array to avoid too much attenuation on the residue-sampling capacitor.

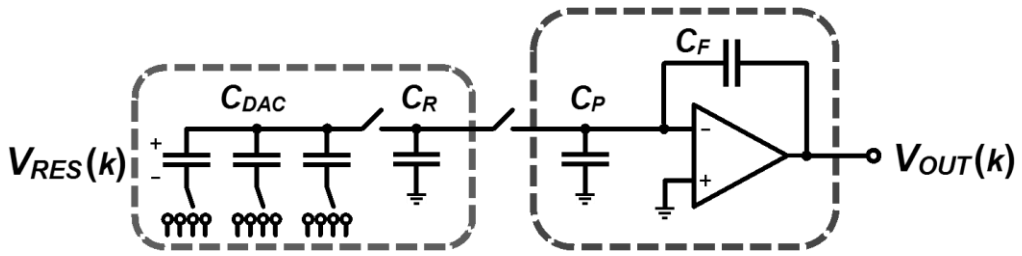

Figure 4-10: Simplified depiction of improved noise-shaping SAR ADC.

We now consider the effect of finite amplifier gain as well as errors related to the parasitic capacitance C_P. For this purpose, we model the residue processing with the signal flow diagram shown in Figure 4-11.

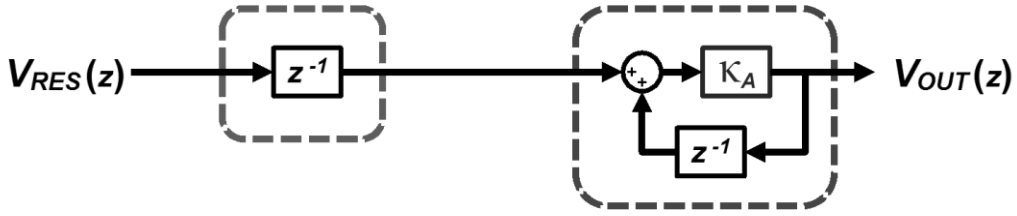

Figure 4-11: Model of the residue processing in the improved noise shaping SAR ADC.

In Figure 4-11, we introduce a quality factor κ_A for the integrator. A κ_A value of 1.0 indicates an ideal integrator, and in practice κ_A is smaller than but very close to unity. This quality factor represents losses due to both finite amplifier gain and the attenuation due to charge sharing between C_R and C_P. Figure 4-12 plots the noise transfer function for three values of κ_A ranging from 0.6 to ideal, 1.0.

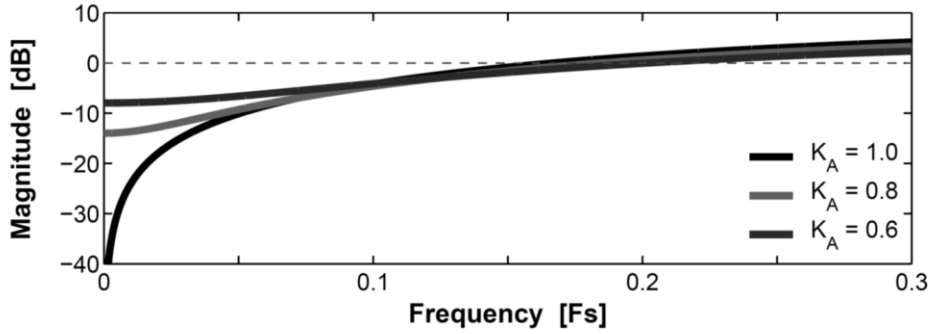

Figure 4-12: Noise transfer function for three values of κ_A.

We see that even with a very poor integrator (i.e. κ_A of 0.6), reasonable noise shaping is still obtained. Nevertheless, a higher value of κ_A is typically desired to achieve an attractive tradeoff between bandwidth and resolution. However, a high value of κ_A requires a high gain amplifier and accurate handling of charge. This extra complexity is undesirable since it is at odds with the scaling friendly nature of the basic switched capacitor SAR architecture.

4.3.D Practical Noise Shaping

We introduce a cascaded Finite-Impulse-Response (FIR) Infinite-Input-Response (IIR) filter as a loop filter to achieve practical noise shaping. To reconcile the design tradeoff between bandwidth and resolution, the additional switched capacitor FIR filter replaces the passive sampling network described in Sections 4.3.B and 4.3.C. Figure 4-13 shows the signal flow diagram for the noise-shaping scheme that incorporates this new FIR-IIR loop filter.

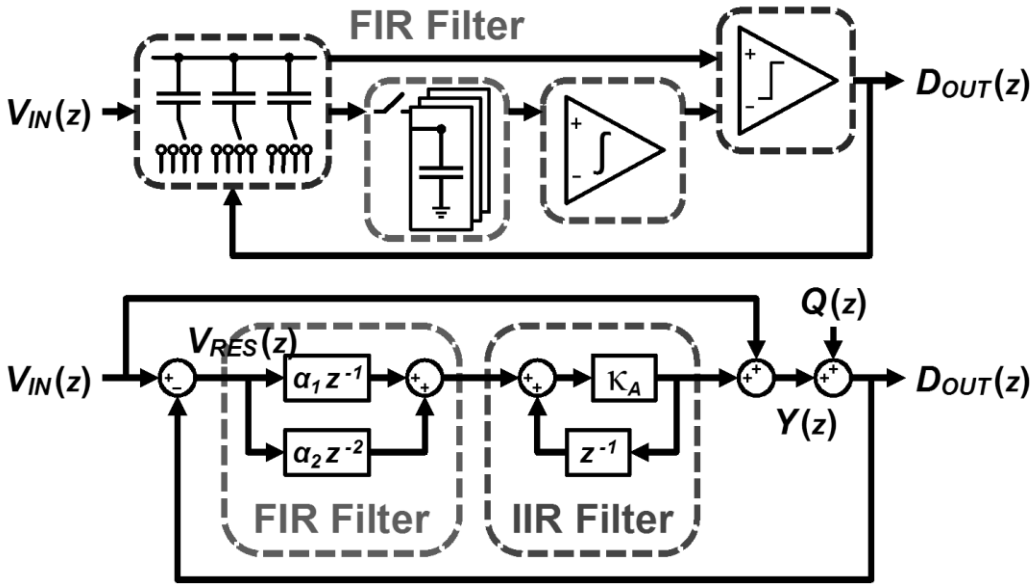

Figure 4-13: Noise shaping with cascaded FIR/IIR filter.

In this system, the residue voltage, $V_{RES}(z)$, is processed by the cascade of the FIR and IIR filters and produces an output $Y(z)$. The filter output, $Y(z)$, is then summed with the input signal feed-forward path and fed to the quantizer. The FIR filter is a two-tap filter with coefficients α_1 and α_2. The IIR filter is formed with the integrator which has a quality factor κ_A. The overall transfer function of this FIR-IIR system is given by (4.9).

$$D_{OUT}(z) = V_{IN}(z) + \frac{1 - \kappa_A z^{-1}}{1 - \kappa_A(\alpha_1 - 1)z^{-1} + \kappa_A \alpha_2 z^{-2}} Q(z) \qquad (4.9)$$

Again the signal transfer function described by (4.9) is all-pass, but now thanks to coefficients α_1 and α_2 there is flexibility in the form of the noise transfer function. In this design, α_1 is set at 3.0 and α_2 is set at 1.0 for a simple integrator with κ_A of 0.64.[26]

The circuit implementation of the cascaded FIR-IIR filter is shown in Figure 4-14. The FIR filter is a two-tap filter constructed as a pair of two-capacitor banks. Alternate DAC residue voltages are alternately sampled onto B_{ANK1} and B_{ANK2} at the end of each ADC conversion cycle.

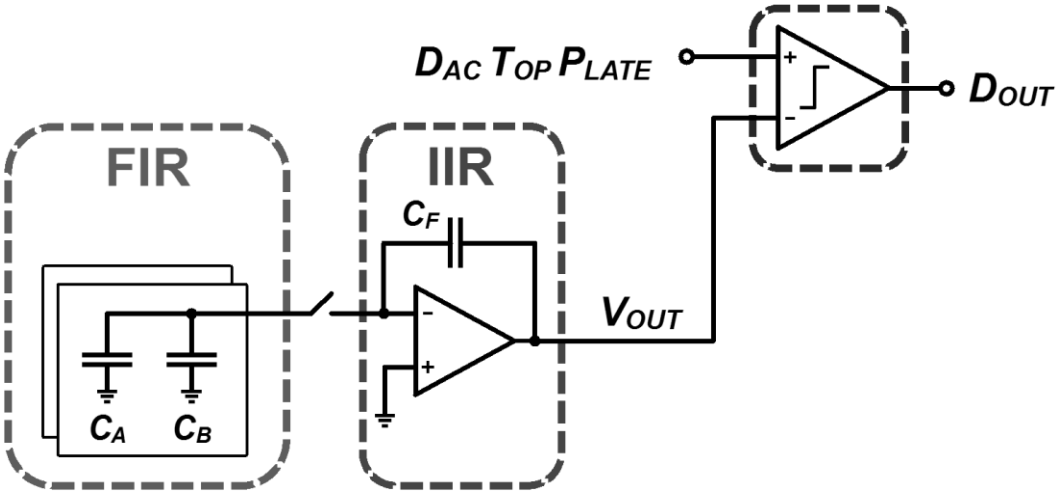

Figure 4-14: Circuit implementation of the cascaded FIR-IIR filter.

As shown in Figure 4-15, B_{ANK1} is formed with capacitor C_{A1} and C_{B1} while B_{ANK2} is formed with C_{A2} and C_{B2}. The FIR tap coefficients α_1 and α_2 are set by the size of capacitors (i.e. capacitor designated *A* and B) within the capacitor banks.

26 The tap coefficients chosen for FIR filter produce an unstable filter, but when combined with losses from the integrator, the noise transfer function is stabilized.

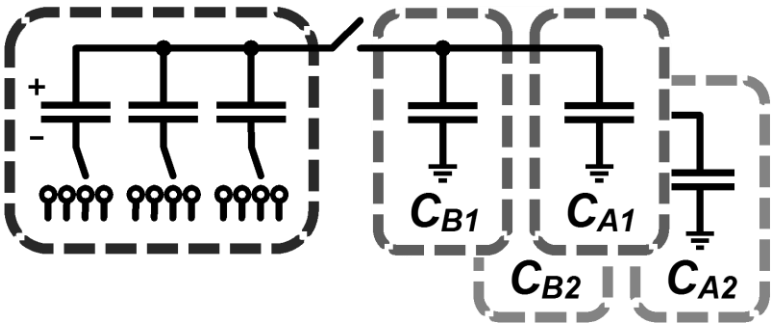

Figure 4-15: Capacitor banks in FIR filter.

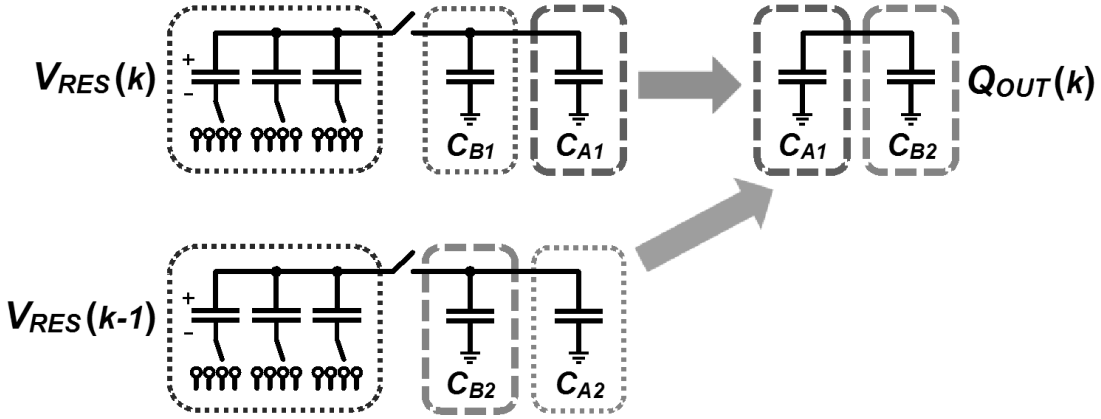

Figure 4-16: Interleaved FIR operation.

As shown in Figure 4-16, the residue voltage, $V_{RES}(k-1)$, is passively sampled onto C_{B2} and C_{A2}. Later, residue, $V_{RES}(k)$, is passively sampled onto C_{B1} and C_{A1}. The charges on C_{B2} and C_{A1} are combined to form the FIR filtered charge $Q_{OUT}(k)$. Next, C_{B2} and C_{A2} are reused to sample the next residue, $V_{RES}(k+1)$, and after this, the charges on C_{B1} and C_{A2} are combined to form the FIR filtered charge $Q_{OUT}(k+1)$. We see that interleaved capacitors are required because each residue value must contribute to two FIR filtered outputs and the sampled charge stored on a capacitor is destroyed after a single charge sharing.

Returning to the simplified schematic representation, shown in Figure 4-14, the IIR filter is implemented with a simple, single-stage opamp along with feedback capacitor C_F, which sums and integrates the FIR filter tap charges onto a feedback capacitor. The FIR filter taps are summed and integrated during the relatively long signal-sampling period of the SAR ADC to ensure that there is sufficient time for the filter outputs to settle before the start of the next ADC conversion cycle. The overall filtered residue is given as:

$$V_{OUT}(z) = \left[\frac{C_A}{C_F}z^{-1} + \frac{C_B}{C_F}z^{-2}\right]\frac{\kappa_A}{1-\kappa_A z^{-1}}V_{RES}(z) \qquad (4.10)$$

Figure 4-17 compares the noise transfer function from a lossy IIR filter with the combined FIR-IIR.[27] Even with the low value of 0.6 for κ_A, it is clear that that the FIR-IIR produces better noise attenuation compared to the IIR structure. The better noise attenuation is the result of additional gain from the FIR filter structure. Furthermore the attenuation bandwidth is wide, facilitating a low oversampling ratio. The wider bandwidth results from additional zeros added by the FIR filter. Figure 4-18 compares the NTF of this noise-shaping SAR ADC and the NTF of an ideal delta-sigma ADC. As shown in Figure 4-18, the NTF for this noise-shaping SAR ADC with FIR-IIR loop filter indicates resolution gains equivalent to a third order delta-sigma modulator at an oversampling ratio of 4.[28]

[27] Both the IIR and FIR-IIR use an itegrator with a low κ_A value of 0.6

[28] Mismatch and noise are not included in the comparison.

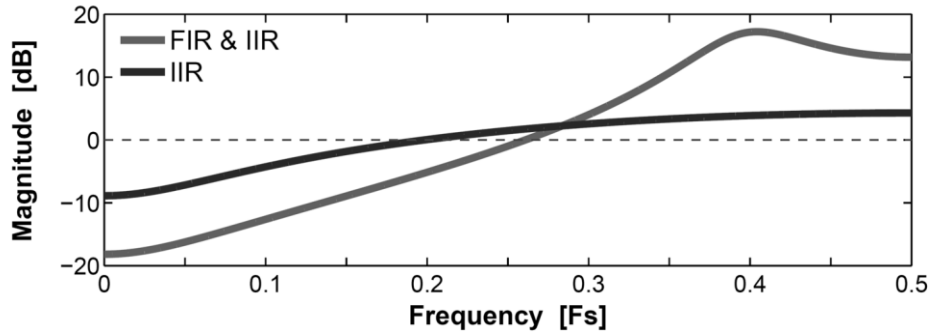

Figure 4-17: Noise transfer function for the IIR filter alone (Section III.C) compared to that of combined IIR with FIR filter.

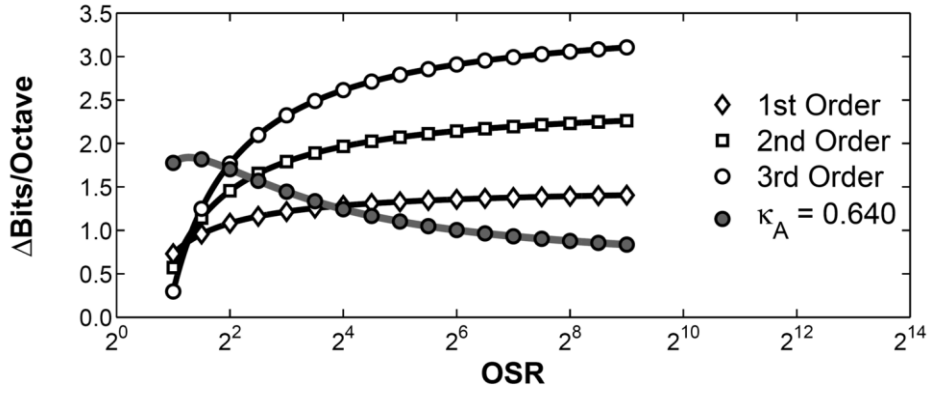

Figure 4-18: Comparison in resolution gains between a noise-shaping SAR ADC using a FIR-IIR loop filter and ideal delta-sigma ADCs. Noise and mismatch are not considered. At a low oversampling ratio of 4, the FIR-IIR filter provides resolution gains comparable to a third order modulator.

4.4 Circuit Details

The SAR ADC is a fully differential implementation of the architecture shown in Figure 4-1, along with a differential realization of the noise shaping circuitry shown in Figure 4-13. The loop filter opamp is a simple single-stage low-gain amplifier. The comparator is shown in Figure 4-19. The comparator compares the differential residue voltage with the differential filtered

residue signal. A simple dynamic structure is employed for energy efficiency and is a double differential version of the double tail latch comparator [49].

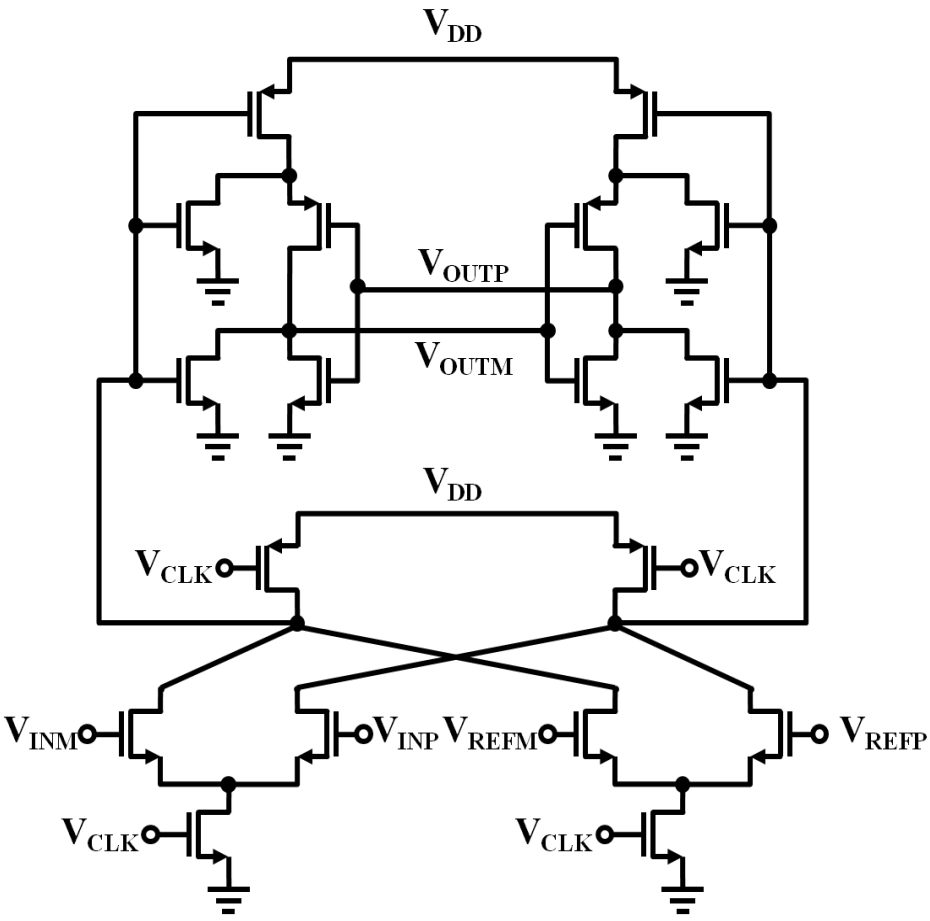

Figure 4-19: Comparator.

To save power and to eliminate the need for a very fast reference clock, the timing for this SAR ADC is generated using an asynchronous clocking scheme. A 90MHz master clock controls the sampling instance, and a single delay element is used to time each of the DAC settling events. The delay element, Figure 4-20, consists of a ring oscillator type structure with one

inversion produced by a flip-flop, which when triggered by the comparator-ready signal, immediately resets the comparator and initiates the inverter delays to time the DAC. By recycling this single delay element, the delays of all DAC settling events match without the use of calibration.

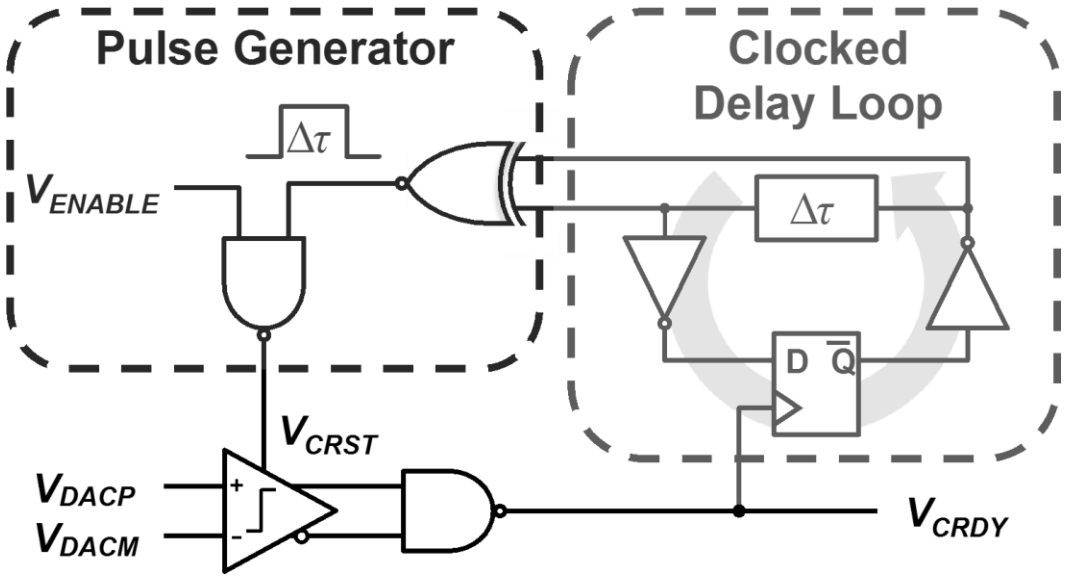

Figure 4-20: Clock generation.

4.5 Prototype and Measurements

The prototype ADC, Figure 4-21, is fabricated in 65nm CMOS. The ADC occupies an area of 0.03mm^2 (231µm by 140µm) and more than half of this area shown is taken up by decoupling capacitance. The DAC is an 8 bit, binary-weighted capacitor array. Each half of the array has a total capacitance of 640fF. The unit capacitors are implemented as stacked, finger capacitors.

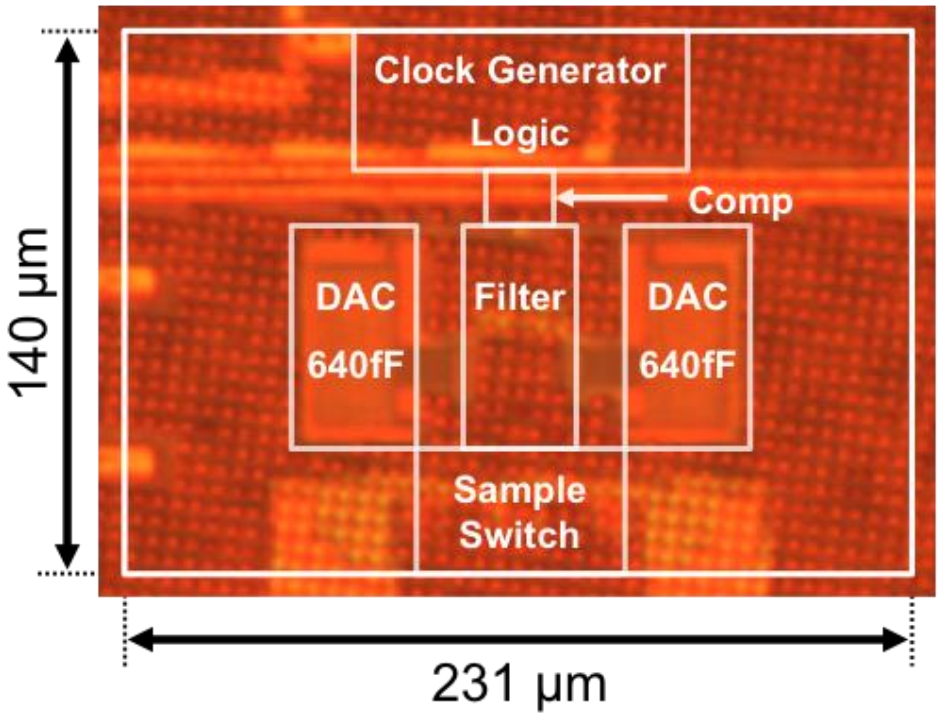

Figure 4-21: Die Photo

The measured spectral density of the converter for a 2MHz input signal sampled at 90MS/s is shown in Figure 4-22. With an OSR of 4, the signal bandwidth is 11 MHz and the ADC achieves a measured ENOB of 10.0 bits with a 2MHz input signal. The measured SFDR is 72dB. With a resolution gain of 2-bits above the 8-bit DAC resolution and a reduction of the signal bandwidth by 4, this noise-shaping SAR ADC trades bandwidth and resolution equally in terms of FOM.

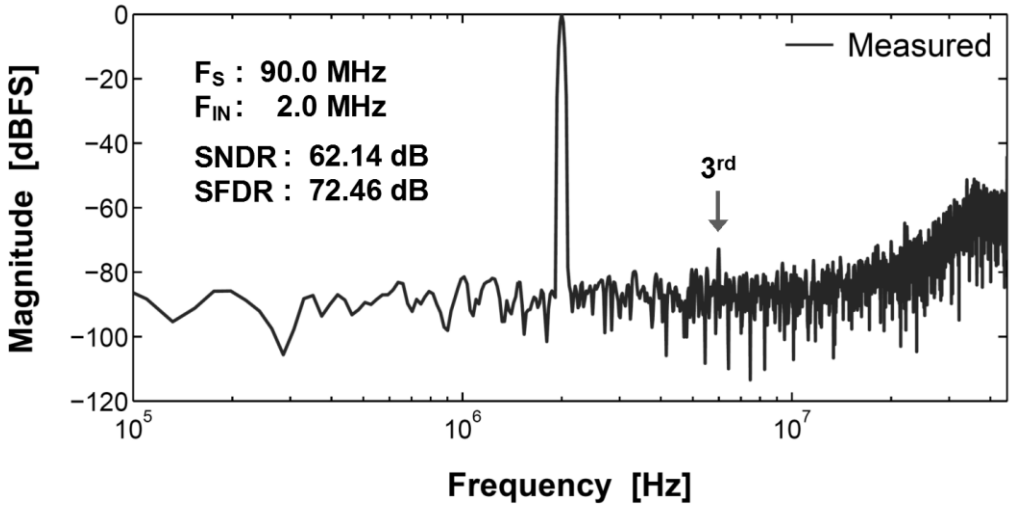

Figure 4-22: The measured spectral density of the converter for a 2MHz input signal sampled at 90MS/s.

The measured SNDR versus input frequency and versus input amplitude are shown in Figures 4-23 and 4-24, respectively. At 90MS/s the total power consumption is 806μW. Of this, the digital power consumption is 608μW, and the analog power consumption is 198μW. The analog power consumption includes 30μW for the comparator, 45μW for the sampling circuit, 44μW for the DAC reference voltages, and 79μW for the FIR-IIR filter. For a 2MHz input the measured FOM for this converter is 35.8fJ/conversion-step.

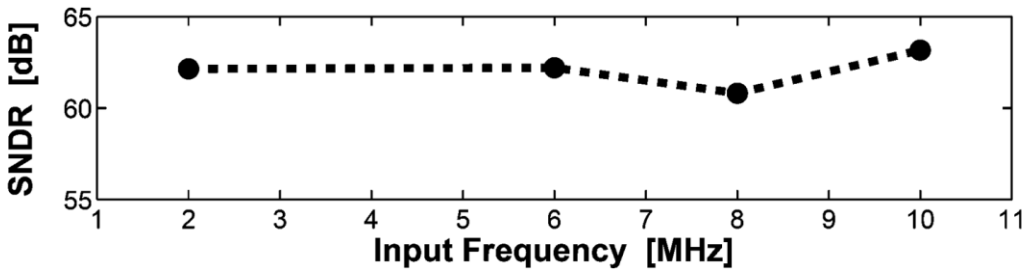

Figure 4-23: Measured SNDR versus frequency with a full-scale input

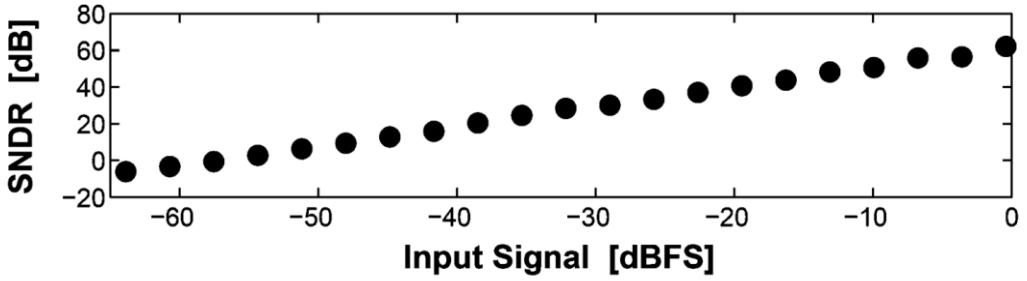

Figure 4-24: Measured SNDR versus input signal amplitude

4.6 Conclusions

This chapter introduces a noise-shaping SAR ADC. This low OSR noise-shaping architecture allows 10 bit ENOB to be achieved with a compact 8 bit DAC array. Noise-shaping shapes both comparator noise and quantization noise, helping to decouple comparator noise from ADC performance. A loop filter comprised of a cascade of a two-tap charge domain FIR filters and an integrator achieves good noise shaping even with a low quality integrator. A wide attenuation bandwidth in noise transfer function facilitates a low over-sampling ratio of 4. A comparison with previously published work is provided in Table 4-1.

TABLE 4-1: COMPARISON WITH OTHER WORK

Specifications	JSSC'10 [34]	VLSI'10 [36]	ISSCC'11 [45]	ISSCC'10 [42]	ISSCC'10 [48]	This Work
Architecture	SAR	SAR	SAR	SAR	SAR	NS-SAR
Technology (nm)	65	180	65	65	90	65
Resolution (bit)	10	10	10	10	10	---
Bandwidth (MHz)	0.5	5	0.01	25	50	11
Power (μW)	1.9	98	0.206	820	1130	806
ENOB (bit)	8.75	9.83	8.84	9.16	9.51	10
FOM (fJ/conv)	4.42	11	22.4	30	15.5	35.8

CHAPTER V

Time-Interleaved MASH SAR

5.1 Introduction

Charge-redistribution successive-approximation (SAR) ADCs dominate low power applications. Because the SAR architecture is highly digital in nature and contains few active components, SAR ADCs achieve phenomenal energy efficiencies [53]. Additionally, noise-shaping SAR ADCs have shown that the extra information produced by the SAR algorithm can be used to construct low-power noise-shaping techniques, which increase the resolution of SAR ADCs and reduce the bottleneck of comparator noise [54]. Our work on noise-shaping SAR ADCs has demonstrated that oversampling can relax both the comparator noise and the sampling *kT/C* noise requirements for SAR ADCs while extending resolutions closer to the limitations of capacitor mismatch, which tend to improve as lithography improves.

This chapter describes a noise-shaping converter built upon time-interleaved SAR ADCs. The prototype ADC consists of four time-interleaved SAR ADC channels, each with a 6 bit binary-weighted capacitor DAC. Like traditional Nyquist-rate SAR ADCs, noise-shaping SAR ADCs present a direct tradeoff between sampling speed and resolution due to the linear progression of the SAR algorithm – sampling speed trades linearly with the resolution as measured in bits. With traditional SAR ADCs, however, time-interleaving has allowed SAR ADCs to extend sampling speeds while maintaining excellent power efficiencies [53]. This

section describes an extension to the time-interleaved technique that extends the bandwidth of noise-shaping SAR ADCs.

5.2 Time-Interleaving

Time-interleaving is a proven technique for combining multiple low bandwidth ADCs into a single high bandwidth ADC. With Nyquist ADCs, time-interleaving involves successive ADC channels converting successive samples of the input signal. Figure 5-1 presents the basic structure of a time-interleaved Nyquist ADC. As shown in Figure 5-1, the time-interleaved ADC consists of multiple Nyquist ADCs, where each channel successively samples and processes the input signal. Because each channel only sees a subset of the samples, the bandwidth requirements for each individual channel, after sampling, are less than the overall ADC bandwidth, thus, lower bandwidth, slower ADCs can be used to construct the channels. With time-interleaving, a higher bandwidth ADC can be constructed using multiple lower bandwidth ADCs.

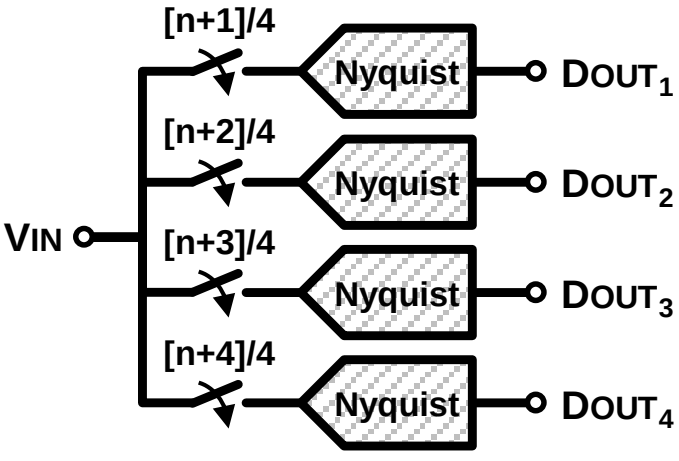

Figure 5-1: Basic time-interleaving structure for Nyquist ADCs

Figure 5-2 plots the output spectrum for the time-interleaved ADC shown in Figure 5-1. Although each of the channels of the time-interleaved ADC from Figure 5-1 subsamples the input signal at a rate of $F_S/4$, the reconstructed time-interleaved spectrum matches the spectrum of an equivalent Nyquist ADC sampling at the full rate of F_S. Time-interleaved converters can, therefore, digitally convert input signals up to the overall converter Nyquist, $F_S/2$. In time-interleaved ADCs, the quantization noise from each channel is aliased across the full Nyquist spectrum of the overall ADC. Although each channel effectively subsamples the input, no such aliasing occurs with the input signal. Only the quantization noise is aliased.

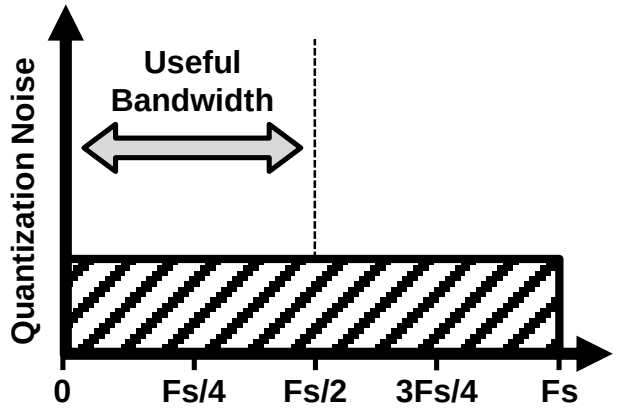

Figure 5-2: Ideal spectrum for time-interleaved Nyquist ADCs – where Fs is expressed as the overall converter sampling rate

Time-interleaved Nyquist ADCs, however, introduce latency into the conversion process. Since each channel of a time-interleaved ADC continues to perform a digital conversion while the next sample is acquired on the free channel, the digital conversion process for a channel does not complete before the next sampling instance of the overall ADC. For Nyquist ADCs, this

time-interleaving latency has no effect the overall ADC operation because time-interleaved Nyquist ADCs do not generally require communication between channels during the digital conversion. Nevertheless, this latency in time-interleaved ADCs can introduce stability problems due to loop delay when time-interleaved ADCs are configured in a mixed-signal feedback loops.

Noise-shaping ADCs, on the other hand, depend on knowing the entire history of the previous conversions from every sample, and unlike Nyquist ADCs, the latency introduced by time-interleaving diminishes the usefulness of noise-shaping ADCs. Figure 5-3 depicts a straightforward time-interleaving of noise-shaping ADCs similar to Figure 5-1. The reconstructed spectrum for the time-interleaved noise-shaping ADCs is presented in Figure 5-4.

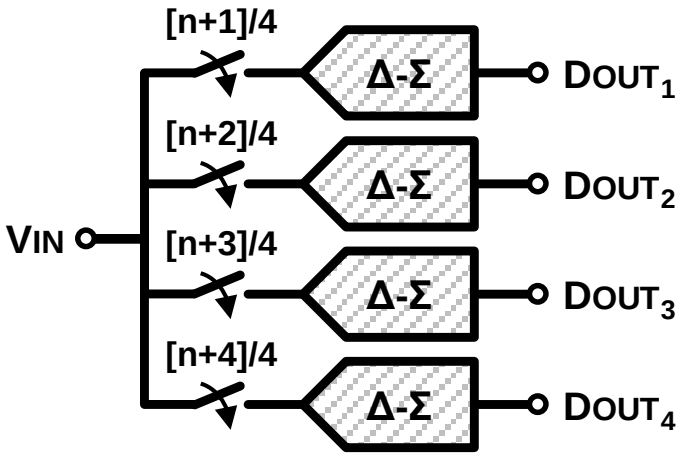

Figure 5-3: Naive time-interleaved delta-sigma ADCs

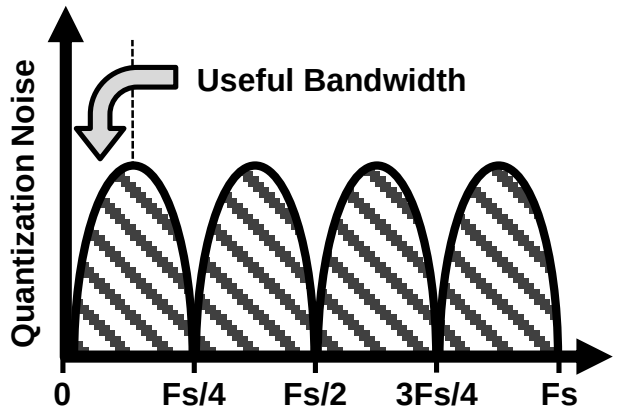

Figure 5-4: Ideal spectrum for naive time-interleaved delta-sigma ADCs, – where Fs is expressed as the overall converter sampling rate

As depicted in Figure 5-4, a straightforward time-interleaving of noise-shaping ADCs produces a spectrum with periodic nulls. These nulls represent the aliasing of quantization noise from each channel. For some applications, these nulls could potentially be useful, but as a general bandwidth extension technique, these nulls defeat the purpose of time-interleaving. Even with time-interleaving, the useful bandwidth over which noise is attenuated does not improve more than the bandwidth of a non-interleaved approach. For instance, in Figure 5-4, the bandwidth of the attenuated noise around DC is the same as using a single noise-shaping ADC at the equivalent channel sampling frequency, $F_S/4$. For noise-shaping ADCs, traditional time-interleaving doesn't extend the bandwidth of the overall converter in the same manner as for Nyquist ADCs.

5.3 Block Level System Description

To overcome the limitations of the naïvely interleaved noise-shaping structure shown in in Figure 5-3, this section introduces a time-interleaved noise-shaping ADC based on the MASH architecture. The overall block level system diagram is shown in Figure 5-5 and described below.

The block diagram in Figure 5-5 consists of three primary signal paths. In the top path, the input signal, *Vin*, is converted directly into a 2-bit output, $Dout_1$. For a typical MASH structure, each of the paths are noise-shaping loops, however, because the magnitude of 2-bit quantization noise introduced by the first conversion, Q_1, is quite large, $\sim V_{FS}/4$, meeting the dynamic range requirement for an integrator based loop filter becomes difficult. As the quantization error scales down in later loops, however, the output swing requirements relax and implementing accurate integrators becomes more feasible. Therefore, the first path directly digitizes the input signal in order to reduce the magnitude of the signal passed to the later noise-shaping loops.

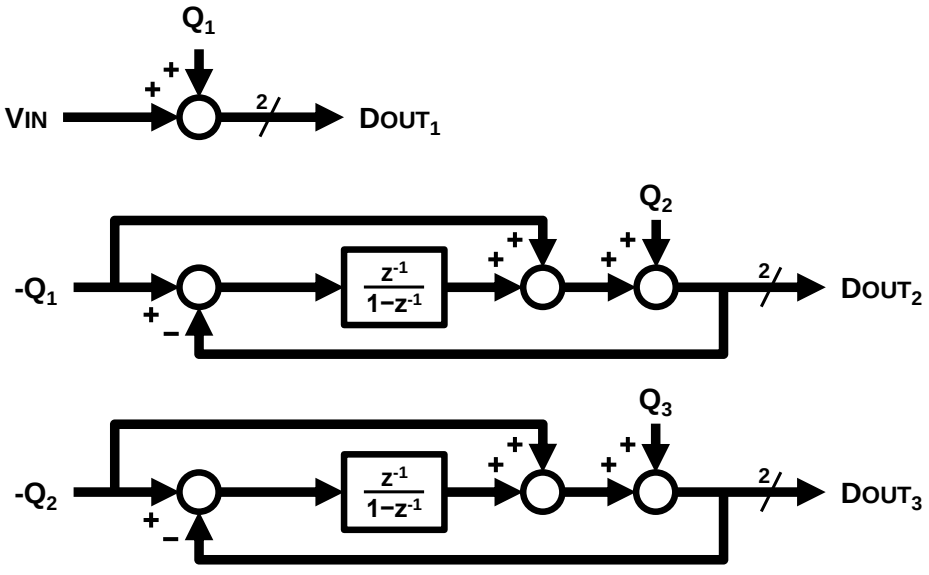

Figure 5-5: Block level system diagram for the time-interleaved noise-shaping ADC

The relationship between *Vin* and $Dout_1$, is shown in (5.1), where Q_1, is the first 2-bit quantization error.

$$Dout_1 = Vin + Q_1 \tag{5.1}$$

In the center path, the quantization error from previous 2-bit MSB conversion, Q_1, is further quantized into another 2-bit output, $Dout_2$, but unlike the top path, the quantization error, Q_2, is noised-shaped by the feedback loop (5.2).

$$Dout_2 = -Q_1 + (1 - z^{-1})Q_2 \tag{5.2}$$

In the bottom path, the quantization error from the center path, Q_2, is again noise shaped. $Dout_3$ is described by (5.3).

$$Dout_3 = -Q_2 + (1 - z^{-1})Q_3 \tag{5.3}$$

Exactly like a MASH structure, the digital outputs are summed after digital filtering. The transfer function for the weighted summation is described in (5.4),

$$Dout = Dout_1 + Dout_2 + (1 - z^{-1})Dout_3 \tag{5.4}$$

Simplifying *Dout* in terms of the quantization error reveals a second-order noise transfer function (5.5) – where the magnitude of Q_3 is set by the *LSB* of the SAR capacitor DAC, $\sim V_{FS}/2^6$.

$$Dout = Vin + (1 - z^{-1})^2 Q_3 \tag{5.5}$$

In this MASH structure, Figure 5-5, the quantization error is passed through multiple noise-shaping loops, which produces higher order noise-shaping. Since each loop operates in series and does not depend on information from later loops to operate, this MASH structure can be parallelized and time-interleaved – unlike the naïve interleaving shown in Figure 5-3.

5.4 Circuit level Implementation

This section describes the circuit level implementation for the system shown in Figure 5-5. For the sake of clarity, this section is broken into two parts. The first section describes a simplified, non-interleaved implementation of the system where different conversion cycles from the SAR algorithm are used to implement the MASH stages. The second section describes a time-interleaved implementation.

5.4.A Simplified Single Channel Circuit Implementation

A schematic for the simplified circuit is given in Figure 5-6. The circuit topology is essentially a SAR ADC with the addition of two integrators. The SAR portion of the circuit contains a 4 bit binary-weighted capacitor DAC, an input sampling switch, and a comparator.[29] The inputs of both integrators connect to top plate the capacitor DAC and the outputs both connect to the comparator reference terminal through a mux. In this configuration, the integrators can accumulate the DAC residue voltages and pass this accumulated value to the comparator reference node. For clarity, the top path from the system block diagram, Figure 5-5, is ignored in this simplified implementation – only the two noise-shaping feedback loops are considered.

[29] For simplicity, the SAR feedback controller is neglected from Figure 5-6.

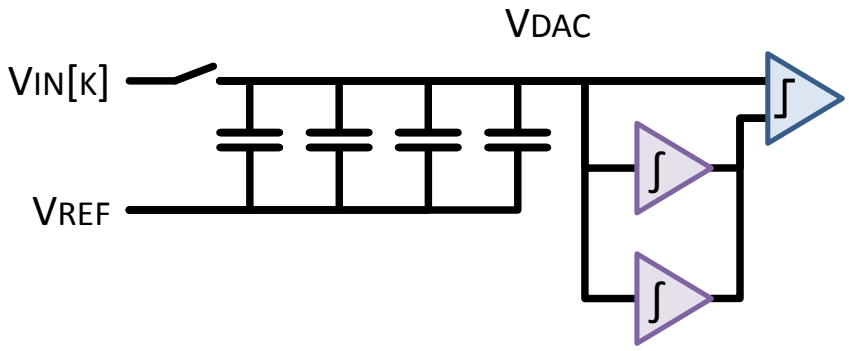

Figure 5-6: Simplified single channel implementation. Circuit contains a SAR ADC with two additional integrators.

The operation of the single channel implementation shown in Figure 5-6 can be broken down into discrete steps. At the onset of the conversion, the two integrators are disabled, and the input signal is sampled onto the cap array of the SAR ADC, Figure 5-7.

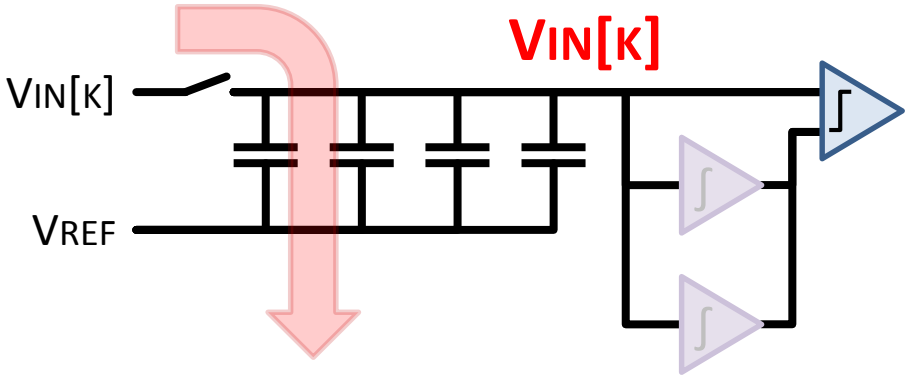

Figure 5-7: Sampling Operation

Following this sampling operation, the top integrator is connected to the comparator, and a 2-bit digital conversion is performed with the accumulated residues from the top integrator used as a reference for the conversion, Figure 5-8.

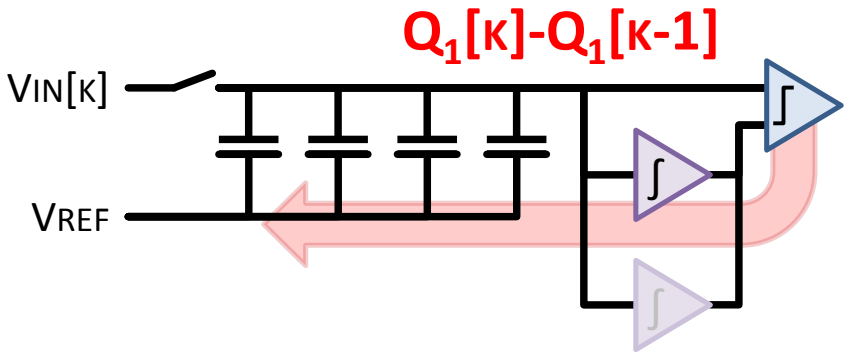

Figure 5-8: Noise-shaping digital conversion (first loop)

After the digital conversion is completed, the residue voltage on DAC top plate voltage equals the noise shaped residue quantization error, $Q_1[k]$-$Q_1[k-1]$, where Q_1 is the quantization error from the two MSB bits. Intuitively, this noise-shaped quantization error results from the ADC using the integrator output as the comparator reference for the SAR conversion. Since the integrator already stores the quantization error from the previous conversion, the previous quantization error, $Q_1[k-1]$, is subtracted from $Q_1[k]$.

Following, the first 2-bit conversion, the top integrator accumulates the DAC residue, Figure 5-9. This accumulation updates the top integrator with the quantization error from the first 2-bit conversion, $Q_1[k]$, which will be used during the next nsampling cycle.

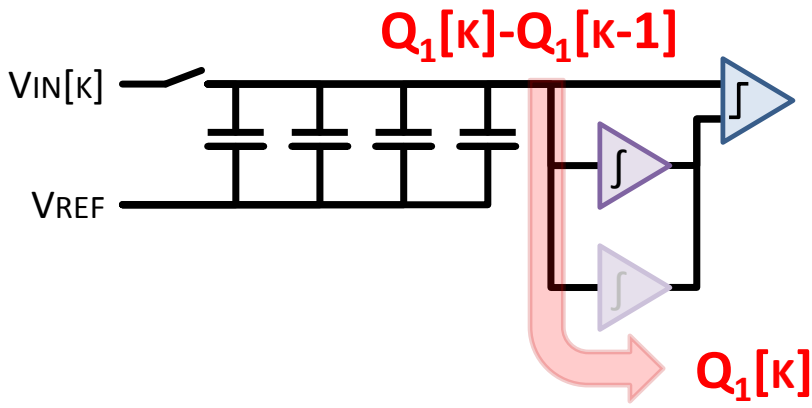

Figure 5-9: Integrate the DAC residue (first loop)

With the first noise-shaping loop completed, the digital conversion continues by performing the second noise shaping loop. In a MASH structure, the unshaped quantization noise from the preceding stages is fed into later noise-shaping stages. As show in Figure 5-9, however, the DAC residue voltage shows a noise-shaped quantization error, $Q_1[k]-Q_1[k-1]$. Since the DAC voltage represents the signal that is processed, the circuit synthesizes an unshaped quantization error on the DAC by integrating the output from the top integrator onto the bottom integrator, Figure 5-10.

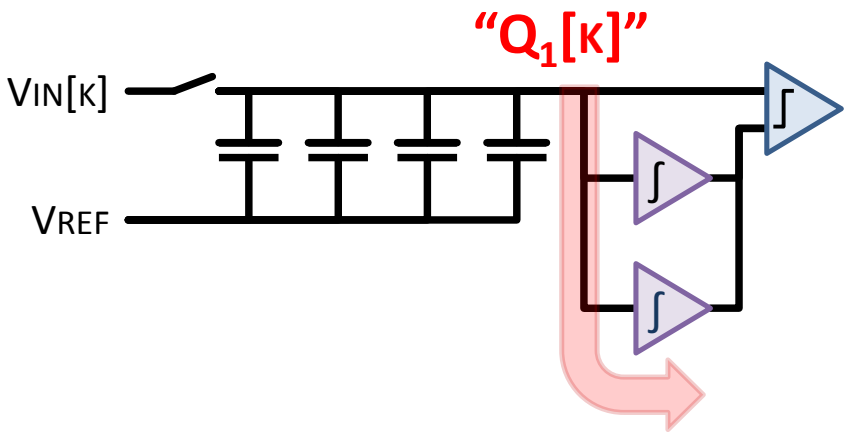

Figure 5-10: Integrate the output of the first integrator onto the second integrator

By integrating the output form the top integrator onto the bottom integrator, the circuit essentially level shifts the reference voltage for the conversion so that the DAC residue voltage appears to hold $Q_1[k]$. The actual residue voltage on the DAC is still $Q_1[k]$-$Q_1[k-1]$, but the output of the second integrator has been shifted by $Q_1[k-1]$ so that the second conversion can proceed as if $Q_1[k]$ is the DAC voltage.[30] Figure 5-11 shows the second digital conversion.

[30] Although not equivalent, this is very similar to connecting two integrators in series as in a second order delta-sigma.

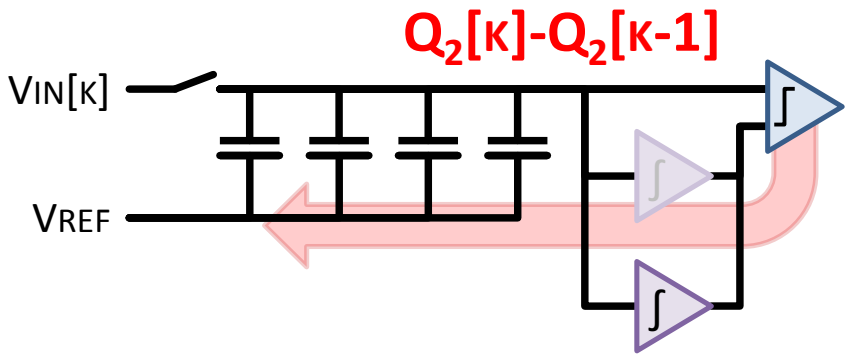

Figure 5-11: Noise-shaping digital conversion (second loop)

Similar to the digital conversion in the first noise-shaping loop, the second digital conversion produces a noise-shaped residue voltage, $Q_2[k]$-$Q_2[k-1]$, where Q_2 is quantization error from the two LSB bits. The second noise-shaping loop finishes by updating the bottom integrator output with the DAC residue voltage, Figure 5-12.

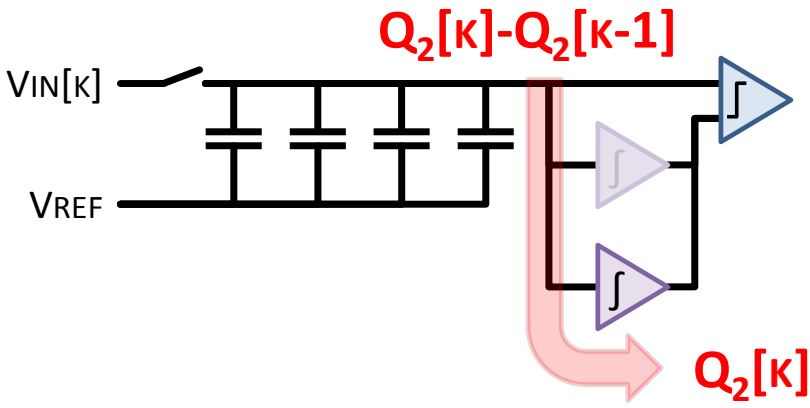

Figure 5-12: Integrate the DAC residue (second loop)

After the second noise-shaping loop finishes, the cycle repeats by sampling input signal as in Figure 5-7. Although the circuit described in this section processes the noise-shaping loop sequentially, the circuit performs a MASH conversion similar to the system level block diagram shown in Figure 5-5.

5.4.B Simplified Time-Interleaved Circuit Implementation

In this section we outline a simplified time-interleaved implementation of the MASH structure given in Figure 5-5. A simplified schematic for the time-interleaved structure is given in Figure 5-13.

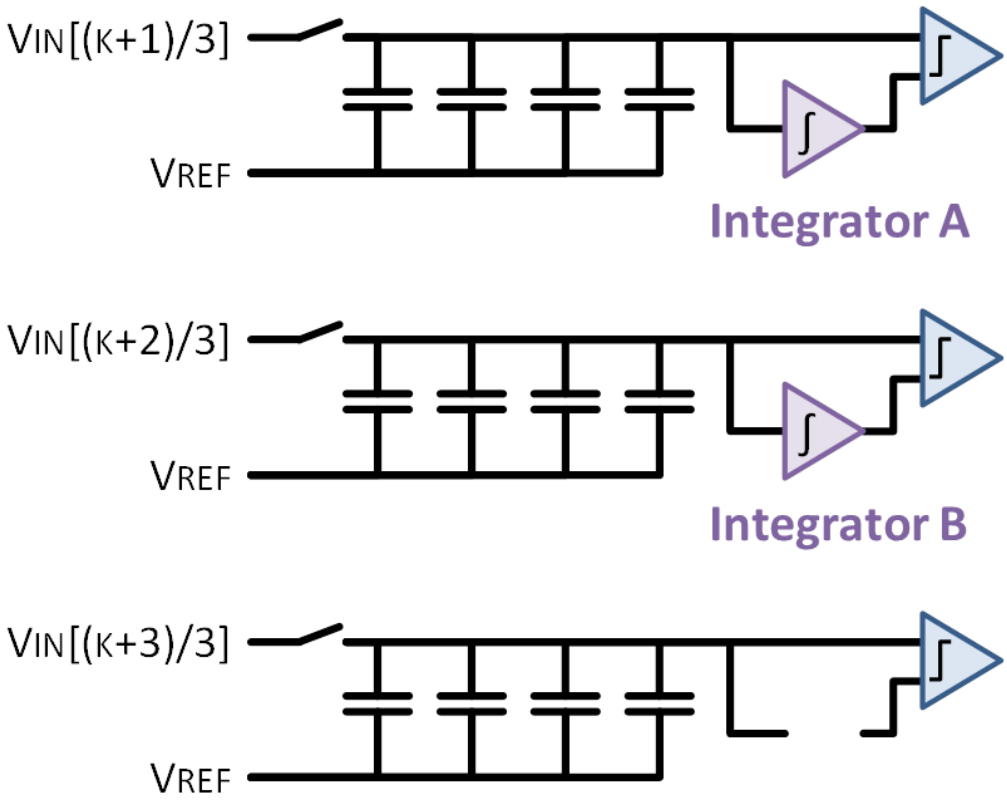

Figure 5-13: Time-interleaved MASH with three channels

Figure 5-13 shows a simplified three-channel implementation for the MASH structure shown in Figure 5-5. This simplified three-channel implementation consists of the three SAR ADCs, each with a 4 bit binary weight capacitor array and comparator, and two integrators. The operation of this time-interleaved ADC parallels the operation described in Section 5.4.A for a signal channel.

In the first step, Channel 1 disconnects the integrator and samples the input signal. This sampling operation occurs in parallel with Channel 2 integrating the DAC residue voltage onto Integrator B and Channel 3 integrating the DAC residue voltage onto Integrator A. Following the sampling operation for Channel 1, Channel 1 connects to integrator A and performs a noise-shaping digital conversion of the MSB bits. Simultaneously, Channel 2 begins sampling the input signal and Channel 3 connects with integrator B to perform a noise-shaping digital conversion of the LSB bits. After Channels 1 and 3 finishes the digital conversions and updates the integrator, Channel 1 connects to integrator B to convert the LSB bits, Channel 2 connects to integrator A to convert MSB bits, and Channel 3 begins samplings.

5.5 Prototype and Measurements

The prototype ADC, Figure 5-14, is designed in 65nm CMOS. The ADC occupies an area of $0.62mm^2$ (960μm by 640μm). The DAC for each channel is a 6 bit, binary-weighted capacitor array. Each half of the array has a total capacitance of 600fF. The unit capacitors are implemented as stacked, finger capacitors, with the top plate shielded on all sides by additional bottom plate routing. Although shielding the top plate with the bottom plate substantially increases the bottom plate parasitic capacitance, this shielding enhances the array linearity by eliminating most of the stray capacitance between the top plate and the bottom plate routing.

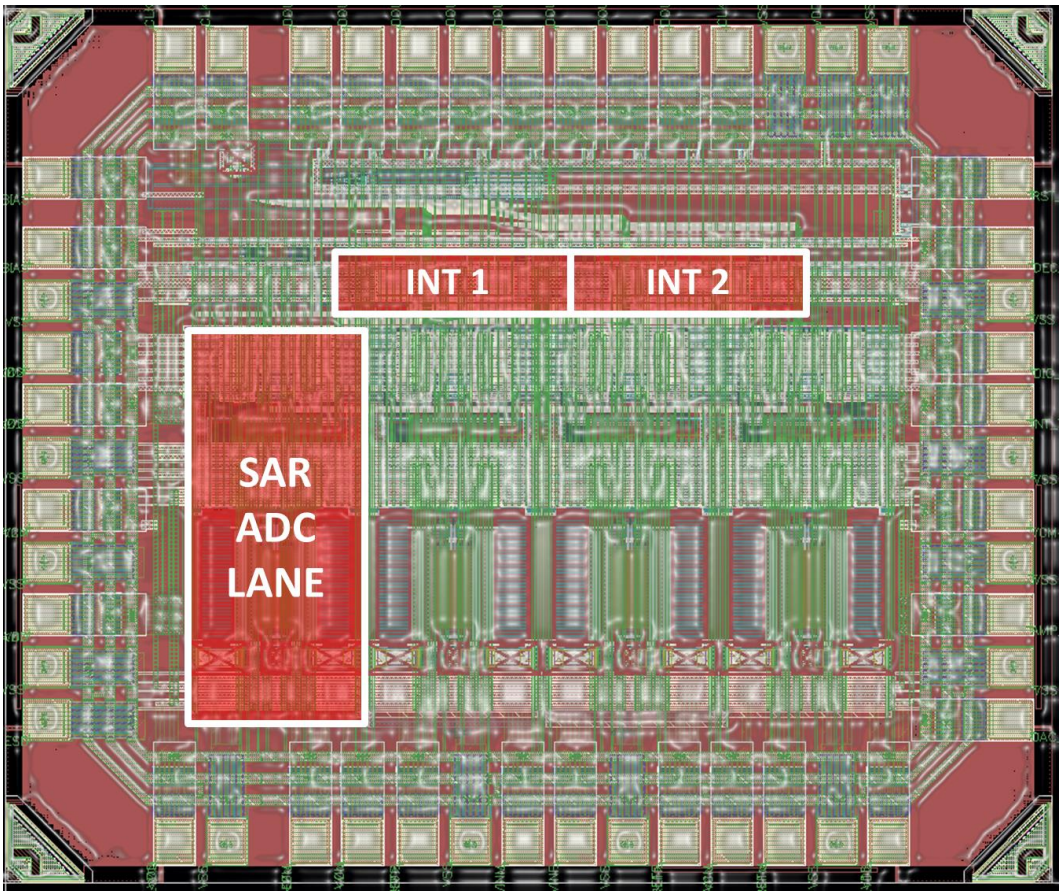

Figure 5-14: Die Screenshot (1.2mm x 1.0mm)

The simulated spectral density of the time-interleaved converter for a 3.42 MHz input signal sampled at 250MS/s is shown in Figure 5-15. As shown in Figure 5-15, the noise floor displays the characteristic 40dB/decade slope of 2nd-order noise shaping systems. The SNDR of the converter is graphed as function of the oversample ratio (OSR) in Figure 5-16 – where the OSR is expressed in octave. At an OSR of 16 (4 octaves), the time-interleaved converter achieves an SNDR of 61.3 dB with a 3.42 MHz input signal. At 250MS/s, the estimated power consumption is 10mW.

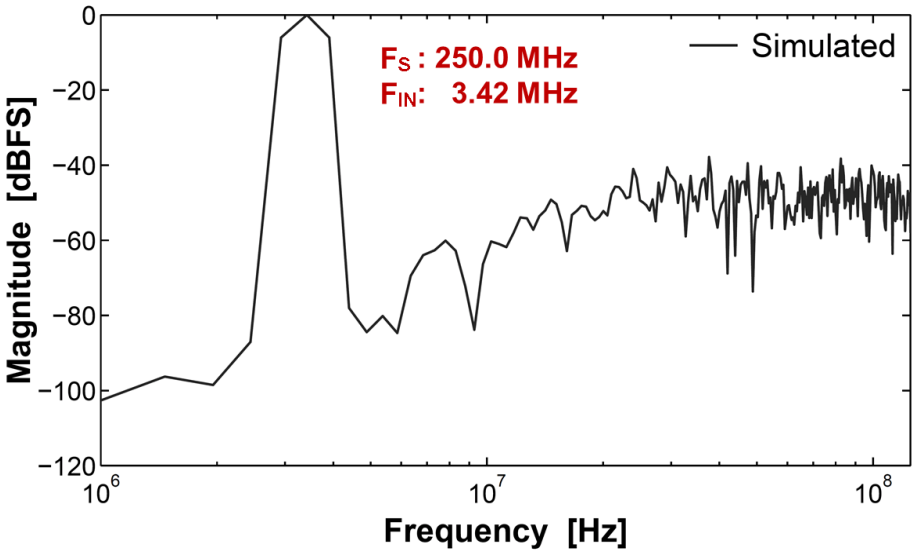

Figure 5-15: Simulated spectral density of the time-interleaved converter for a 3.42MHz input signal sampled at 250MS/s.

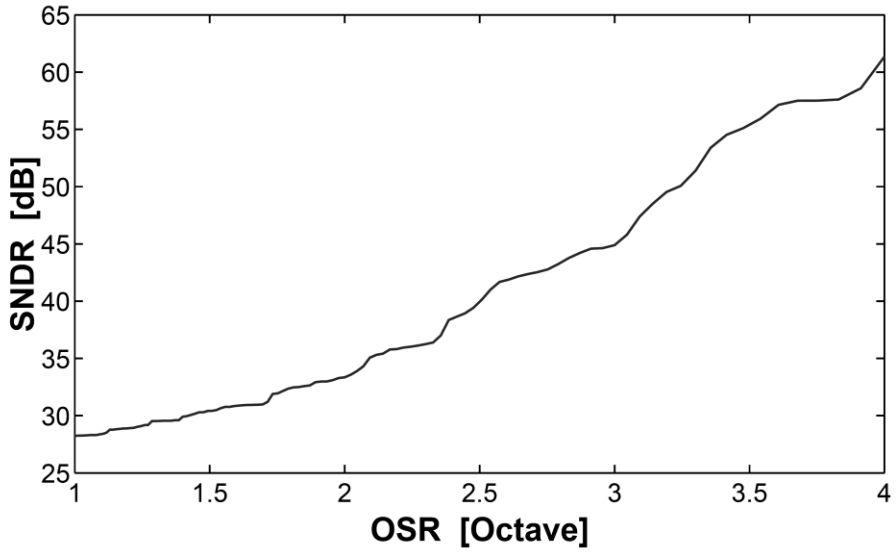

Figure 5-16: Simulated SNDR versus OSR for the measured spectral density given in Figure 5-15.

5.6 Conclusions

This section describes a time-interleaved MASH ADC using multiple SAR ADC channels. Like traditional time-interleaving, each of the SAR ADC channels performs time-interleaved sampling, which improves the bandwidth of the overall converter up to the speed of a 2 bit SAR Unlike traditional time-interleaving, however, the loop filter is time-shared between channels to create 2-bit noise-shaping loops. When the digital outputs from the all of channels are combined, the overall ADC shows second order noise shaping with a $V_{FS}/2^6$ LSB size. Although this method of interleaving does not reduce the bandwidth requirements of the loop filter integrators, which ultimately limit the speed of the ADC, the additional channels increase the order of the noise-shaping while providing a moderate bandwidth improvement of 30% to the noise-shaping SAR ADC described in CHAPTER IV. This 2nd-order noise-shaping architecture achieves 10 bit ENOB at *16x* OSR with *4x* time-interleaved noise shaping SAR ADCs. Noise-shaping reduces both the comparator noise and the quantization noise. A comparison with previously published work is provided in Table 5-1.

TABLE 5-1: COMPARISON WITH OTHER WORK

Specifications	JSSC'10 [34]	VLSI'10 [36]	ISSCC'11 [45]	ISSCC'10 [42]	ISSCC'10 [48]	This Work
Architecture	SAR	SAR	SAR	SAR	SAR	TI-MASH SAR
Technology (nm)	65	180	65	65	90	65
Resolution (bit)	10	10	10	10	10	---
Bandwidth (MHz)	0.5	5	0.01	25	50	15.6
Power (μW)	1.9	98	0.206	820	1130	10e3
ENOB (bit)	8.75	9.83	8.84	9.16	9.51	10
FOM (fJ/conv)	4.42	11	22.4	30	15.5	313

CHAPTER VI

Conclusion

6.1 Contributions

In Chapter 2, the energy efficiency for the analog portions of charge-redistribution SAR ADCs are examined. First, we derive an expression for the DAC switching energy of a SAR ADC, which provides insight into the fundamental energy limit of the DAC. We also analyze the transient operation of the DAC and calculate the DAC settling time, which allows us to estimate the maximum sampling frequency of a SAR ADC. Finally, we derive an expression for the minimum energy consumption of a regenerative comparator that does not depend on technology parameters such as threshold voltage, transition frequency, trans-conductance, etc. The analysis provides an estimate for the lower bound of the comparator energy consumption in terms of the ADC bit resolution and the comparator input referred noise.

In Chapter 3, the effects of capacitor mismatch on ADC resolution and yield are examined. We develop an alternative statistical model using ENOB as a yield metric. First, we examine the effects of mismatch in a binary weighted, charge redistribution SAR ADC. We then derive an exact algebraic formulation relating capacitor mismatch to the average noise power of the ADC output, and from this algebraic formulation, we derive ENOB as a function of capacitor mismatch. Next, we explore the statistics of this ENOB expression and develop a statistical expression that predicts yield in terms of ENOB and mismatch. Finally, we generalize the results of this work by presenting a compact design equation, which accurately relates resolution,

mismatch, and ENOB to yield for all binary weighted, ratiometric converters. The design equation offered is accurate to within ±0.17 bits for yield values between 0.5% and 99.5% and is consistent with standard test methodology.

In Chapter 4, we describe a hybrid noise-shaping SAR ADC which combined a SAR ADC with a switch cap FIR filter to produce a low energy, moderate resolution oversampling ADC. Although charge redistribution successive approximation (SAR) ADCs are highly efficient, comparator noise and other effects limit the most efficient operation to below 10 bits ENOB. This work introduces an oversampling, noise-shaping SAR ADC architecture that achieves 10b ENOB with an 8-bit SAR DAC array. A noise-shaping scheme shapes both comparator noise and quantization noise, thereby decoupling comparator noise from ADC performance. The loop filter is comprised of a cascade of a two-tap charge-domain FIR filter and an integrator to achieve good noise shaping even with a low quality integrator. The prototype ADC is fabricated in 65nm CMOS and occupies a core area of 0.03mm^2. Operating at 90MS/s, it consumes 806μW from a 1.2V supply.

Chapter 5 describes an extension to the noise-shaping SAR ADC from Chapter 4. We present a noise-shaping ADC structure that combines a time-interleaved SAR ADC with a delta-sigma MASH ADC to produce a time-interleaved oversampling converter. Although the time-interleaved MASH ADC described in Chapter 5 does not solve all the issues for time-interleaving general delta-sigma ADCs, the time-interleaved MASH ADC leverages time-interleaving to expand the bandwidth of noise-shaping SAR ADCs.

6.2 Future Research Directions

6.2.A Pseudo-Non-Causal Noise-Shaping SAR ADC

By definition, non-causal filters cannot be implemented in any practical system because such filters require knowledge of the future state of system. We will address this concern in a moment, but let us begin by exploring what we could achieve if non-causal filters could be practically realized. Figure 6-1 shows the block diagram for a proportional controller with a feedforward path, constant forward path gain, *A*, and a single delay, z^{-1}, in the feedback path.

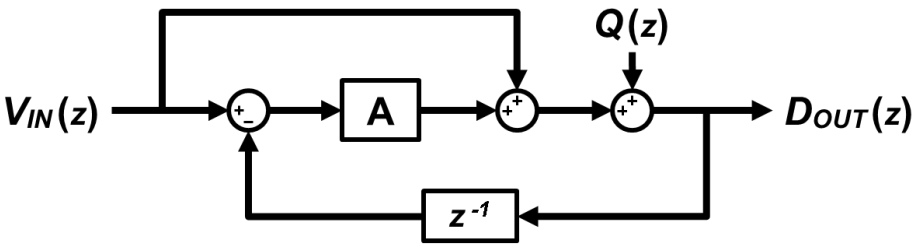

Figure 6-1: Block diagram of simple proportional controller with a feedforward path, a constant forward path gain, and delay in the feedback path.

The closed-loop transfer function is described by (6.1) – where $V_{IN}(z)$ represents the input signal in the *z*-domain, $Q(z)$ represents the quantization noise, and $D_{OUT}(z)$ is the digital output.

$$D_{OUT}(\mathrm{z}) = \frac{1+A}{1+Az^{-1}} V_{IN}(\mathrm{z}) + \frac{1}{1+Az^{-1}} \mathrm{Q}(\mathrm{z}) \tag{6.1}$$

Both the signal-transfer function (STF) and the noise-transfer function (NTF) from (6.1) have an unstable pole when $A \geq 1$. If we ignore this instability concern for a moment, we can asymptotically approximate (6.1) with (6.2) when forward gain is really big, $A \gg 1$.

$$D_{OUT}(\mathrm{z}) = \mathrm{z}V_{IN}(\mathrm{z}) + \frac{z}{A}\mathrm{Q}(\mathrm{z}) \tag{6.2}$$

Examining the STF and NTF from (6.2), we see that the magnitude response of the STF is all-pass and the magnitude of the NTF attenuates the quantization across the entire bandwidth of the system. The problem, however, is that both the STF and NTF have non-causal phase responses and can never be achieved in practice. Although (6.2) describes a system where future samples of the input signal, $V_{IN}(z)$, and the quantization noise, $Q(z)$, appear in time before the signal has even arrived, *if non-causal filters were practical*, we could suppress all the quantization of the system throughout the entire bandwidth of the system – which would be an extremely powerful technique to improve the resolution of SAR ADCs and all noise-shaping systems.

Due to the unstable pole, we cannot directly implement system described in Figure 6-1, but we can implement a similar system. In Table 6-1, we describe a practical process for implementing pseudo-non-casual FIR filters with a SAR ADC. If we assume that our ADC is constructed using five separate capacitor DACs, we can time-interleaved the sampling operation across each of the DAC. Although we only perform a digital conversion of the signal sampled on the first DAC, time-interleaved sampling continues across each of the other DACs.

TABLE 6-1: TIME-INTERLEAVED SAMPLING FOR A SAR ADC

DAC #1	SAMPLE	BITS 7-6	BITS 5-4	BITS 3-2	BITS 1-0
DAC #2		SAMPLE			
DAC #3			SAMPLE		
DAC #4				SAMPLE	
DAC #5					SAMPLE

When the digital conversion from the first channel finishes, we can feed the digital output from the first channel to each of the other DACs. If the input frequency is small relative to *overall* time-interleaved sampling, the digital representation from the first channel will serve as a decent approximation for the other samples held by the additional DACs.

After feeding the digital output code from the first DAC to the other DACs, we can perform a weighted sum of the residue charge from each of the samples. Relative to the sampling instance on the first DAC, this weighted sum can be described by the FIR filter transfer function shown in (6.3) – where $V_{IN}(z)$ is the input signal, $D_{OUT}(z)$ is the digital code from the first channel, $V_{OUT}(z)$ is the output of the filter, and α are the FIR filter weights.

$$V_{OUT}(\mathrm{z}) = \left(1 + \sum_{k=1}^{4} \alpha_k z^{k/5}\right) V_{IN}(\mathrm{z}) \; - D_{OUT}(\mathrm{z}) \tag{6.3}$$

There are two important things to notice about the filter equation shown in (6.3). First, the delays associated with each of the filter taps are positive. With positive delay, each of the filter taps represents a future sample of the input signal. Typically, future samples are prohibited in systems, but due to the latency of the SAR conversion process, we were able to accumulate samples of the input signal that appear as future samples relative to the sampling instance of the

first channel. The filter described by (6.3) is pseudo-non-casual. It is not strictly non-causal because we are not using samples of the input signal before it is even possible to sample, but relative to the original input sample on the first DAC, the filter can contain pseudo-non-causal taps up to the limit allowed by the latency of the SAR operation.

The second important point about (6.3) is the fractional exponents of the delay terms. Typically, the exponents for *z*-domain transfer functions are represented as whole numbers, which allows the interpreting the exponent as unit delays with a duration equal to the period of the sampling clock. With fractional delays, however, the apparent sampling rate is much higher than the clock rate at which the digital outputs are produces. In this case, the time-interleaved sampling is *5x* greater than the rate at which the first channel produces digital outputs. This means that the bandwidth of the FIR filter in (6.3) is five times greater than what could be achieved with non-interleaved sampling. In effect, we get an additional *5x* enhancement to noise-shaping oversampling rate of the converter.

A block diagram for a hypothetical system is provided in Figure 6-2. As shown in Figure 6-2, the pseudo-non-causal filter takes the form of the pre-compensation filter, *C(z).*

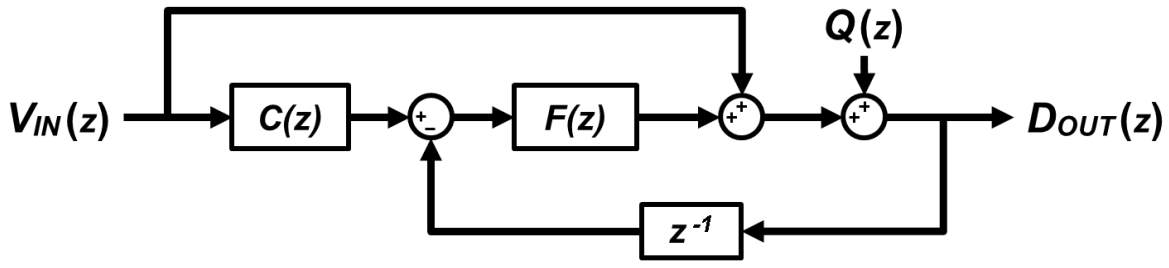

Figure 6-2: Block diagram for a SAR noise-shaping system using a pseudo-non-casual filter, *C(z)*. The transfer function *C*(z) can be implanted similar to (6.3).

In the simplest implantation, each of the filter taps, α, as described in (6.3), are equal to unity. For unity weighted taps, and for a DC input signal, the pre-compensation filter, *C(z)*, will produce an effective *5x* increase to the loop gain of the system. This *5x* gain produced by the passive FIR results from summing the same residual charge from each of the five DACs after the digital output from the first channel is applied to each of the DACs. Since the input signal is DC, each DAC produces the same residue. Furthermore, this *5x* gain will suppress quantization across the whole "Nyquist band" of the non-interleaved sample rate.

Due to the time-interleaved sampling, and the pseudo-non-casual filter, we can produce pseudo-non-casual STFs and NTFs, which allows us to develop very powerful noise-shaping systems that do extraordinary things like the one shown in Figure 6-1. With this system, we can suppress noise within the bandwidth of the first channel, and push the noise into the artificial bandwidth of the time-interleaved system. After digital processing and digital filtering, we can recover a NTZ that appears non-causal within the single channel bandwidth, and we can push the bandwidth over which we improve resolution much close to the single channel bandwidth. Furthermore, we save power from the digital controller because we are only running one comparator and one digital controller, so this system seems suitable for low power, low bandwidth applications.

6.2.B Statistical Analysis of the SFDR yield for a Binary Weighted DAC

In CHAPTER III, we presented a statistical analysis of yield in terms of ENOB for a binary weighted DAC. A derivation of the yield in terms of spurious-free-dynamic-range (SFDR) is an important extension of that work. In some applications, ENOB is an effective measure of performance, but in other application, SFDR may be more important. In this section, we discuss a possible strategy for developing an SFDR probability model.

In Figure 6-3, we show the typical transfer function of a Nyquist-rate ADC. Since we digitally approximate the input signal with discrete digital values, the transfer function assumes a staircase shape and the quantization error assume a saw-tooth shape.

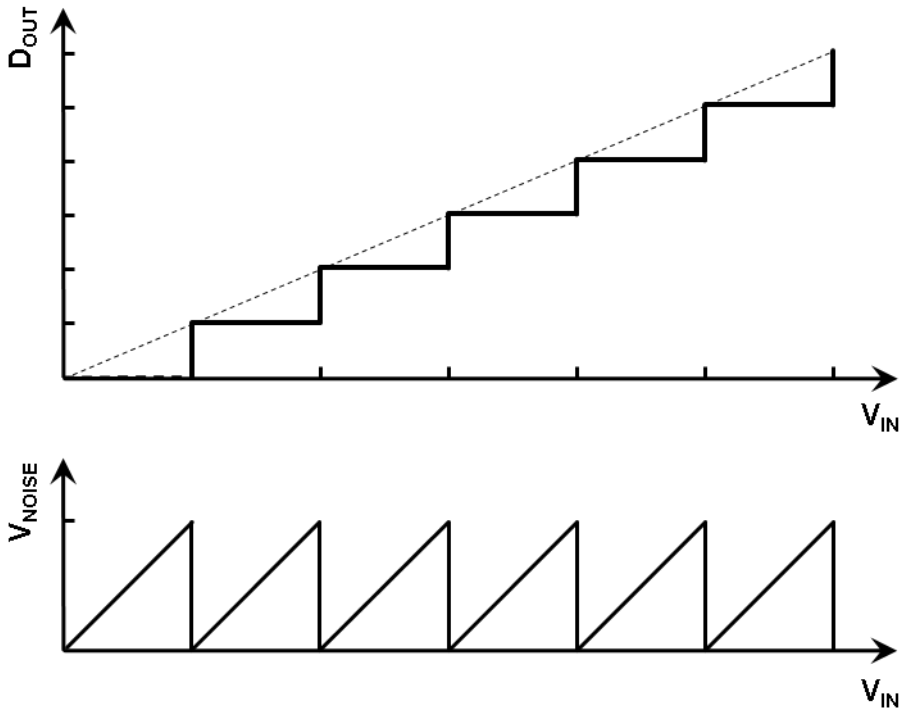

Figure 6-3: Transfer function of a typical Nyquist-rate ADC.

If we perform a polynomial expansion of the transfer function show in Figure 6-3, we can derive an approximation for the transfer function as provided in (6.4) – where *n* represents the nth power of V_{IN}.

$$D_{OUT} = \sum \beta_n {V_{IN}}^n \tag{6.4}$$

The coefficients, β, from (6.4) characterize the magnitude of the harmonics generated from the ADC transfer function. For a very linear ADC, $\beta_1 = 1$, and every other β assume some value which makes the quantization noise floor look flat. If we summed the power of the coefficients, β, we would derive the expected $LSB^2/12$ noise power for an ADC. For a nonlinear ADC, the magnitude the n-th coefficient, β_n, would tell us how much power to expect from the n-th harmonic, which provides a means to calculate the SFDR. The analysis from CHAPTER III shows how to relate the DNL and INL in terms of the noise power for a binary-weighted DAC, and the foundation for that analysis provides enough insight about DNL and INL to formulate a probabilistic derivation for SFDR using (6.4).

APPENDICES

APPENDIX A

Correction Factor for Sinusoidal Distributions

In this Appendix, we derive a linear scaling factor α which estimates the noise power contributed by sinusoidal signals in terms of the noise power contributed by uniformly distributed signals. As shown by (3.4) and (3.5), the average noise power of a single-ended DAC is expressed as the sum of the quantization noise and the mean squared INL. This noise power formulation suggests that we can obtain the noise power for a sinusoidal distribution by reweighting the mean squared INL values according to the probability mass function of a sinusoidal distribution. In what follows, we first develop an expression for the squared INL values and then use this expression to calculate the average INL noise power contributed by both uniform and sinusoidal DAC code distributions. Finally, we extract the linear scaling factor α from the ratio of these noise powers.

We obtain an analytic formulation of the of the squared INL values by simulating the INL of mismatched differential DACs and numerically calculating the mean squared INL at each code. Figure A-1 graphs the simulated mean square INL values, and (A.1) provides a quadratic approximation of the results – where Φ_i is the INL at code i, N is the number of bits, and A is the peak amplitude of the squared INL values.

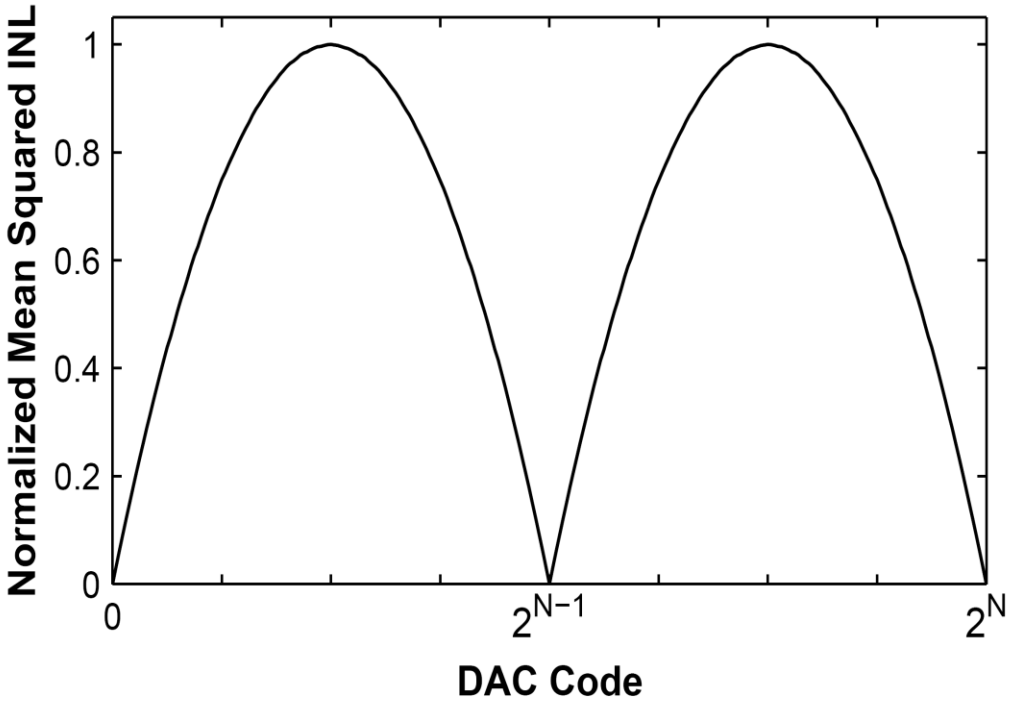

Figure A-1: Simulated values of the normalized mean squared INL at each code of an N bit DAC. The INL values were obtained by averaging the squared INL values at each code across randomly mismatched DACs.

$$\Phi_i{}^2 = \begin{cases} A \cdot \dfrac{i \cdot (2^{N-1} - i)}{2^{2N-4}} & 0 \le i \le 2^{N-1} \\ A \cdot \dfrac{(2^N - i)(i - 2^{N-1})}{2^{2N-4}} & 2^{N-1} \le i \le 2^N \end{cases} \tag{A.1}$$

To simplify the calculations, we normalize the code range of (A.1) to half a period. The normalized INL expression is given in (A.2) – where *x* is the normalized index variable.

$$\Phi_i{}^2 = \begin{cases} 4A \cdot x(1 - x) & 0 \le x \le 1 \\ 4A \cdot (x - 1)(2 - x) & 1 \le x \le 2 \end{cases} \tag{A.2}$$

Assuming an infinite resolution DAC with continuously distributed INL values, we estimate the average noise power contributed from uniformly distributed INL errors by integrating (A.2) across one period of the normalized code range. This calculation is shown in (A.3).

$$V_{noise,INL}{}^2 = 4A\int_0^1 x(1-x)dx = \frac{2A}{3} \tag{A.3}$$

Similarly, we calculate the average noise power contributed from sinusoidally distributed INL errors by weighting the squared INL values according to a sinusoidal probability distribution and again integrating across a normalized code range. This calculation is shown in (A.4).

$$V_{noise,INL}{}^2 = \frac{8A}{\pi}\int_0^1 \frac{x(1-x)}{\sqrt{1-x^2}}dx = \frac{2A(4-\pi)}{\pi} \tag{A.4}$$

Letting α equal the ratio of these noise contributions, we obtain the estimate for α given in (A.5).

$$\alpha = \frac{3(4-\pi)}{\pi} \cong 0.8197 \tag{A.5}$$

With this value for α, we can now estimate the noise power contribution from a sinusoidal signal in terms of a uniformly distributed signal, which enables us to estimate the ENOB of a sinusoidal distribution and compare results with standard ADC test measurements.

APPENDIX B

Full Yield Approximation – CDF of *X* Derivation

As shown in (3.34), the probability of maintaining some minimal ENOB can be expressed as the probability that *X* remains within some bound determined by the desired ENOB. Therefore, we can estimate the ENOB yield using the cumulative distribution function (CDF) of *X*. In what follows we will derive the CDF for *X* and compare the analytic expression for the ENOB yield to simulation results.

Treating the η_i in (3.34) as independent random variables, we can express the PDF of *X* as the series convolution of the η_i PDFs, which are defined in (3.29). This expression for the PDF for *X* is shown in (B.1) – where Π* denotes the convolution operator.

$$f_X(x) = \prod\nolimits_{i=0}^{N-1}{}^{\star} f_{\mathrm{H}_i}(x) \tag{B.1}$$

To obtain the CDF for X, we must compute the series convolution shown in (B.1). Although we can efficiently compute this convolution by transforming each of the ηi PDFs into the s domain and multiplying their moment generating functions[31], the resulting s domain expression we obtain for X is poorly structured and lacks a clear procedure for inverting the transformation

[31] Moment generating functions are analogous to Laplace transformations, where convolutions become products in the *s* domain.

and recovering a PDF.[32] Furthermore, a direct numerical calculation of this series convolution is troublesome since the PDFs of η_i are singular at the origin, which leads to numerical instability. In lieu of these difficulties, a less straightforward approach is used to derive a computationally feasible expression for the PDF of X. [33]

The method we employ to derive the CDF of X is based on an s domain transformation over convolved pairs of η_i. First, we will analytically compute the convolution over particular pairs of η_i, and then perform an s domain transformation on these expressions to obtain the moment generating functions. Next, we multiply the resulting moment generating functions and obtain a form for the moment generating function of X which is invertible. Finally, we recover the PDF of X and integrate to obtain the CDF. Following this procedure, we obtain an analytic expression for the CDF of X amenable to numerical approximation using standard numerical integration techniques.

Letting η_{ij} represent the sum of η_i and η_j, we can express the PDF of η_{ij} as the convolution of the η_i and η_j PDFs as in (B.2) – where σ_i and σ_j are described in (B.1).

$$f_{\mathrm{H}_{ij}}(\eta) \cong \frac{1}{2\pi \cdot \sigma_i \cdot \sigma_j} \int_0^{\eta} \frac{1}{\sqrt{t(\eta - t)}} exp\left[\frac{-t}{2{\sigma_i}^2} + \frac{-(\eta - t)}{2{\sigma_j}^2}\right] dt \qquad \text{(B.2)}$$

Using the trigonometric substitution $t = \eta\cos^2\theta$, we reduce (B.2) into the alternative formulation shown by (B.3), which resembles the PDF of an exponential random variable. The integral expression given in (B.3) represents our final simplification for the PDF of η_{ij}.

[32] The moment generating functions for η_i each contain distinct branch points along the real axis.

[33] A simple closed form expression for the CDF of a linear combination of Chi-Squared variables does not exist, but possible alternative computational solutions to this problem can be found in [51] – [52], where [51] offers an algorithm for numerically inverting the moment generation function, and [52] presents a asymptotic expansion based on an infinite series of incomplete gamma integrals.

$$f_{\mathrm{H}_{ij}}(\eta) \cong \frac{1}{\pi \cdot \sigma_i \cdot \sigma_j} \int_0^{\frac{\pi}{2}} exp\left(-\lambda_{ij}\eta\right) d\theta$$
$$\lambda_{ij} = \frac{\cos^2\theta}{2 \cdot {\sigma_i}^2} + \frac{\sin^2\theta}{2 \cdot {\sigma_j}^2} \tag{B.3}$$

Next, we transform (B.3) into the *s* domain by calculating its moment generating function. Since the expression in (B.3) is absolutely convergent within the region of convergence for *s*, we can apply Fubini's Theorem and interchange of the order of integration during this transformation. The resulting moment generating function for η_{ij} is shown in (B.4).

$$M_{\mathrm{H}_{ij}}(s) \cong \frac{1}{\pi \cdot \sigma_i \cdot \sigma_j} \int_0^{\frac{\pi}{2}} \frac{d\theta}{\lambda_{ij} - s}$$
$$\lambda_{ij} = \frac{\cos^2\theta}{2 \cdot {\sigma_i}^2} + \frac{\sin^2\theta}{2 \cdot {\sigma_j}^2} \tag{B.4}$$

The expression given in (B.4) is the moment generating function of the convolution over the pairs η_i and η_j. From a practical perspective, however, the integrand of (B.4) is just the moment generating function of an exponential random variable. Since moment generating functions formed through products of exponential random variables are easily inverted using Cauchy's Residue Theorem,[34] we can generate an invertible representation for *X* in the *s* domain by multiplying the moment generating functions derived from convolved pairs of η_i.

[34] Computationally, inverting moment generating functions formed through products of exponential random variables is the same as performing an Inverse Laplace Transform on a transfer function with a polynomial denominator.

Assuming that the resolution of the ADC is an even number of bits, the number of terms summed in X is even, and we can reduce the series convolution described by (B.1) into a series product of moment generating functions of the form given by (B.4). We will address the case where X is the sum of an odd number of terms later. Multiplying each of these moment generating functions, we obtain (B.5) – where $M_X(x)$ is the moment generating function of X, i denotes the moment generating function derived from i-th convolved pair of η, σ_{2i-2} is the i-th even σ from (B.1) including σ_0, and σ_{2i-1} is i-th odd σ from (B.1).[35]

$$M_X(s) = \int_0^{\frac{\pi}{2}} \cdots \int_0^{\frac{\pi}{2}} \prod_{i=1}^{N/2} \left(\frac{A_i}{\lambda_i - s}\right) d\theta_1 \cdots d\theta_{N/2}$$
$$A_i = \frac{1}{\pi \cdot \sigma_{2i-2} \cdot \sigma_{2i-1}} \qquad \lambda_i = \frac{\cos^2 \theta_i}{2 \cdot {\sigma_{2i-2}}^2} + \frac{\sin^2 \theta_i}{2 \cdot {\sigma_{2i-1}}^2} \tag{B.5}$$

Using Cauchy's Residue Theorem to invert (B.4), we recover the PDF of X. Integrating this PDF, we obtain the CDF for X as given in (B.6) – where σ_{2i-2} is the i-th even σ from (B.1) including σ_0, σ_{2i-1} is i-th odd σ from (B.1).

$$F_X(x) = \int_0^x \int_0^{\frac{\pi}{2}} \cdots \int_0^{\frac{\pi}{2}} \sum_{i=1}^{N/2} B_i e^{-\lambda_i t} \, d\theta_1 \cdots d\theta_{N/2} \, dt$$
$$B_i = \left[(\lambda_i - s) \prod_{j=1}^{N/2} \left(\frac{A_j}{\lambda_j - s}\right) \right|_{s \to \lambda_i} \tag{B.6}$$
$$A_i = \frac{1}{\pi \cdot \sigma_{2i-2} \cdot \sigma_{2i-1}} \qquad \lambda_i = \frac{\cos^2 \theta_i}{2 \cdot {\sigma_{2i-2}}^2} + \frac{\sin^2 \theta_i}{2 \cdot {\sigma_{2i-1}}^2}$$

[35] As long as the sigma values are sequenced in order of their magnitudes, the poles of the integrand in (B.4) remain unique throughout each dimension of the integration and complications arising from repeated roots are avoided when applying Cauchy's Residue Theorem.

Equation (B.6) is the CDF for X when the ADC resolution is even. Since the integrand of this expression is finite across all dimensions of θ, and the limits of integration are well defined, (B.6) is numerically well behaved. Through (B.6), we can numerically estimate the CDF of X and subsequently the ENOB yield of even resolution ADCs.

When the resolution of the ADC is an odd number of bits, however, the number of terms summed in X is odd. If we imagine constructing an ADC with odd number of bits by adding one bit to an even resolution ADC, we can treat (B.4) as the moment generating function for the even contribution. Using Cauchy's Residue Theorem to invert (B.4) and then convolving the resulting PDF with the PDF for η_{N-1} to capture the extra bit, we obtain the PDF for X in the case of odd resolutions. Integrating this expression, we obtain the CDF of X as shown in (B.7) – where σ_{2i-2} is the i-th even σ from (B.1) including σ_0, σ_{2i-1} is i-th odd σ from (B.1), and σ_{N-1} denotes the unpaired value from (B.1).

$$
\begin{aligned}
F_X(x) &= \int_0^x \int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{2}} \cdots \int_0^{\frac{\pi}{2}} \sum_{i=1}^{N/2} C_i e^{-\omega_i t}\, d\theta_1 \cdots d\theta_{N/2}\, d\phi\, dt \\
B_i &= \left[(\lambda_i - s) \prod_{j=1}^{N/2} \left(\frac{A_j}{\lambda_j - s} \right) \right|_{s \to \lambda_i} \\
A_i &= \frac{1}{\pi \cdot \sigma_{2i-2} \cdot \sigma_{2i-1}} \qquad \lambda_i = \frac{\cos^2 \theta_i}{2 \cdot {\sigma_{2i-2}}^2} + \frac{\sin^2 \theta_i}{2 \cdot {\sigma_{2i-1}}^2} \\
C_i &= B_i \sin\phi \sqrt{\frac{2t}{\pi \cdot {\sigma_{N-1}}^2}} \qquad \omega_i = \lambda_i \sin\phi + \frac{\cos^2 \phi}{2 \cdot {\sigma_{N-1}}^2}
\end{aligned}
\tag{B.7}
$$

From a computational standpoint, the CDFs given by (B.6) and (B.7) are quite expensive, and the time required to numerically estimate the CDF for X with reasonable error becomes

impractically large as the dimension of the integrals becomes large. Nevertheless, the CDFs given by (B.6) and (B.7) along with probability relationship given by (3.34) allow us to estimate the ENOB yield of a SAR ADC more efficiently than circuit level Monte-Carlo simulations. We provide a more convenient approximation for the CDFs given in (3.35) and (3.36) in Section 3.5.

BIBLIOGRAPHY

Introduction

[1] R. Feynman, “Plenty of Room at the Bottom,” presented at American Physical Society Meeting, California Institute of Technology, Pasadena, CA, 1959.

[2] Isaac Newton, Letter to Robert Hooke, Feb. 15, 1676.

[3] Plato, The Republic, 514a–520a, 380 BCE.

[4] F. Dostoyevsky, “Notes from Underground,” Epoch, 1864.

[5] W. Kester, The Data Conversion Handbook. New York, NY: Newnes, 2005.

[6] B. Murmann, "ADC Performance Survey 1997-2014," [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html. (Older 2012 version).

[7] R. H. Walden, “Analog-to-digital converter technology comparison,” in *Proc. of GaAs IC Symp*., pp. 228–231, Oct., 1994.

[8] R. H. Walden, “Analog-to-digital converter survey and analysis,” *IEEE J. Selected Areas in Communications*, no. 4, pp. 539–550, Apr. 1999.

[9] R. Walden, “Analog-to-digital conversion in the early twenty-first century,” *Wiley Encyclopedia of Computer Science and Engineering*, pp. 126–138, Wiley, 2008.

[10] R. H. Walden, “Analog-to-digital converters and associated IC technologies,” in *Proc. Compound Semiconductor Integrated Circuits Symp.*, Monterey, Oct. 2008, pp. 1–2.

[11] W. C. Black Jr. and D. A. Hodges, “Time Interleaved Converter Arrays,” IEEE International Conference on Solid State Circuits, Feb 1980, pp. 14–15.

[12] W. C. Black Jr. and D. A. Hodges, “Time Interleaved Converter Arrays,” IEEE Journal of Solid State Circuits, Dec 1980, Volume 15, pp. 1022–1029.

SAR ADC

[13] Lee, C.C.; Flynn, M.P., "A 12b 50MS/s 3.5mW SAR assisted 2-stage pipeline ADC," VLSI Circuits (VLSIC), 2010 IEEE Symposium on, vol., no., pp.239,240, 16-18 June 2010

[14] M. Yip and A. P. Chandrakasan, “A resolution-reconfigurable 5-to-10b 0.4-to-1V power scalable SAR ADC,” ISSCC Dig. Tech. Papers, pp. 190-192, 2011.

[15] Y. Zhu, et al., "A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS," IEEE Journal of Solid-State Circuits, vol.45, no.6, pp.1111-1121, June 2010.

[16] P. Nuzzo, et al., "Noise analysis of regenerative comparators for reconfigurable ADC architectures," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.55, no.6, pp.1441-1454, July 2008.

Capacitor Mismatch

[17] B. Murmann, "A/D converter trends: Power dissipation, scaling and digitally assisted architectures," IEEE Conf. Custom Integrated Circuit, pp.105-112, 21-24 Sept. 2008.

[18] Z. Zheng, U.-K. Moon, J. Steensgaard, B. Wang, and G. Temes, "Capacitor mismatch error cancellation technique for a successive approximation A/D converter," IEEE Int. Symp. Circuits Syst., vol.2, pp.326-329, Jul. 1999.

[19] B. Gregoire and U.-K. Moon, "Reducing the effects of component mismatch by using relative size information," IEEE Int. Symp. Circuits Syst., pp.512-515, May 2008.

[20] M. Saberi, R. Lotfi, K. Mafinezhad, W.A. Serdijn, "Analysis of Power Consumption and Linearity in Capacitive Digital-to-Analog Converters Used in Successive Approximation ADCs," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 8, pp. 1736-1748, Aug. 2011.

[21] J. McNeill, K. Chan, M. Coln, C. David, C. Brenneman, "All-Digital Background Calibration of a Successive Approximation ADC Using the 'Split ADC' Architecture," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 58, no. 10, pp. 2355-2365, Oct. 2011.

[22] S.-C. Lee and Y. Chiu, "Digital Calibration of Capacitor Mismatch in Sigma-Delta Modulators," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.58, no.4, pp.690-698, April 2011.

[23] S. Weaver, B. Hershberg, P. Kurahashi, D. Knierim, U.-K. Moon, "Stochastic Flash Analog-to-Digital Conversion," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.57, no.11, pp.2825-2833, Nov. 2010.

[24] Y.-Z. Lin, S.-J. Chang, Y.-T. Liu, C.-C. Liu, G.-Y. Huang, "An Asynchronous Binary-Search ADC Architecture With a Reduced Comparator Count," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.57, no.8, pp.1829-1837, Aug. 2010.

[25] K. R. Lakshmikumar, R. A. Hadaway, and M. A. Copeland, "Characterization and modeling of mismatch in MOS transistors for precision analog design," IEEE J. Solid-State Circuits, vol. SC-21, no. 6, pp. 1057–1066, Dec. 1986.

[26] A. Van den Bosch, M. Steyaert, and W. Sansen, "An accurate statistical yield model for CMOS current-steering D/A converters," IEEE Int. Symp. Circuits Syst., vol. 4, pp. 105–108, May 2000.

[27] Y. Cong and R. L. Geiger, "Formulation of INL and DNL yield estimation in current-steering D/A converters," IEEE Int. Symp. Circuits Syst., pp. 149–152, May 2002.

[28] M. Kosunen, J. Vankka, I. Teikari, and K. Halonen, “DNL and INLyield models for a current-steering D/A converter,” IEEE Int. Symp. Circuits Syst., May 2003, pp. 969–972, May 2003.

[29] G. I. Radulov, M. Heydenreich, R. W. van der Hofstad, J.A. Hegt, and A.H.M. van Roermund, “Brownian-Bridge-Based Statistical Analysis of the DAC INL Caused by Current Mismatch”, IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 2, pp. 146-150, Feb. 2007.

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

[31] M.J.M. Pelgrom, A.C.J. v. Rens, M. Vertregt, and M.B. Dijkstra, "A 25-Ms/s 8-bit CMOS A/D converter for embedded application," IEEE J. Solid-State Circuits, vol.29, no.8, pp.879-886, Aug. 1994.

[32] A. Moschitta and D. Petri, “Stochastic properties of quantization noise in memoryless converters affected integral nonlinearity,” IEEE Trans. Instrum. Meas., vol. 53, no 4, pp. 1179-1183, Aug. 2004.

[33] R. van de Plassche, CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters. Boston, MA: Kluwer Academic, 2003.

[34] B. Murmann. (2010) EE315B VLSI Conversion Circuits. [online]. Available:https:// ccnet.stanford.edu/cgi-bin/course.cgi?cc=ee315b&action=handout_download &handout_id=ID125185307612754.

[35] S. Kuboki, K. Kato, N. Miyakawa, and K. Matsubara, "Nonlinearity analysis of resistor string A/D converters," IEEE Trans. Circuits and Syst. vol.29, no.6, pp. 383- 390, Jun. 1982.

[36] Y. Cong and R.L Geiger, "Switching sequence optimization for gradient error compensation in thermometer-decoded DAC arrays," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process, vol.47, no.7, pp. 585-595, Jul. 2000.

[37] W. Kester, The Data Conversion Handbook. New York, NY: Newnes, 2005.

[38] J. A. Gubener, Probability and Random Processes for Electrical and Computer Engineers. New York: Cambridge University Press. 2006.

[55] J. A. Fredenburg and M. P. Flynn, “Statistical Analysis of ENOB and Yield in Binary Weighted ADCs and DACS with Random Element Mismatch,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol.59, no.7, pp.1396-1408, July 2012.

[56] R. B., Davies, “The distribution of a linear combination of chi-squared random variables,” J. Royal Statistical Society. Series C (Applied Statistics), vol. 29, pp 323-333, 1980.

[57] P. G. Moschopoulos and W. B. Canada, “The distribution functions of a linear combination of the chi-squares,” Computers and Mathematics with Applications, vol. 10, no. 4/5, pp 383-386, 1984.

Noise-Shaping SAR

[39] M. V. Elzakker, et al., "A 10-bit charge-redistribution ADC consuming 1.9μW at 1MS/s," JSSCC, Vol.45, No.5, May 2010 pp.1007-1015

[40] A. Shikata et al., “A 0.5V 1.1MS/sec 6.3fJ/conversion-step SAR-ADC with tri-Level Comparator in 40nm CMOS,” IEEE Symposium on VLSI Circuits, Jun. 2011, pp. 262-263

[41] C. C. Liu et al., **“**A 1V 11fJ/Conversion-Step 10bit 10MS/s Asynchronous SAR ADC in 0.18μm CMOS,” IEEE Symposium on VLSI Circuits, Jun. 2010, pp. 241-242.

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

[43] F. Kuttner, "A 1.2V 10b 20MSample/s non-binary successive approximation ADC in 0.13μm CMOS," IEEE International Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC., pp.176-177 vol.1, 2002

[44] J. J. Kang, and M. P. Flynn, "A 12b 11MS/s successive approximation ADC with two comparators in 0.13μm CMOS," Symposium on VLSI Circuits, 2009, pp.240-241, June 2009

[45] M. Miyahara, et al., "A low-noise self-calibrating dynamic comparator for high-speed ADCs," Proc. A-SSCC, pp. 269-272, Nov. 2008

[46] M. Yoshioka, et al., "A 10b 50MS/s 820μW SAR ADC with on-chip digital calibration," ISSCC Dig. Tech. Papers, pp. 384-385, 2010

[47] Y. Zhu, et al., "A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS," IEEE Journal of Solid-State Circuits, vol.45, no.6, pp.1111-1121, June 2010.

[48] Y. Chen, et al., "Split capacitor DAC mismatch calibration in successive approximation ADC," Custom Integrated Circuits Conference, 2009. CICC '09. IEEE, vol., no., pp.279-282, 13-16 Sept. 2009.

[49] M. Yip and A. P. Chandrakasan, "A resolution-reconfigurable 5-to-10b 0.4-to-1V power scalable SAR ADC," ISSCC Dig. Tech. Papers, pp. 190-192, 2011.

[50] L. Liu, et al., "A 95dB SNDR audio ΔΣ modulator in 65nm CMOS," Custom Integrated Circuits Conference (CICC), 2011 IEEE, vol., no., pp.1-4, 19-21 Sept. 2011.

[51] P. Nuzzo, et al., "Noise analysis of regenerative comparators for reconfigurable ADC architectures," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol.55, no.6, pp.1441-1454, July 2008.

[52] C. C. Liu, et al., "A 10b 100MS/s 1.13mW SAR ADC with binary-scaled error compensation," ISSCC Dig. Tech. Papers, pp. 386-387, 2010.

[58] K. Kim, J. Kim, and S.H. Cho, "Nth-order multi-Bit ΣΔ ADC using SAR quantizer," Electronic Letters, vol. 46, no.19, September 2010.

[59] J.A. Fredenburg, M.P Flynn, "A 90-MS/s 11-MHz-Bandwidth 62-dB SNDR Noise-Shaping SAR ADC," Solid-State Circuits, IEEE Journal of, vol.47, no.12, pp.2898,2904, Dec. 2012

[60] J. A. Fredenburg and M. P. Flynn, "A 90MS/s 11MHz Bandwidth 62dB SNDR Noise-Shaping SAR ADC," *ISSCC Dig. Tech. Papers*, 27.6, February, 2012.

Time-Interleaved MASH SAR

[53] B. Murmann, "ADC Performance Survey 1997-2014," [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html.

[54] J. A. Fredenburg and M. P. Flynn, "A 90-MS/s 11-MHz-Bandwidth 62-dB SNDR Noise-Shaping SAR ADC," Solid-State Circuits, IEEE Journal of, vol.47, no.12, pp.2898,2904, Dec. 2012