ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

Analysis of SAR ADC Quantization Nonidealities and Measurement of a 50Vpp Input Range 14Bit 250kS/s SAR ADC

M.Sc. THESIS

Çağrı GÜRLEYÜK

Department of Electronics and Communications Engineering

Electronics Engineering Programme

29 MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

Analysis of SAR ADC Quantization Nonidealities and Measurement of a 50Vpp Input Range 14Bit 250kS/s SAR ADC

M.Sc. THESIS

Çağrı GÜRLEYÜK
(504121360)

Department of Electronics and Communications Engineering

Electronics Engineering Programme

Thesis Advisor: Prof. Dr. Ali Toker

29 MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SAR ADC'lerde Kuantalama İşleminde Bozucu Etkilerin Analizi ve 50Vpp Girişli 14Bit 250kS/s SAR ADC Ölçümü

YÜKSEK LİSANS TEZİ

Çağrı GÜRLEYÜK
(504121360)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisiği Programı

Tez Danışmanı: Prof. Dr. Ali Toker

29 MAYIS 2015

Çağrı GÜRLEYÜK, a M.Sc. student of ITU Institute of Science and Technology 504121360 successfully defended the thesis entitled **"Analysis of SAR ADC Quantization Nonidealities and Measurement of a 50Vpp Input Range 14Bit 250kS/s SAR ADC"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Ali Toker** Istanbul Technical University	.....
Jury Members :	**Doç. Dr. Metin Yazgı** Istanbul Technical University	.....
	Prof. Dr. Osman Palamutçuoğulları Beykent University	.....
		.....

Date of Submission : **4 Mayıs 2015**
Date of Defense : **29 Mayıs 2015**

FOREWORD

I would like to thank my advisor, Prof. Dr. Ali Toker and my previous advisor, Assoc. Prof. Devrim Yılmaz Aksın for their contributions to this thesis. I would also like to thank the Duran Leblebici ITU VLSI Measurement and CAD Laboratories for supplying the equipment used in this thesis.

29 May 2015 Çağrı GÜRLEYÜK

TABLE OF CONTENTS

Page

ABBREVIATIONS

AC	**:** Alternating Current
ADC	**:** Analog to Digital Converter
CDAC	**:** Capacitive Digital to Analog Converter
DAC	**:** Digital to Analog Converter
DC	**:** Direct Current
ENOB	**:** Effective Number of Bits
FFT	**:** Fast Fourier Transform
HVSBS	**:** High Voltage Sampling Bootstrapped Switch
KSPS	**:** Kilosamples per second
LSB	**:** Least significant Bit
MSB	**:** Most Significant Bit
MSPS	**:** Megasamples per second
SAR	**:** Successive Approximation Register
SARADC	**:** Successive Approximation Register Analog to Digital Converter
SC	**:** Switched Capacitor
SFDR	**:** Spurious Free Dynamic Range
SNDR	**:** Signal to Noise and Distortion Ratio
SNR	**:** Signal to Noise Ratio
OPAMP	**:** Operational Amplifier
POR	**:** Power on Reset
VPP	**:** Volts peak-to-peak

LIST OF TABLES

Page

LIST OF FIGURES

Page

Analysis of SAR ADC Quantization Nonidealities and Measurement of a 50Vpp Input Range 14Bit 250kS/s SAR ADC

SUMMARY

Analog to digital converters are instruments that convert a physical quantity, a voltage or a current are the most common quantities in an electrical conversion scenario, to a digital value that represents the amplitude of the physical quantity with respect to a reference. Due to certain error factors, the digital value obtained after the conversion is not the perfect representation of the physical quantity. The design of an analog to digital converter integrated circuits requires the identification of these error factors and their optimization and minimization, and at the same time requiring the optimization of several other performance parameters such as power consumption, chip area and the number of external discrete components required. Reliable methods for characterizing and assessing the performance of analog to digital converters are required to verify and validate the design work.

This study focuses on the successive approximation register type of analog to digital converter, in an analysis, design and measurement scenario. The operation principle of the successive approximation register analog to digital converter is analyzed and the primary error factors, stemming from the quantization operation, that deteriorate the performance from an ideal analog to digital converter are identified. The analysis is carried over to the design of a novel 50Vpp input range, with 14bit resolution 250kS/s SAR ADC. The SAR ADC and its operation is presented, with the previously identified error sources are correlated to the operation of the various circuit elements that make up the circuit. Finally, the measurement setup for the SAR ADC is presented. With the measurement setup, the SAR ADC is characterized and its performance parameters are extracted.

SAR ADC'lerde Kuantalama İşleminde Bozucu Etkilerin Analizi ve 50Vpp Girişli 14Bit 250kS/s SAR ADC Ölçümü

ÖZET

Analog-dijital çeviriciler fiziksel değerlerin büyüklüklerini, elektronik bağlamında en yaygın olarak bir gerilim veya akım değerini, bir referansa göre temsil eden bir dijital değere çeviren elektronik enstrumanlardır. Bu çevrim işlemi, çeşitli hata faktörlerinden dolayı hiç bir zaman çevrilen fiziksel değerin mükemmel bir temsili olamamaktadır. Analog-dijital çevirici bütünleşmiş derelerin tasarımı bu hata faktörlerinin tespit edilmesini, optimizasyonunu ve küçültülmesini gerektirmektedir. Bunun yanı sıra, bu iyileştirme süreci çeviricinin güç harcaması, silikon üzerinde harcadığı alan ve devreyi çalıştırmak için gereken harici komponentlerin sayısı gibi diğer ikincil faktörlerin de minimizasyonunu gerekmektedir. Ek olarak, tasarlanan ve optimize edilen devrelerin karakterizasyonu ve performanslarının incelenmesi, tasarımın doğruluğunu garanti etmek açısından büyük bir önem taşımaktadır.

Bu çalışmada SAR (successive approximation register) tipi analog-dijital çeviriciler, bir analiz, tasarım ve ölçüm senaryosunda incelenmiştir. SAR, ya da diğer disiplinlerde de bilindiği ismi ile ikili arama algoritması arama algoritmalarında gerektirdiği düşük işlem sayısı sebebiyle en uygun bir noktaya sahiptir. SAR tipi analog-dijital çeviriciler elektronik dünyasında vakum tüplerinin yaygın olduğu 1950'li yıllarda dahi gerçekleşmeye başlanmış olmalarına rağmen, bütünleşmiş elektronik teknolojisinin gelişmesiyle beraber yaygınlık kazanmışlardır. SAR algoritmasını gerçekleyen bütünleşmiş lojik devreler 1970'li yıllarda ürün olarak sunulmaya başlanmış, ancak ilk tamamen bütünleşmiş SAR tipi analog-dijital çevirici, 1978 yılında piyasaya sürülmüştür. Uzun zamandır bilinen bir yapı olmasına rağmen, analog-dijital çevirici mimarileri arasında hız ve çözünürlük açısından kapladığı yer sebebiyle hala yaygın olarak kullanılmaktadır. Aynı zamanda, günümüzün düşük enerji tüketimi gerektiren mobil teknolojileri yaygınlaştıkça, analog-dijital çevirici mimarileri arasında en düşük örnek başına enerji tüketimleri sebebiyle SAR analog-dijital çeviriciler hala yaygın bir alan kaplamaktadır.

SAR analog-dijital çeviricilerin temel çalışma prensipleri incelenmiş ve analiz sırasında çevrim süresince görülen bozucu etkiler ve kuantalama işlemi sırasında çevrimde oluşan hata faktörleri tespit edilmiştir. Bu algoritmanın veri çeviriciler bağlamında gerçeklediği kuantalama işlemi matematiksel bağlamda tanımlanmıştır. Kuantalama işleminin matematiksel tanımı, işlem sonrası ortaya çıkan kuantalama hatası olarak isimlendirilen bozucu etkinin analizi için kullanılmıştır. Kuantalama işleminin sinyal üzerine getirdiği bozucu etkinin analizini kolaylaştırabilmek için istatistiksel bir analiz ile kuantalama gürültüsü tanımlanmıştır ve sinyal-gürültü oranı (signal to noise ratio, SNR) performans ölçütüne varılmıştır. Bu analiz, MATLAB ortamında kurulan bir model ile desteklenmiş ve analiz sonuçlarının nümerik

benzetimler ile tutarlılığı doğrulanmıştır. Bir sonraki adımda SAR analog-dijital çeviricilerin kuantalama işlemini gerçekleyen ikili kapasitif dijital-analog çeviricinin matematiksel analizi yapılmıştır. Analizin amacı, daha sonra tanımlanan ve nonlineerlik gibi bozucu etkiler getiren üretim süreci bozulmalarını bu bağlamda inceleyebilmektir. Ciddi bir nonlineerlik kaynağı olarak üretim süreci sonucunda oluşan kapasitif elemanlar üzerindeki rastlantısal dağılım incelenmiştir. Bu rastlantısal davranış, SAR analog-dijital çevirici kuantalayacısı ile birleştirilip bir nümerik model oluşturulmuş ve bu nümerik modelin devre benzetimi ile tutarlılığı gösterilmiştir. Bu bölümde yapılan analizlerin ve modellemelerin bütünü, daha sonra gelecek ölçüm ve tasarım adımlarının temellendirilmesini sağlamıştır.

Ölçüm sırasında çeviricinin davranışını irdeleyebilmek için çeviricinin çalışma prensiplerinin incelenmesi bir ölçüm stratejisi oluşturulması ve tasarımın bir sonraki iterasyonu için bir yön çizilmesi açısından kritik önem taşımaktadır. Ölçüm için 50Vpp giriş aralığı bulundan 14bit çözünürlüklü ve 250kS/s örnekleme hızlı bir SAR analog-dijital çevirici ele alınmıştır. Bu çeviricinin iki adımlı algoritmik yapısı bir 8bit ikili kapasitif dijital-analog çevirici kuantalayıcıyı iki ayrı çevrim adımında kullanmak üzerine kurulmuştur. Bu yapı düşük alan kullanımı ile beraber yüksek çözünürlük elde edilmesini sağlamaktadır. Yapı, ilk çevrimden oluşan kuantalama hatasını yükseltip tekrar kuantalama işleminden geçirmektedir. İki çevrim işlemi sonucunda elde edilen iki 8 bitlik değer, aradaki 64 kazanç sebebiyle toplam 14bit çözünürlüklü bir çevrim ile sonuçlanmaktadır. Çeviriciyi oluşturan devre elemanlarından, yüksek gerilimli anahtarlama devresi, yükseltici ve komparatör ve 8bit kapasitif dijital-analog çevirici detaylı olarak incelenmiştir. Bu analiz sonucunda her devre elemanının nihai çevrim sonucunda elde edilen dijital değere nasıl bir bozucu etki getireceği irdelenmiş ve tespit edilen hata kaynakları SAR analog-dijital çevirici sistemini gerçekleyen elemanlar ile ilişkilendirilmiştir.

Ölçüm ve karakterizasyon için ele alınan SAR analog-dijital çeviricinin gereksinimlerine özel bir ölçüm düzeneği kurulmuştur. Ölçüm düzeneği kurulurken hedeflenen statik ve dinamik ölçümler olarak iki gruba ayrılmış ölçüm gruplarının gerçeklemesi hedeflenmiştir. Ölçüm düzeneğinde bir saat işareti kaynağı, dörtlü gerilim kaynağı, lojik analizör, çok yüksek çözünürlüklü bir giriş işaret kaynağı ve bütün cihazların senkronize çalışmasını sağlayan bir referans saat işaret kaynağı kullanılmıştır. Çeviriciye bozucu etkicileri en aza indirgeyecek bir ara yüz kurulabilmesi için bir baskı devre yapılmış ve ürettirilmiştir. Baskı devrenin tasarımında çeviriciye olacak parazitik kapasitif ve direnç etkilerini minimize edecek ve ölçüm düzeneğindeki aletler ile uygun çalışabilecek bir yapı kullanılmıştır.

Ölçüm sırasında alınan veriler iki kategoride incelenmiştir. Statik ölçümler analog-dijital çeviricinin kuantalama aralıklarını belirlemek ve kuantalayıcının doğrusallığı gibi performans karakteristiklerini çıkarmak amacıyla gerçekleştirilmiştir. Bu ölçümleri elde edebilmek için çeşitli yöntemler incelenmiştir. Bu yöntemler arasından histogram metodu düzenekte gerekecek elemanların azlığı ve giriş sinyal üreteci ile uyumluluğu sebebiyle tercih edilmiştir. Giriş genlik olasılık dağılımı bilinen bir sinyalin analog-dijital çeviricinin çıkışındaki genlik olasılık dağılımının incelenmesine dayanan bu metot için analiz yürütülmüştür. Bu yöntem kullanılarak analog-dijital çeviricinin DNL ve INL performans parametrelerine varmak mümkün olmuştur. İkinci ölçüm olan dinamik ölçümler ise devrenin dinamik sinyallere olan

cevabını incelemek amacıyla gerçekleştirilmiştir. Bu ölçümler sırasında Fourier dönüşümü için gerekli örnekleme aralıklarının belirlenmesi ve kullanılacak giriş işareti frekansı sonuçları optimize edecek şekilde belirlenmiştir. Çeviricinin girişine yüksek hızlı bir sinüs sinyal uygulanırken çıkışının Fourier dönüşümü alınarak içerilen harmonik ve gürültü güç seviyeleri çıkartılmıştır. Bu güç seviyeleri oranlanarak SNR, SFDR ve SNDR gibi analog-dijital çevirici performans parametreleri elde edilmiştir. Ölçüm sonuçlarında elde edilen grafikler sunulmuş ve ölçüm sonuçları yorumlanarak çeviricinin performansı hakkında nihai bir sonuca varılmıştır. Ele alınan analog-dijital çevirici tam olarak karakterize edilmiş ve performans parametreleri çıkartılmıştır.

Bu çalışmada SAR analog-dijital çeviriciler, bir analiz, tasarım ve ölçüm senaryosunda incelenmiştir. Yapılan ölçüm, üretilmiş bir çeviricinin karakterizasyonunun tamamlanmasını sağlamış ve bir sonra gelecek olan nesil için yeni tasarım hedefleri konulmasına imkan vermiştir. Ölçüm sırasında elde edilen sonuçların anlamlandırılması adına, ele alınan 50Vpp girişli, 250kS/s 14bit SAR analog-dijital çeviricinin çalışma prensipleri ve alt elemanlarının getirdiği bozucu etkilerin analizi ölçüm sonuçlarıyla ilgileşim göstermiştir. Kuantalayıcılar üzerine yapılan teorik analiz ve analizin modeller ile doğrulanması ise bütün senaryonun temellendirilmesi adına önemli bir yer taşımıştır.

1. INTRODUCTION

1.1 Historical Review

Analog to digital converters are instruments that convert a physical quantity, a voltage or a current are the most common quantities in an electrical conversion scenario, to a digital value that represents the amplitude of the physical quantity with respect to a reference. Due to certain nonidealities, the resulting digital value after the conversion operation is not the exact representation of the physical quantity. The design of an analog to digital converters requires identification of these nonidealities and their optimization and minimization, and at the same time requiring the optimization of several other parameters such as power consumption, chip area and the number of external discrete components required. To this end, reliable methods for characterizing and assessing the performance of analog to digital converters are required to verify and validate the design work.

There exists two primary limiters in analog to digital conversion, a continuous in time and amplitude physical quantity has to be converted into a discrete in time and amplitude signal. The physical quantity subject to analog to digital conversion is very often continuous in time, such as a current representing the amount of light a laser diode is receiving or a voltage representing the change of resistance of a metal alloy due to it bending under a certain amount of weight. The amplitude of the current or the voltage will change instantaneously as the physical quantity changes. However a converter will often require a certain amount of time where its input is held at the value to be converted, to successfully convert the amplitude into a digital number. Converters commonly employ a *sample and hold* circuit to sample this continuous in time value and hold it at the sampled value for the duration of the conversion; namely converting the continuous input signal to a discrete signal. While sampling a continuous signal of limited bandwidth with a high enough sampling frequency as

dictated by the Shannon-Nyquist theorem will retain all the information content of the signal for subsequent reconstruction; the limit on the frequency imposes a limit on the application of the specific converter. Similarly, while both of the physical quantities mentioned are continuous in value, for all practical purposes of the contents of this thesis, an analog to digital converter can only represent a digital value in a finite number of bits that imposes a limit the resolution of the converter. To achieve the representation by a number of finite bits, the analog to digital converter maps a range of continuous values into a discrete value.

The SAR algorithm, also known as feedback subtraction as it was historically known can be traced back to 1500s relating to the solution of a mathematical puzzle about determining the least number of weighting operations to weigh any number of pounds between 1lbs to 40lbs [2]. The solution to this puzzle was presented initially by the mathematician Tartaliga in 1556, where he proposed using the series of weights 1lbs, 2lbs, 4lbs, 8lbs, 16lbs and 32lbs. Figure 1.1 shows the algorithm in action. The first comparison of the 45lbs target weight is made to the 32lbs weight, and if the result is greater, the second weight of 16lbs is added to the comparison. In the second comparison, the weight is smaller, so 16lbs is rejected and replaced with 8lbs, and so on. The algorithm is the same used in modern SAR ADCs.

The first inception of a SAR ADC architecture was by J.C. Schelleng of Bell Telephone Laboratories in a patent filed in 1946. The vacuum tube implemented ADC lacked the elegance of its modern counterparts, however, it followed the basic SAR algorithm to make comparisons to known voltages and select the appropriate next comparison steps according to the results. The next incarnation of the SAR ADC, again from Bell Telephone Laboratories in a 1947 article showed a 5bit, 8kS/s ADC where a primitive sample and hold structure was implemented, and appropriate charge depending on the code and the reference were subtracted from the sampled charge [3].

While both these ADCs showed the basic operational principle of the SAR ADC, they did not use binary weighted DACs to generate an approximate of the input signal. The more traditional SAR algorithm was described by H.R. Keiser and B.D. Smith in 1953, using both binary weighted and nonlinear DAC arrays, similar to the ADC structures used today.

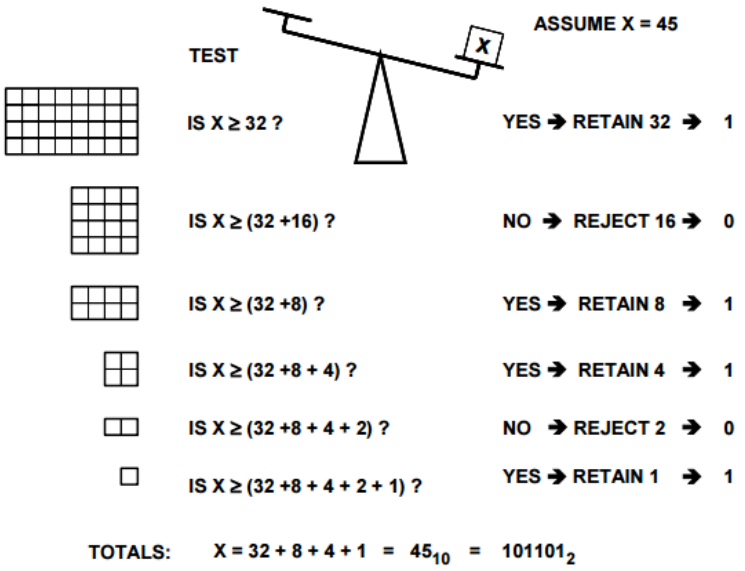


Figure 1.1: SAR algorithm weight analogy

Figure 1.2: DATRAC, 11bit 50kSPS SAR ADC

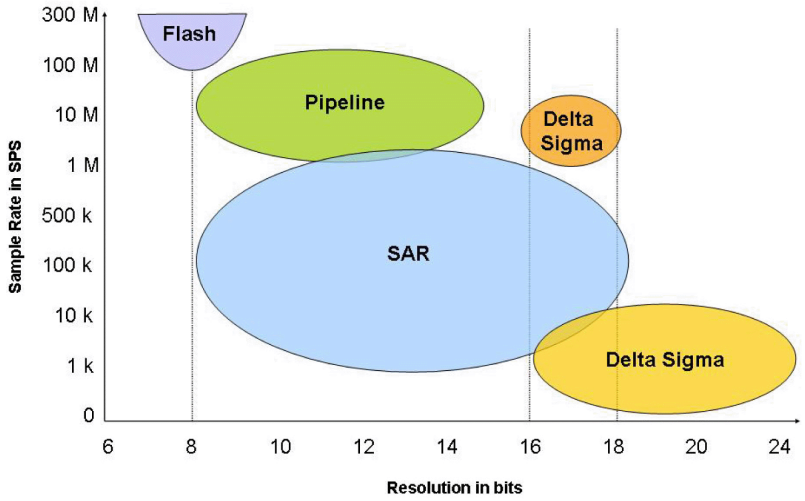


Figure 1.3: Analog to digital converter architectures grouped by resolution and sampling rate

The first commercially successful SAR ADC was DATRAC, seen in Figure 1.2, a 11bit 50kS/s SAR ADC. It was due to the revolutionary work by Bernard M. Gordon at EPSCO [4]. The DATRAC was a huge machine (almost 50cm wide, 70cm high and 40cm deep), dissipated several hundred watts and costed about 8000$.

The SAR algorithm was implemented by National Semiconductor and AMD, in te popular 2502/2503/2504 family of IC logic chips. These discrete chips would implement the SAR logic algorithm to facilitate the construction of all modular and hybrid SAR ADCs in th 1970s and 1980s.

With the advent of monolithic ICs and data converters. Analog Devices released AD574 a 12bit 3MS/s ADC in 1978. The use of laser trimming the thin film resistors to increase the accuracy and the linearity of the converter to the desired levels was highly successful and the AD574 became the state of the art of its time. The ADC is still in production and sold by Analog Devices as a general market product.

As CMOS technologies became popular and the feature size of the processes have dropped along with supply voltage scaling, SAR ADCs are almost always implemented in CMOS processes. The CMOS process allows for better control of the DAC sizes, that are considerably larger than any other component of a SAR ADC.

Analog-to-digital converters have a number of different architectures that optimize certain performance parameters in lieu of others to benefit a specific application

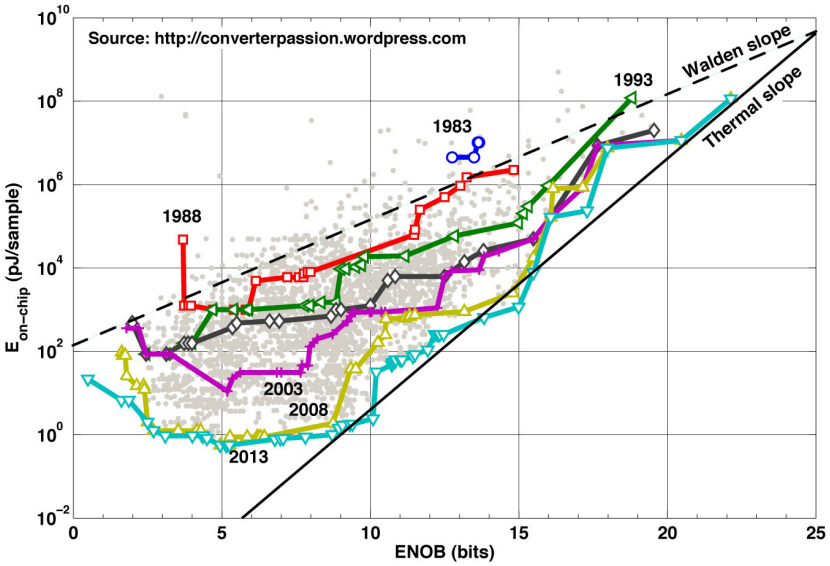


Figure 1.4: Energy per sample vs. Effective number of bits from 1985 to 2013 [1]

area. Figure 1.3 show the most commonly encountered ADC architectures by their resolution and sampling rate. The most common trade-off is between the sampling rate of the ADC and resolution, and this trade-off is what separates the most common architectures: flash ADCs provide the fastest sampling rates (hundreds of MSPS up to many GSPS) while providing the least resolution (3 to 5 bits) whereas delta-sigma ADCs provide the least sampling rates (10SPS to MSPS) while providing high resolution (up to 24 bits.) Successive approximation register, or SAR, analog to digital converters cover a versatile range among the common ADC topologies, covering a range from high resolution (8 bits), medium sampling rates (a few KSPS) to low resolution, fast sampling rates (100s of MSPS).

Another primary benefit of the SAR ADC architecture is that it has proven to be one of the most power efficient architectures available [5]. Power efficiency has been an important focus of optimization with the advent of mobile devices that rely on battery power, and the SAR algorithm proves to show the lowest energy consumption per sample obtained. Figure 1.4 shows the energy consumption trend for ADCs decreasing steadily over time, and optimization of ADCs utilizing the SAR algorithm has been a major contributer.

1.2 Purpose of the Thesis

As previously mentioned, the design of analog-to-digital converters relies on the identification and analysis of these nonidealities that cause the deviation from the perfect representation of the continuous in time and amplitude value in a finite number of digital bits. To this end methods of measurement and characterization must be developed, to be able to subsequently optimize the design enough to minimize the effect of these error factors. This thesis mainly focuses on the SAR (successive approximation register) architecture of the analog-to-digital converter, further focusing on the analysis of the nonidealities of the quantization stage in a SAR converter. The analysis is then supplemented by the measurement of a 50Vpp input range 14Bit 250kS/s ADC in a fully automated measurement environment where the characterization methodology is implemented and the effects of the nonidealities are observed in the experimental setup.

1.3 Organization of the Thesis

The organization of the thesis can be summarized as follows:

In Chapter 1, the thesis is introduced, a literature review and historical analysis of the SAR algorithm is conducted and the primary aim of the thesis is stated. In Chapter 2, the SAR algorithm is introduced and an analysis of the quantization operation is conducted. The operation of the quantizer in the SAR ADC is described and the primary error factors are identified. In Chapter 3, an overview of the operation of the 50Vpp input range 14bit 250kS/s SAR ADC to be measured is analyzed and error sources are correlated with the previous analysis. In Chapter 4, the methodology employed in characterizing and measuring the 50Vpp input range 14bit 250kS/s SAR ADC is presented. In Chapter 5, the thesis is concluded.

2. SAR ALGORITHM AND QUANTIZATION

2.1 Introduction

In this chapter the SAR algorithm will be defined and how it implements a quantizer will be described. Following an introduction of the general operation principle of the SAR algorithm, a mathematical definition for the quantization operation will be defined, and performance metrics for a non-ideal quantizer will be defined. In the course of the analysis, the results for a MATLAB model developed to verify the analytic definitions will be presented. The analysis will be further extended to a circuit level realization of a quantizer found primarily in SAR ADCs, the capacitive DAC. Finally, the common effects that cause nonlinearity degradation in SAR ADCs will be explained and the analysis will be verified by the presentation a statistical model that was developed which allows the simulation of a nonlinearity introducing effect aforementioned capacitive DAC.

2.2 General Operation Principle of the SAR ADC

The successive approximation ADC uses a method also known as to different disciplines as the binary search algorithm or the half search algorithm, all of which operate on the same basic principle. The position of an input in the search space is found by comparing the value of the input to the middle value of the search space iteratively. In every iteration the algorithm, depending on the result of the comparison the search space is halved. If the input voltage is greater than the midpoint, the new search space for the next iteration is the one enclosed by the high value and the middle value of the current search space. Similarly, if the input is smaller than the mid point, the new search space for the next iteration is the one enclosed by the middle value and the low value for the current search space. The halving operation yields a quite an efficient algorithm, where each iterative step increases the resolution of the conversion

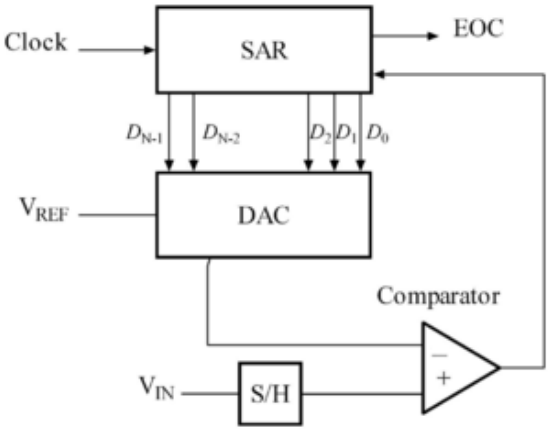


Figure 2.1: Block diagram of a SAR ADC

with a single additional bit and in the field of computer science, it yields the search result in $O(log(N))$ computational steps. It is worth noting that, this algorithm when implemented for a discrete input, discrete output case, can benefit from the equality operator as well to reduce the number of steps to reach the end of a search operation; however, this doesn't apply for the case in the implementation of an ADC, where the input value is continuous and the output value is discrete.

For the case of the SAR ADC, the binary search algorithm is implemented by digital logic that controls the input of a DAC to generate the middle voltage value of the search space, which is subsequently compared with the input value to yield if the input is on the greater or the lesser half of the search space. Figure 2.1 shows the block diagram of a SAR ADC.

The sample and hold block samples the input at a fraction of the clock signal of the system, implementing the time discretization function and holding the input of the comparator at the same value for the duration of the conversion. In the initial condition, the output of the SAR logic is at the middle code, 10000... which keeps the DAC is at the middle voltage of the voltage range of the ADC, defined by the reference voltage inputs, V_{ref_p} and V_{ref_n}, where the middle voltage is $V_{sar_0} = (V_{ref_p} - V_{ref_m})/2$. The comparator determines if the input voltage is greater or lower than this voltage, and feeds it back to the SAR logic block, which determines the next code. If the output is greater, the new search space should be defined by V_{ref_p} and V_{sar_0} which yield would

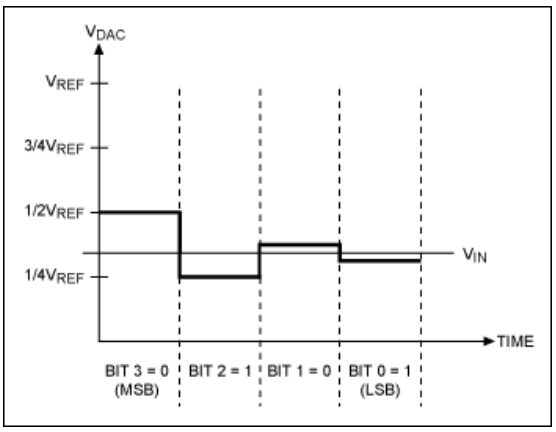


Figure 2.2: SAR ADC operation, converging to the input voltage

the following input for the comparator in the next iteration:

$$V_{sar_1}^{+} = \frac{V_{ref_p} - V_{ref_n}}{2} + \frac{V_{ref_p} - V_{ref_n}}{4} = \frac{3}{4}(V_{ref_p} - V_{ref_n}) \tag{2.1}$$

Similarly, if the output is smaller, the new search space would be defined by V_{sar_0} and V_{ref_n} which would yield the following input for the comparator in the next iteration:

$$V_{sar_1}^{-} = \frac{V_{ref_p} - V_{ref_n}}{2} - \frac{V_{ref_p} - V_{ref_n}}{4} = \frac{1}{4}(V_{ref_p} - V_{ref_n}) \tag{2.2}$$

This operation is repeated for $V_{sar_{\{0,1,2,...n\}}}$ where n is the number of bits of the SAR converter. Figure 2.2 shows the a SAR ADC converging to a input voltage in successive steps.

It should be noted that in this case, the SAR converter requires n clock cycles for every conversion, and thus the sampling clock (also known as conversion clock) is at a frequency $1/n$th of the system clock.

2.3 Quantization Analysis

An analog-to-digital converter is required to perform amplitude discretization of the analog input signal, since the digital output signal is required to be represented by a finite number of digits. This amplitude discretization operation is called *quantization*, and the functional block that performs this operation is called a *quantizer*.

Figure 2.3 shows the transfer function of a quantizer. The input range, $x_{max} - x_{min}$ is divided into a number of *quantization intervals*, with *quantization interval widths*, Δ_n, for the n-th interval, defined by the difference of its *quantization interval edges*,

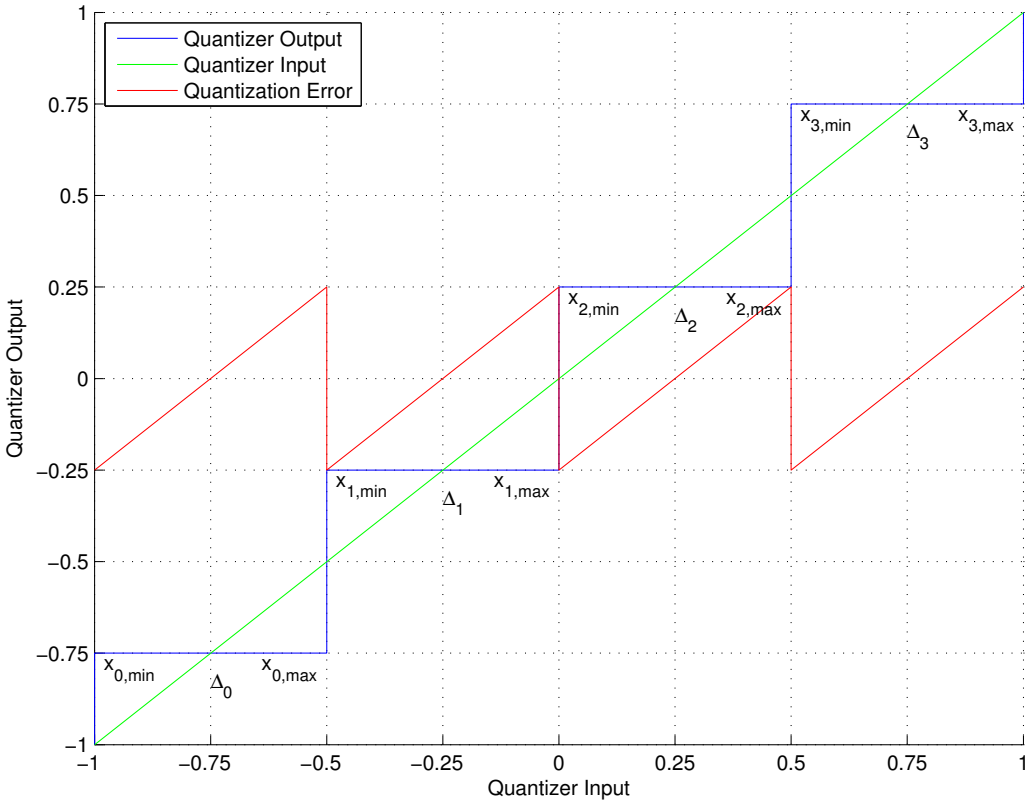


Figure 2.3: Quantizer transfer function and quantization error for an uniform quantizer

$x_{n,max}$ and $x_{n,min}$ where $\Delta_n = x_{n,max} - x_{n,min}$. The quantizer maps the continuous range corresponding to the quantization interval to a value that represents the quantization interval, usually chosen to be the midpoint of the quantization interval to facilitate analysis, $y_n = x_{n,mid} = x_{n,min} + \Delta_n/2$.

A *uniform quantizer*, or a *linear quantizer*, has the property that $\Delta_0 = \Delta_1 = ... = \Delta_M = \Delta = (x_{max,M} - x_{min,0})/M$ where the number of quantization intervals is M. Figure 2.3 shows the transfer function for an uniform quantizer and the quantization error, which will be defined in the following section.

Quantization is a many-to-one mapping operation, where a larger range (uncountably larger in the case of a continuous signal) is mapped onto a smaller range which renders the quantization operation an inherently nonlinear one.

To achieve Figure 2.3, a model for a 2-bit uniform quantizer was coded in the MATLAB environment and the model was driven with a linear ramp signal to arrive at the quantized values and the quantization error.

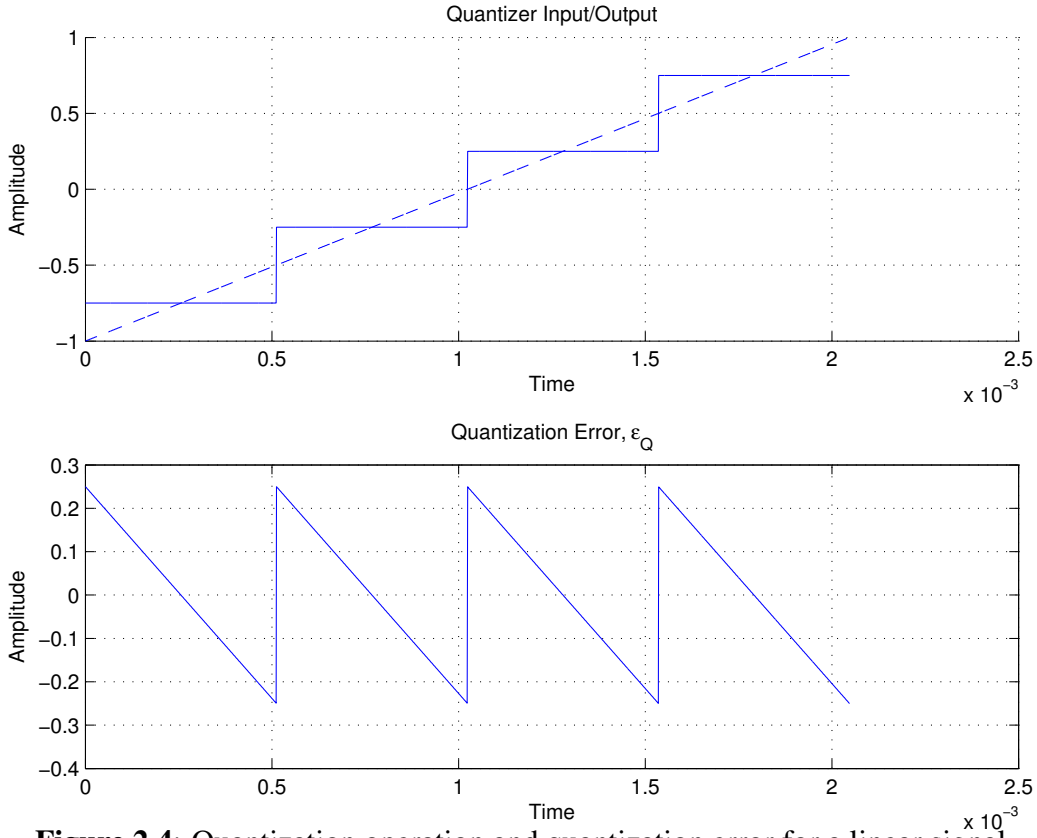


Figure 2.4: Quantization operation and quantization error for a linear signal

2.3.1 Quantization Error

The quantization operation introduces a difference between the original signal and the quantized resulting signal. The quantization operation can be summarized as in (**2.3**), where y is the quantized output signal, $Q(x)$ is the quantization operation, x is the original signal, and ε_q is the quantization error. Figure 2.4 shows the quantized output and the quantization error of a two bit (four level) quantization operation on a linear input signal. Figure 2.5 shows the quantized output and the quantization error of the same two bit quantization operation on a sinusoidal input signal. As can be observed from the figures, the quantization error is a highly input dependent signal, that is quite hard to approach analytically.

$$y = Q(x) = x + \varepsilon_q \tag{2.3}$$

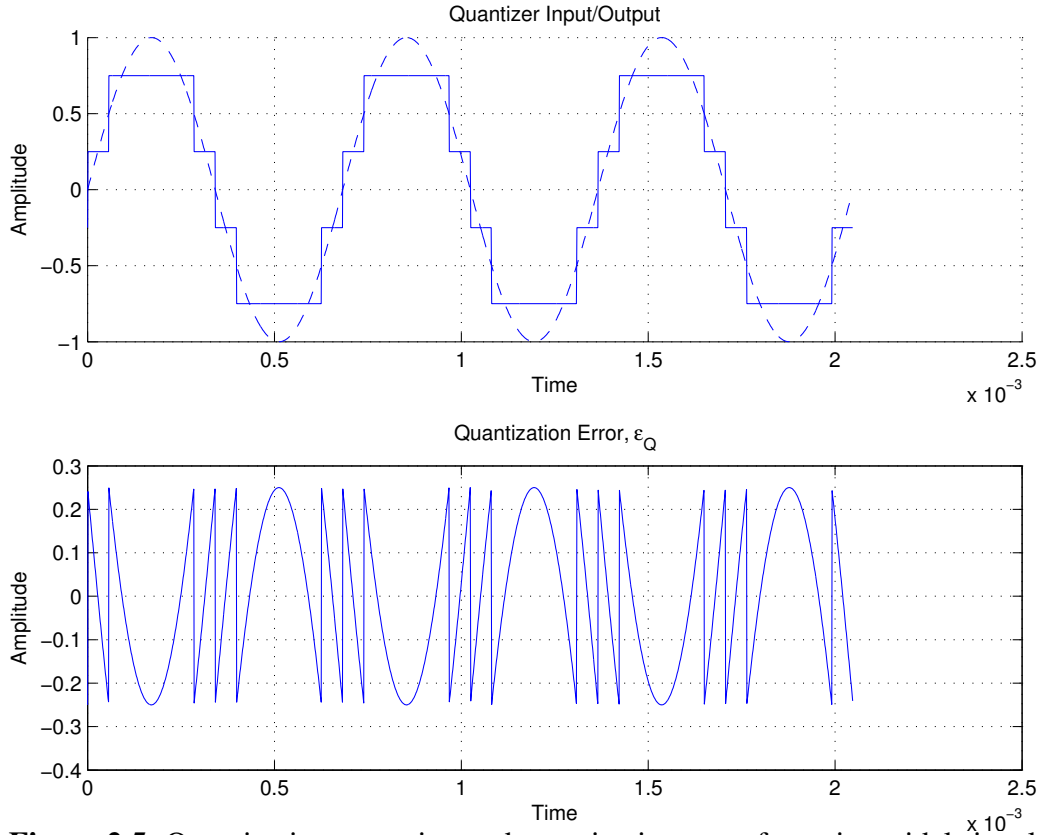


Figure 2.5: Quantization operation and quantization error for a sinusoidal signal

To achieve Figures 2.4 and 2.5 the quantizer model previously developed was driven with a linear ramp signal and a sinusoidal signal to show the nonlinear nature of the quantization error.

2.3.2 Quantization Noise

There exists a requirement to analyze and understand the effects of the quantization operation to facilitate an understanding of trade-offs and yield more robust designs. However, since the quantization transfer function is highly nonlinear and discontinuous, regular analytic methods fail to model it. A quantization error model can be reached by considering the quantization error to be probabilistic, and to be uniformly distributed for each quantization interval.

Figure 2.6 shows the power spectral density for a 114.257KHz sinusoidal signal, and the same signal after it has been through an 8 bit quantizer. The quantization error model aims to reach a power figure for the quantized and the original signal to quantify the effect of the quantization error.

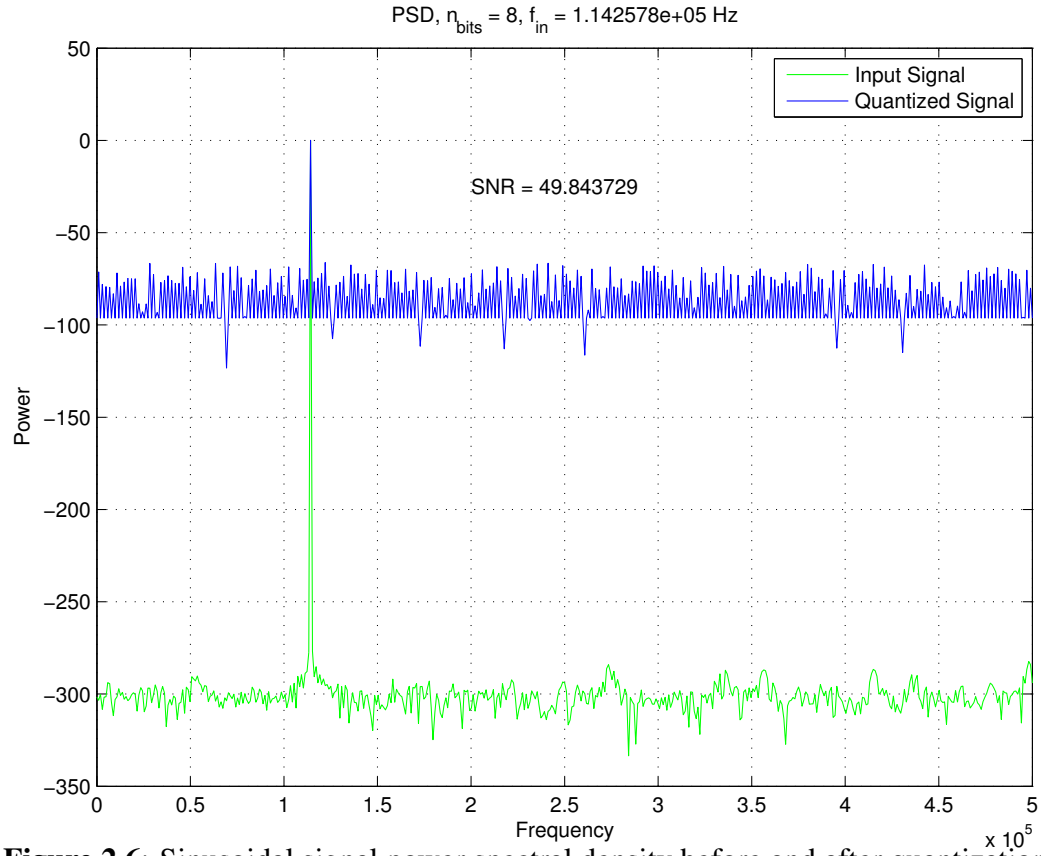


Figure 2.6: Sinusoidal signal power spectral density before and after quantization

To achieve Figure 2.6 the previously developed quantizer model was subjected to a sinusoidal signal, and the Fourier Transform was utilized to convert the signal into the frequency domain. After conversion to the frequency domain, the signal to noise ratio figure was reached by taking the ratio of the power in the signal frequency bin and the remaining frequency bins.

In (**2.4**), the power for the uniformly distributed quantization error is calculated. To arrive at a ratio that will define the effect of the quantization error relative to the signal, in (**2.5**), signal power is calculated for a sinusoidal signal and the amplitude of the sinusoidal signal is referred to the quantization interval and the number of bits, thus the full-scale input range of the quantizer.

$$P_Q = \int_{-\infty}^{\infty} \varepsilon_q^2 p(\varepsilon_q) d\varepsilon_q = \int_{-\Delta/2}^{\Delta/2} \frac{\varepsilon_q^2}{\Delta} d\varepsilon_q = \frac{\Delta^2}{12} \tag{2.4}$$

$$P_{sin} = \frac{1}{T}\int_0^T (\frac{A_{pp}}{2} sin(2\pi ft))^2 dt = \frac{A_{pp}^2}{8} = \frac{(\Delta 2^n)^2}{8} \tag{2.5}$$

The signal to quantization noise ratio (SNQR) can now be written, as in (**2.6**) by the ratio of the power of the input sinusoidal signal to the quantization noise power. In the decibels scale, the same figure can be written as seen in (**2.7**). The results reached are consistent with the SNR calculated for the quantized sinusoidal signal in 2.6.

$$SQNR = \frac{P_{sin}}{P_q} = \frac{12\Delta^2 2^{2n}}{8\Delta^2} = \frac{3}{2} \cdot 2^{2n} \tag{2.6}$$

$$SQNR_{dB} = 10 \cdot log\left(\frac{P_{sin}}{P_q}\right) = 10 \cdot log(\frac{3}{2} \cdot 2^{2n}) = 6.02 \cdot n + 1.78 \tag{2.7}$$

In Figure 2.7 a quantizer with n bits is simulated with a full-scale sinusoidal signal at its input and SNR values are extracted from the output quantized signal. To achieve Figure 2.7 the quantizer model previously developed was extended to be customizable with respect to n, the number of bits. n was swept and the signal to noise ratio for each case was calculated. The simulation Matlab code for generating the quantization noise by n-bits figure is appended to the thesis in Appendix A.1.

The calculated SNR from the analytical result obtained in (**2.7**) is highly consistent with the simulation results, showing that the model is a good approximation of the degrading effects of the quantization operation. This result important implication in defining the maximum signal to noise ratio achievable in a quantizer, and thus any data conversion system. Also, this definition of the signal to noise ratio versus the number of bits can be used to reach the ENOB (effective number of bits) figure for a measured ADC system that have degrading effects that worsen their SNR beyond the number of bits their quantizer has.

From this study of quantizers, it is reached that the quantizer puts a fundamental limit on the SNR in a data conversion system, and there is a minimum number of bits requirement to achieve a given SNR. Any second order effects will come as further degradations on this SNR number, and the circuit designer must take this into consideration while choosing a specific number of bits for the data conversion system.

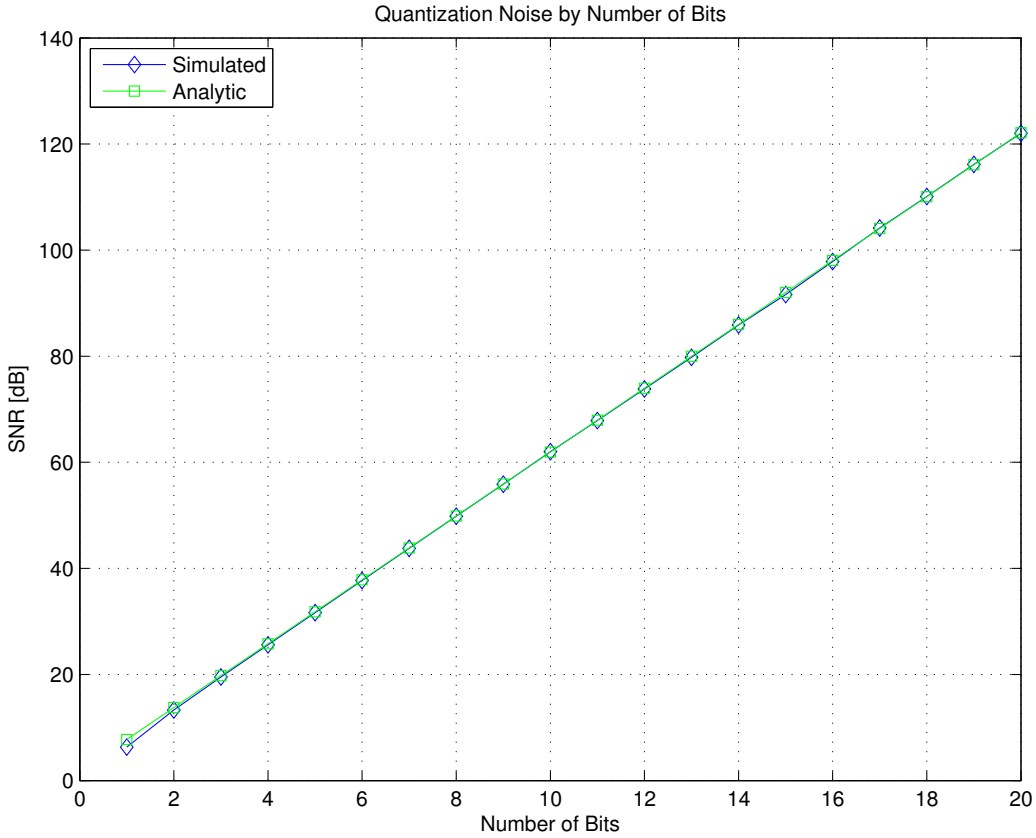


Figure 2.7: Signal to quantization ratio by number of bits of an uniform quantizer

2.3.3 Quantization Error and Quantization Noise in Nonuniform Quantizers

A nonuniform, or nonlinear, quantizer is defined by the property that the quantization intervals are not the same over the full range of the quantizer. The non-uniformity of the quantization intervals bring about several effects that invalidate the above analysis for quantization noise, since the resulting output signal usually has harmonic components. This characteristic of the nonuniform quantizer makes it nearly impossible to study analytically, and makes the designer reliant on simulations to see the signal degradation effects. However, it is also important to quantify the nonlinearity of quantizers, and the integral and differential nonlinearity figures will be defined in the following section to address that requirement.

2.3.3.1 Differential and Integral Nonlinearity

At this point, it is beneficial to define two measures used in characterization of quantizers, also frequently used in the context of data converters as a performance

metric: *differential nonlinearity*, *DNL*, and *integral nonlinearity*, *INL*. Differential and integral nonlinearity quantify the difference of a quantizer from a uniform quantizer.

As shown in **2.8**, DNL is the difference of the n-th quantization interval width, Δ_n, and the uniform quantizer interval width, $\Delta = (x_{max} - x_{min})/M$.

$$DNL_n = \Delta_n - \Delta \tag{2.8}$$

$$INL_n = \sum_{i=0}^{n} DNL_i \tag{2.9}$$

2.4 Quantizers in SAR ADCs

The quantizer in a SAR ADC is realized by the comparator, making the decision for each bit at every consecutive stage. The decision is made by comparing the input that has been sampled to the output of a DAC for the value of the respective code at that stage. Code edges and the quantization intervals in the SAR ADC is defined by the values that the DAC outputs for the respective codes. Thus, the analysis of the quantizer in the SAR ADC leads to the analysis of the DAC used to generate the voltages for the comparator.

In this section, firstly the binary weighted capacitive DAC will be analyzed. Next, the next step in the architecture, segmented or split capacitive DAC will be described. Finally, performance degrading secondary effects of these structures will be discussed.

2.4.1 Binary Weighted Capacitive DAC

An effective circuit for realizing the SAR ADC DAC is by utilizing the binary weighted capacitive DAC. This method relies on generating the voltages for the comparator by charging and discharging capacitors with known positive and negative reference voltages [6]. The circuit schematic for a binary weighted capacitive DAC is presented in Figure 2.8.

Consider the charge on node X to be, Q_x. The voltage at this node is V_x, the positive reference voltage is V_{ref_p} and the negative reference voltage is V_{ref_n}. The code input

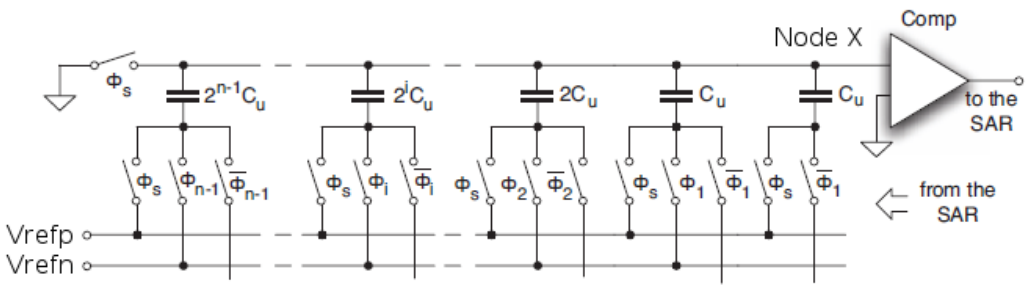


Figure 2.8: Schematic of a binary weighted capacitive DAC

for the DAC is $\vec{code}$ and its complement is $\overrightarrow{code}$. The charge on node X can be written as shown in (**2.11**).

$$\vec{bin} = \begin{bmatrix} 2^{n-1} \\ 2^{n-2} \\ \vdots \\ 2^0 \end{bmatrix} \tag{2.10}$$

$$Q_x = C_u(V_x - V_{ref_p})\vec{bin} \cdot \vec{code} + C_u(V_x - V_{ref_n})\vec{bin} \cdot \overrightarrow{code} \tag{2.11}$$

Consider the transition from $\vec{code}$ from a zero vector to a one LSB increase. The operation is shown in (**2.12**).

$$\vec{code} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad \overrightarrow{code} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \implies \tag{2.12}$$

$$Q_x = C_u(V_x^0 - V_{ref_n}) \sum_{i=0}^{n-1} 2^i$$

$$\vec{code} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \overrightarrow{code} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \implies \tag{2.13}$$

$$Q_x = C_u(V_x^1 - V_{ref_p}) + C_u(V_x^1 - V_{ref_n}) \sum_{i=1}^{n-1} 2^i$$

The charge at node X is constant during this operation. The two equations can be solved together to find the voltage at node X, which will correspond to the LSB voltage of the CDAC, as shown in (**2.15**).

$$\sum_{i=0}^{n-1} 2^i V_x^0 = \sum_{i=1}^{n-1} 2^i V_x^1 + (V_{ref_p} - V_{ref_n}) \tag{2.14}$$

$$V_{lsb} = V_x^1 - V_x^0 = \frac{V_{ref_p} - V_{ref_n}}{2^n - 1} \tag{2.15}$$

The fullscale range for the DAC wil be $(2^N - 1) * V_{lsb}$, and this range will be divided into $2^n - 1$ quantization intervals.

2.4.1.1 Charge Redistribution CDAC

The charge redistribution DAC is a slight variation on the binary weighted DAC, allowing more robust operation. In the charge redistribution DAC, the DAC array is used both for sampling the input voltage, and comparing the output to the negative reference voltage. According to the superposition principle, the operation of the DAC with respect to the code doesn't change, and the constant input of the comparator allows the common mode input of the comparator to be referenced to the ground. This allows for the analysis results reached in the binary weighted CDAC section to be utilized for charge redistribution DACs as well.

2.4.1.2 Segmented CDAC

The segmented CDAC is another variation on the binary weighted CDAC, where the DAC is split into two main banks, bridged by a series capacitor. The series capacitor causes the lower half of the array to show lower effective capacitance, therefore reducing the ratio requirement between the LSB unit and MSB unit of the DAC. This reduces the size of the capacitive array, and in conditions of ideal voltage division by the bridge capacitor, the analysis is the same as the binary weighted CDAC.

2.4.2 Linearity Degradation in Capacitive DACs

2.4.2.1 Linearity Degradation due to Process Mismatch

Process mismatch is the first contributor that degrades the linearity of the CDAC. The statistical characteristic of capacitor mismatch is given in (**2.16**), where M_a is the area parameter, and M_l and M_w are the width and length mismatch parameters for a given process [7].

$$3\sigma\left(\frac{\Delta C}{C}\right) = \sqrt{\frac{M_a}{W \cdot L} + \frac{M_l}{L^2} + \frac{M_w}{W^2}} \tag{2.16}$$

A binary CDAC was simulated with the above equations and the mismatch parameter both in MATLAB and SPICE, to see the effect of the unit capacitor geometry on the linearity characteristics. For the MATLAB simulation Equation (**2.16**) was implemented as the variation on the values of the unit capacitors with the use of Equation (**2.12**) were implemented to model a quantizer. For the SPICE simulation, the schematic in Figure 2.8 was implemented with ideal switches and process supplied MIM (Metal-Instulator-Metal) capacitors in a $0.35\mu m$ process. A linear signal was applied to the input of the quantizer in both cases, and INL and DNL values were extracted. For the variation parameters in the MATLAB simulation, the statistical variation parameters supplied by the process vendor was used. The code for the MATLAB simulation is in Appendix A.2. The results of the simulation are shown in Figure 2.9.

The correlation of the SPICE simulation with the analytical results verify the analysis. The size dependence of the binary sized capacitors show DNL spikes at code transitions that are respective to the switching from a (...01...11) code to a (...10...000). The assumption here was that each capacitor would be sized according to its value.

A method to counter this DNL spike and INL increasing behavior is to use thermometric coding [8]. When the value of the DAC input changes by a bit, in the binary CDAC, it might be required that a capacitor assigned to a larger bit value to be charged, and several small capacitor values decharged, or vice-versa. The

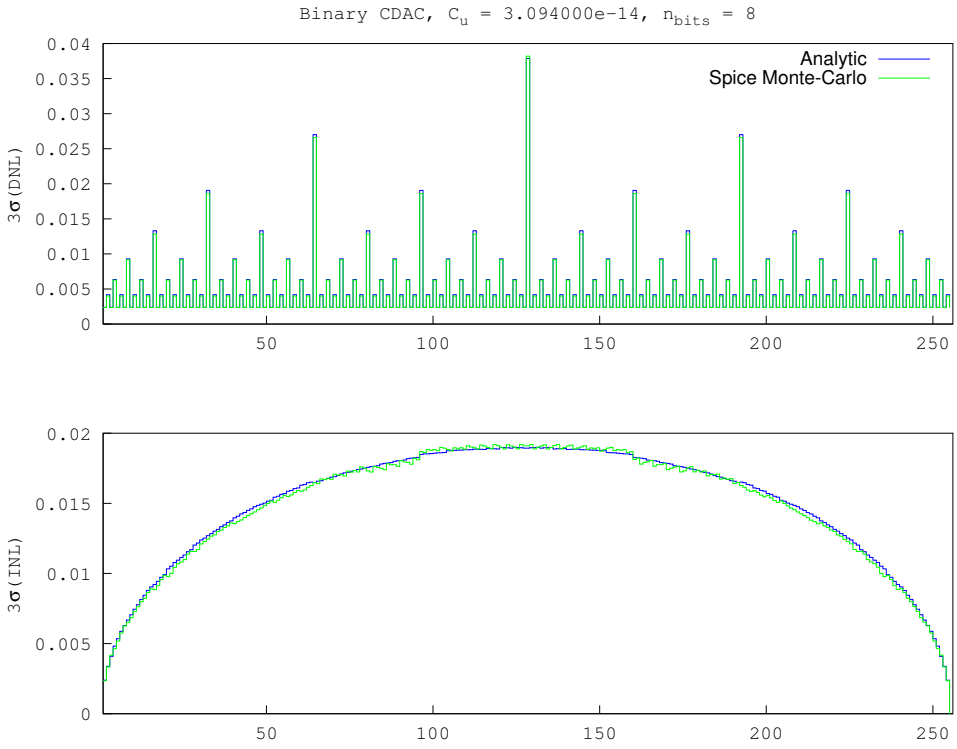


Figure 2.9: Process mismatch effect on CDAC linearity - binary switching

thermometric switching scheme only switches a number of unit cells equal to the code difference, meaning that the maximum error between two cells is dictated by the unit cell mismatch error.

The same simulation was run with a thermometric switching scheme, and the results are shown in Figure 2.10. It should be noted that while the thermometric switching scheme improves the DNL greatly, the integrated mismatch power still gives rise to the same INL.

The benefit of developing such an analysis was simulation time. A Monte-Carlo simulation conducted in a SPICE environment takes up to 10 hours using top end computing equipment, whereas the MATLAB simulation lasts only a few minutes on a regular home PC. This decrease in simulation time eases the analysis of statistical effects that degrade the linearity of quantizers greatly.

2.4.2.2 Linearity Degradation due to Process Gradients

Gradient errors are the process parameter changes on the wafer. Insulator thickness change in a given direction on the wafer would cause mismatch in the values of the

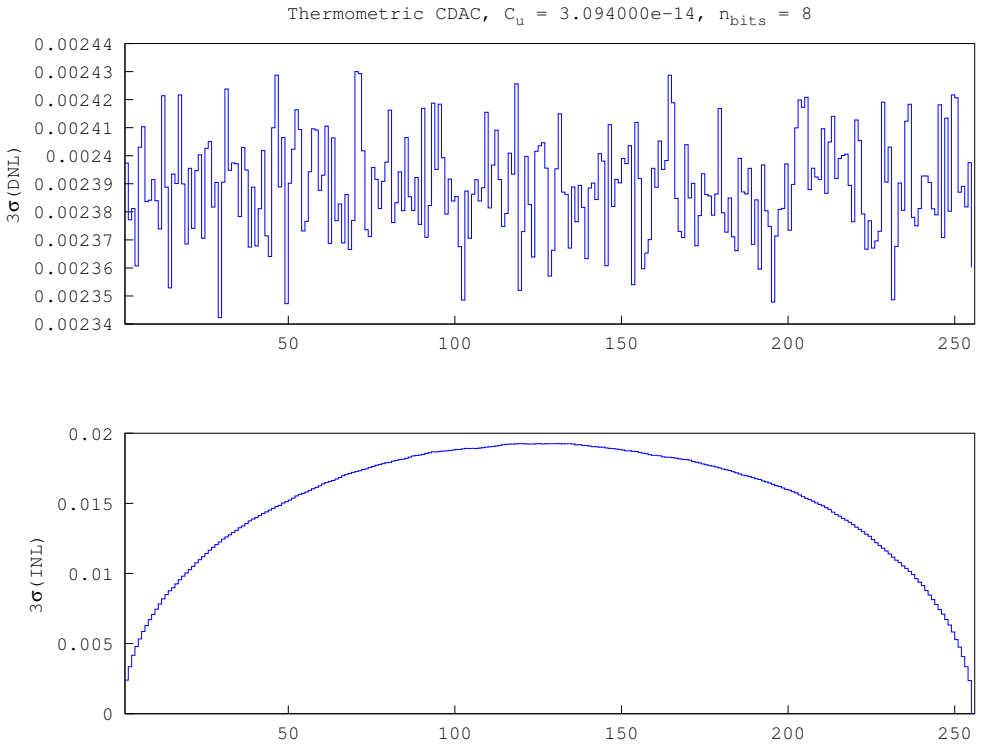


Figure 2.10: Process mismatch effect on CDAC linearity - thermometric switching

unit capacitors. A serious problem about gradient error is that there is not a proper way to simulate and foresee its effects. Careful layout consideration and placement of unit capacitors is required to overcome gradient effects. To overcome gradient effects, placement methods, such as Q^2 random walk have been proposed in the literature [9].

3. OPERATION PRINCIPLE OF THE MEASURED SAR ADC

3.1 Introduction

An understanding of the operation of the device to be measured is crucial in developing a measurement strategy and a following optimization scheme for the next, higher performance design iteration. In this chapter, the analysis conducted in the previous chapter will be related to the operation principle of the ADC that will be used in the measurement setup. The ADC that will be analyzed is a 50Vpp input range, 14Bit 250kS/s SAR ADC, developed in the ITU VLSI Laboratories previous to this thesis [10]. Firstly, architecture of the SAR ADC and its algorithmic structure will be explained, and the circuit components that are utilized in the operational stages will be presented. Next, the circuit components themselves will be presented in detail and primary sources of nonlinearity originating from these components will be highlighted and related to the analysis.

3.2 Architecture and General Operation Principle

In this section, the architecture and the general operation principle of the SAR ADC will be presented. The 50Vpp input range, 14Bit 250kS/s ADC is an extension of the SAR architecture, designed with the aim to reach a high resolution conversion by using an area efficient CDAC [10]. The ADC is designed to achieve the high resolution aim by employing the area efficient 8bit converter twice sequentially, where the quantization error of the first conversion stage is amplified, and fed back into the converter again in a subsequent 8bit conversion stage. This operation is analogous to a pipelined converter with two 8bit stages, with an interstage gain as set up by the amplification stage of operation. The block diagram of the stages used in the algorithmic SAR ADC is shown in Figure 3.1. The detailed diagram of the complete

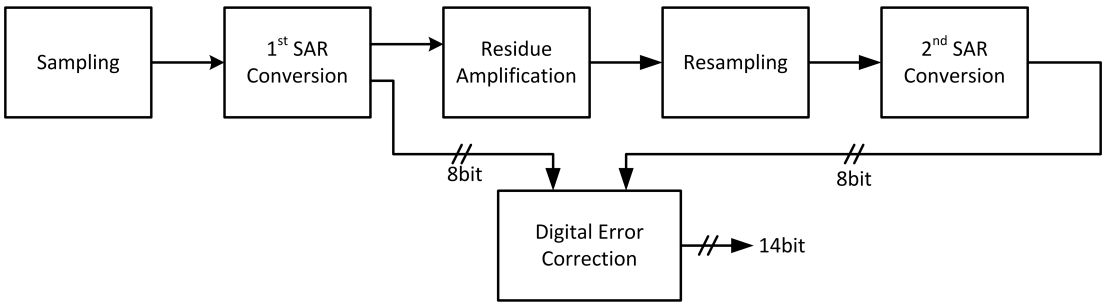


Figure 3.1: Block diagram of the SAR ADC

system including the circuit components and the timing for the stages are shown in Figure 3.2.

The operation consists of 5 phases: input sampling, 1^{st} 8bit SAR conversion, residue amplification, residue resampling and 2^{nd} 8bit SAR conversion. The high voltage bootstrapped switch (HVBS) is utilized to sample the high voltage input to MIM sampling capacitor C_S [11]. After the high voltage sampling, the sampled charge is redistributed on the C_{DAC}, which protects the internal circuitry from the high voltage input signal. Next, the system extracts the most significant 8 bits using the charge redistribution poly capacitor DAC array (C_{DAC}) in the 1^{st} SAR conversion. In the amplification phase, the remaining residual charge is amplified on the feedback capacitor C_F by an interstage gain of 64, dictated by linearity constraints. In the resampling phase, the amplified residue voltage is resampled on the C_{DAC}. Finally, another 8 bits are obtained in the 2^{nd} SAR conversion. The digital error correction block combines the results of the 1^{st} and the 2^{nd} conversion cycles to generate 14bit output code, dictated by the interstage gain, while the system samples the next input.

3.3 Circuit Sub-blocks

In this section, the circuit components that make up the 50Vpp input range, 14Bit 250kS/s ADC is presented. The circuit components presented are the high voltage bootstrapped switch, the 8bit capacitive DAC array and the amplifier/comparator block.

3.3.1 High Voltage Bootstrapped Switch

The circuit schematic of the high voltage bootstrapped switch is shown in Figure 3.3.

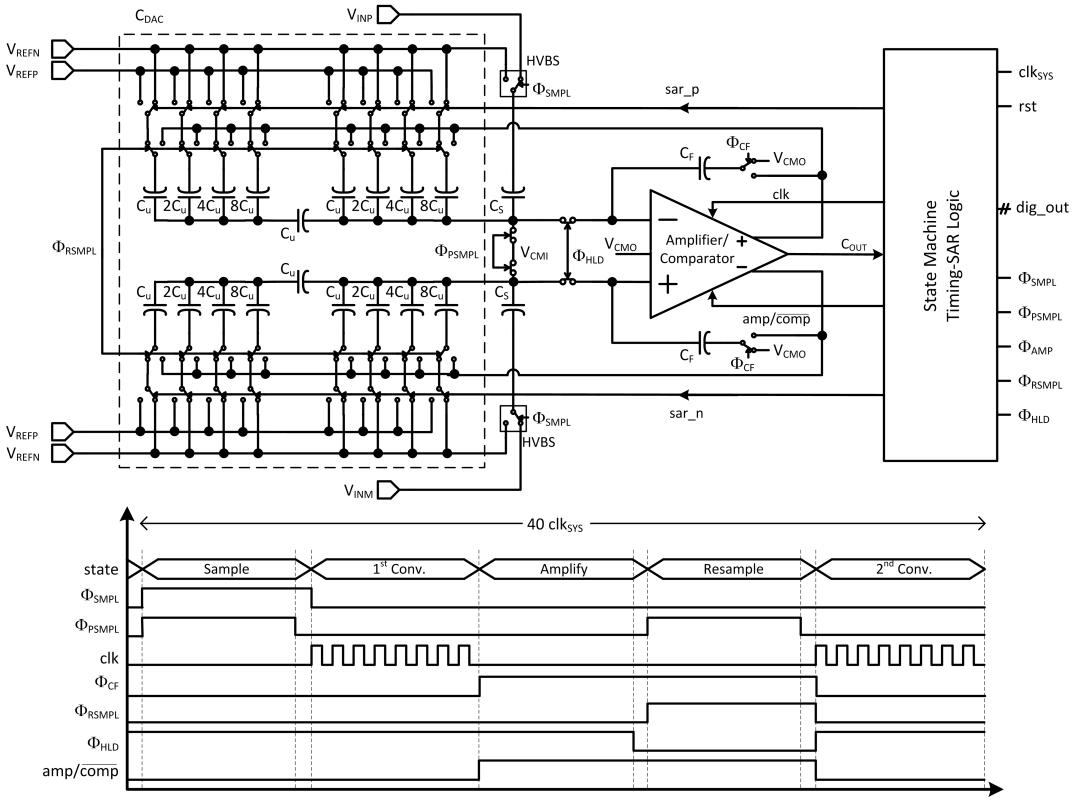
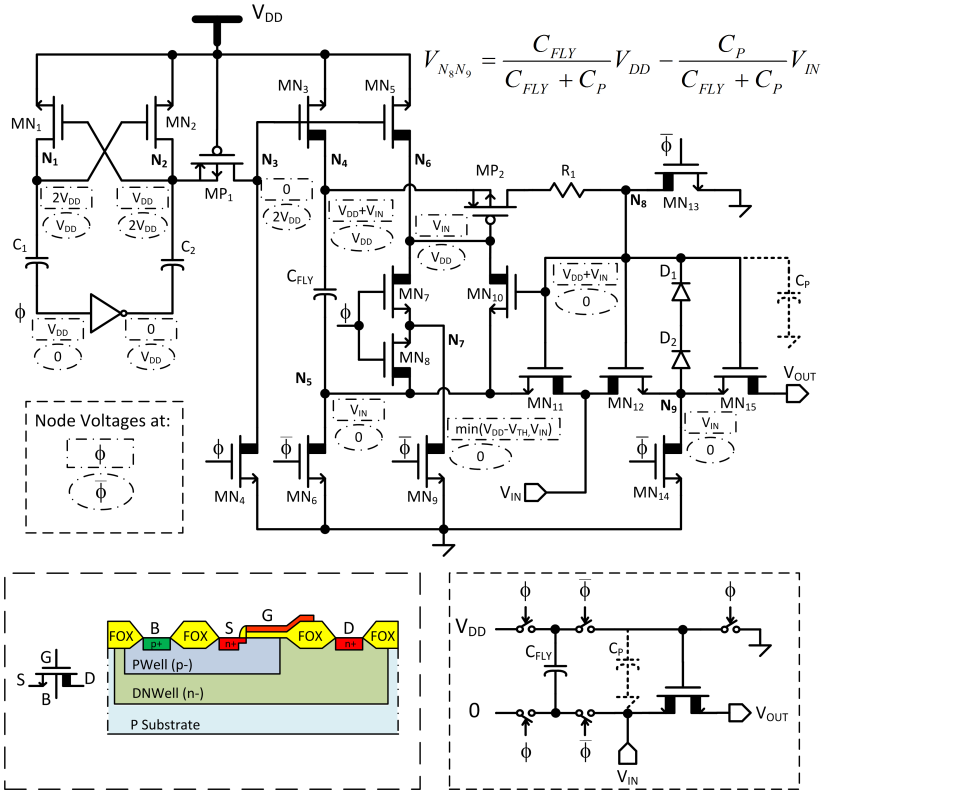


Figure 3.2: System diagram of the SAR ADC



$$V_{N_8N_9} = \frac{C_{FLY}}{C_{FLY}+C_P}V_{DD} - \frac{C_P}{C_{FLY}+C_P}V_{IN}$$

Figure 3.3: High voltage bootstrapped switch circuit schematic and the DNMOS device

The HVBS operates by charging the flying capacitor, C_{fly} to V_{DD} and connecting it between the gate source terminals of the switch transistors. The primary aim of the bootstrapping structure is to keep the V_{gs} of MN_{12} and MN_{15} transistors constant at V_{DD} while tracking the input signal. With a constant voltage of V_{DD} between the gate and source terminals of the MOSFETs, the devices are forced into linear operation with a constant on resistance, minimizing the nonlinear error in the input sampling stage. However, there are secondary effects that degrade the fully linear sampling operation. The primary degrading effect stems from parasitic capacitor at node N8, C_p, which causes charge sharing in the tracking phase resulting in V_{gs} modulation with V_{in}. It can be shown that the V_{gs} of the switching transistor is dependent on the input voltage V_{in} with a factor depending on the parasitic capacitor C_p and the flying capacitor C_{fly}, as presented in (**3.1**). To minimize the modulation and consequently the tracking non-linearity, the flying capacitor has to be sufficiently large compared to the parasitic capacitor at this node, so that the voltage division effect is minimal.

$$V_{N_8N_9} = V_{gs} = \frac{C_{fly}}{C_{fly}+C_p}V_{DD} + \frac{C_p}{C_{fly}+C_p}V_{in} \tag{3.1}$$

In addition to minimizing the nonlinear effects, the circuit has a high requirement of ensuring reliability, since the devices are subjected to high voltages during transient operation. R_1, on the switch gate charging path, protects the gate oxide by matching the time constants of the input to output path and the input to gate path, by slowing down the switch turn on speed.

The high voltage bootstrapped switch operates from a single 3.3V supply within its entire input signal range, covering the entire 50Vpp range without forward biasing any parasitic diodes. Figure 3.3 shows the cross-section of a DNMOS device. DNMOS devices do not introduce any additional process steps in a twin-well process, and thus, no additional costs in the process used [12]. The measured ADC reveals that the HVBS can reliably track and sample with at least 16bit linearity up to the V_{BD} of the isolated DNMOS, having 50V and 3.3V V_{BD}, drain breakdown voltage, and V_{BG}, gate breakdown voltage [11].

3.3.2 8Bit Segmented Capacitive DAC Array

The differential capacitor array is built with a 50V MIM sampling capacitor C_s in parallel with 4 by 4 segmented 8bit charge redistribution C_{dac}. The C_s capacitor is charged during the input sampling phase to the voltage value of the input signal. This voltage value results in a charge accumulation on the capacitor, defined by $Q_{C_s} = V_{in}/C_s$. This charge, in the following step, is transferred to the C_{dac}, where the charge redistribution causes the voltage to be attenuated to levels that will protect the internal circuitry that has a maximum voltage rating of 3.3V from the high voltage input.

As mentioned in the analysis, the C_{dac} segmented capacitor array has linearity degradation effects due to three primary sources. Process mismatches cause value mismatches between the capacitor cells independent of their placement dependent on their dimensions. The first step in countering random mismatch effects is to use unit cells, and assign a number of unit cells to every bit. With every cell being the same size, this essentially results in the same amount of mismatch in every cell. The second step is to use thermometric coding and switching. When the value of the DAC input changes by a bit, in the classical binary switching scheme, it might be required that a capacitor assigned to a larger bit value to be charged, and several small capacitor values decharged. As an example, switching between the binary values of 7 (0111) to 8 (1000) in an hypothetical 4 bit array, would cause an error depending on the mismatch of the capacitor assigned to the 2^3th bit and the sum of the 2^2th, 2^1th and 2^0th bit capacitors. This causes code dependent mismatch errors, which show as INL spurs at the respective codes. The thermometric switching scheme only switches a number of unit cells equal to the code difference, meaning that the maximum error between two cells is dictated by the unit cell mismatch error. As a continuation to the above example, switching between the binary values of 7 (0000000011111111 in thermometric code) to 8 (0000000111111111 in thermometric code) now results in the switching of a single unit. The final step in linearity optimization is deciding on the size of the unit cell. It is clear that choosing bigger dimensions for the unit cell will reduce the effect of mismatch between the cells relative to their size. However, the overall

capacitance of the capacitor array, C_{dac}, is dictated by the value of the unit cell which in turn dictates the value of C_s due to the fixed attenuation that is required to achieve the full scale swing in the input of the comparator block. This entails limitations on the bandwidth and has implications in the noise optimization of the ADC.

Gradient mismatches cause a different effect than the random degradation caused by process mismatches, as the capacitance value of unit cells change dependent on their physical placement. Since every bit is assigned a number of unit cells, this dependence of the unit cell values for each bit causes a fixed mismatch between the bit values. Orderly assignment of unit cells in a DAC will cause code dependent DNL, and degrade the linearity of the overall converter. The action taken to counter gradient mismatch effects was to place the unit cells in a Q^2 random walk pattern to minimize the effect of process gradients.

Mismatches arising from parasitic capacitors are usually of no big concern in regular capacitive DACs; however, for a segmented capacitive DAC, any parasitic capacitor at the LSB side of the segmentation capacitor results in a highly code dependent error. To counteract this effect, the parasitic capacitors arising from the metal connections should be minimized in the layout of the capacitive DAC. The thermometric driven C_{dac} unit capacitors are implemented with 4 identical poly capacitor cells whose top plates are completely enclosed/isolated by their bottom plates. This sandwiching action results in any parasitic capacitor from the top plate of the capacitor to be to the bottom plate of the capacitor, effectively increasing the capacitor value, but not causing any parasitics to other cells.

Combined, the parasitic isolation, thermometric drive and the placement strategy allows for a high linearity conversion without trimming, since the overall conversion linearity is limited by the first conversion nonlinearity in the two-step architecture.

3.3.3 Amplifier/Comparator

The circuit schematic for the amplifier/comparator block is presented in Figure 3.4.

The amplifier/comparator block satisfies two roles depending on the respective operational phase the SAR ADC is in. In the phases where the 8bit conversion is

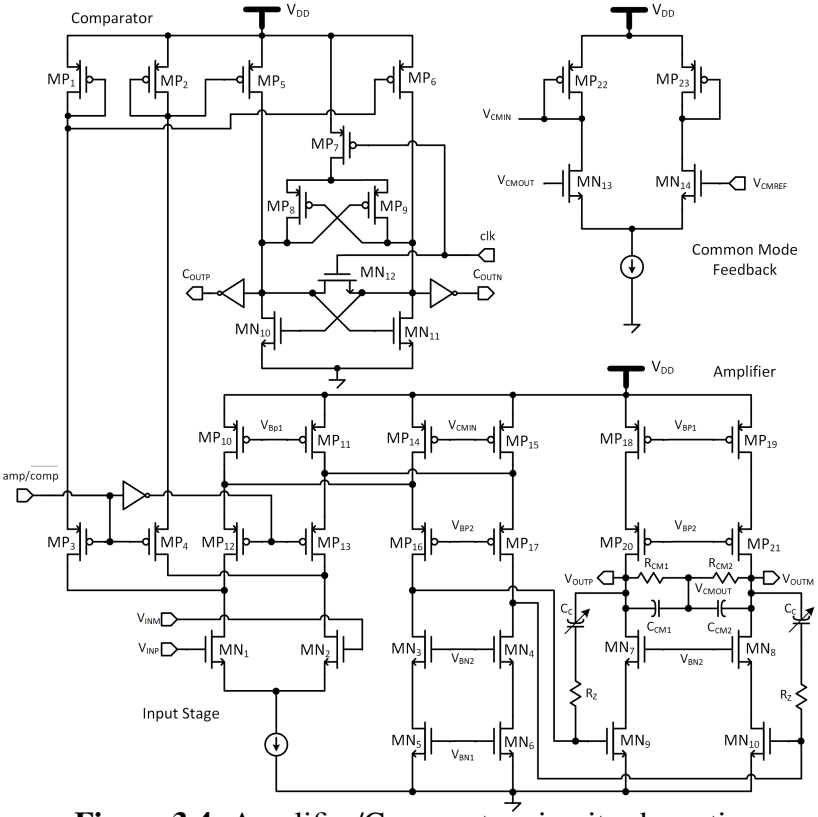


Figure 3.4: Amplifier/Comparator circuit schematic

taking place, the amplifier/comparator block operates as a comparator, outputting the result of the comparison for the respective bit. In this mode, the NMOS differential input stage loads the preamp/latch comparator stage, and the resulting digital output is fed into the SAR logic for the determination of the next bit. In the phase where the first conversion residue needs to be amplified, the amplifier/comparator block operates as an amplifier, the NMOS differential input stage is loaded by a folded cascode amplifier with continuous time common mode feedback and Miller compensation. The amplification factor is determined by the ratio of the feedback capacitor and the capacitive DAC capacitance, resulting in an overall gain of 64 as dictated by the system design. Following this phase, the amplifier/comparator keeps its operational mode as an amplifier, and resamples the final amplified residue value on the capacitive DAC, where the capacitive DAC is switched to be the load capacitor for the amplifier.

An important design aspect of the amplifier is its power consumption. In the amplification and resampling phases, the amplifier block is exposed to two drastically different load conditions. In order to assure that the amplifier is compensated, the case where the bigger load would present itself as the worst-case condition, and if

the circuit was biased for this mode of operation, excess power would be wasted to ensure the circuit is compensated in the amplification stage. The compensation and the bias current of the amplifier are switched with respect to its load condition (the amplification phase requires a smaller compensation capacitor) and configuration to improve settling time and power consumption.

4. CHARACTERIZATION AND MEASUREMENT RESULTS

4.1 Introduction

In this chapter the characterization effort and the measurement setup for the 50Vpp input range 14bit 250kS/s SAR ADC is presented. In the first section, the measurement setup and the devices used to conduct the measurement procedure are explained. In the second section, the two primary measurement methods for static and dynamic measurements are explained. In the final section, the measurement results are presented and the observed characteristics of the ADC are presented.

4.2 Measurement Setup and Printed Circuit Board

The SAR ADC presented unique challenges in its measurement, primarily due to the high input range requirement. The measurement setup designed were built by utilizing an input signal source, a clock source, a quad voltage supply and a logic analyzer, and a reference clock generator. Figure 4.1 shows the block diagram for the measurement setup. The reference clock generator synchronized the clock source and the input signal source, removing any frequency inconsistency between these two pieces of equipment to allow for coherent sampling that will be discussed below. The quad voltage supply was used to supply the 3.3V power rail to the circuit, in addition to supplying the reference voltages required for the SAR ADC operation. The clock source produced a 10MHz clock source, resulting in a 250kS/s sampling frequency for the ADC. The input source supplied the very high spectrally pure sinusoidal at 50Vpp. Finally, the logic analyzer was used to collect the digital output data from the ADC, which was then further analyzed in the MATLAB environment. The equipment used are presented in Table 4.1.

Table 4.1: Measurement equipment used for characterization

Use	Vendor	Model
Signal Source	Audio Precision	SYS-2722
Clock Source B	Rohde&Schwarz	SMA 100A
Reference Clock	Stanford Research	
Logic Analyzer	Agilent	16800
DC Voltage Source	Agilent	N6705

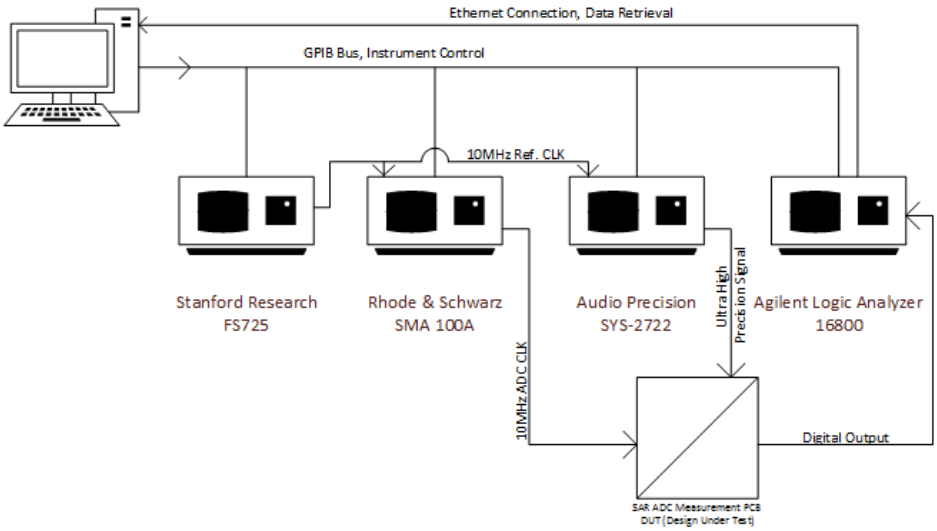


Figure 4.1: Block diagram for the measurement setup

The measurement environment was fully automated in the Labview environment. The Labview devices developed are appended to the thesis in Appendix A.5 to Appendix A.7.

The characterization of the SAR ADC in question is dependent on interfacing the measurement equipment to the chip, and to achieve that a printed circuit board has been designed and manufactured. The PCB is a two layer (double sided) FR4 substrate board with 1oz(35um) copper. Figure 4.2 shows the PCB designed for and used in the measurement setup Figure 4.3 shows the same PCB with the chip decapped and the die visible through the package.

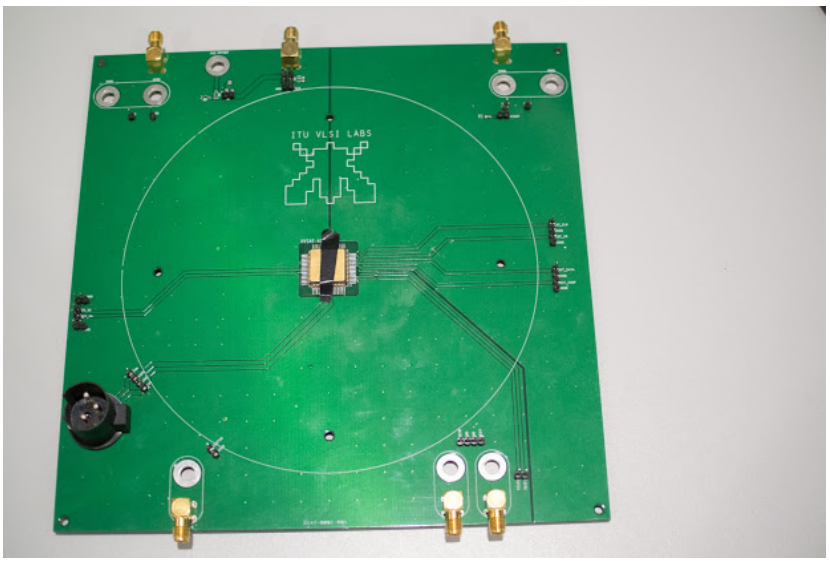


Figure 4.2: Printed circuit board used in the measurement setup

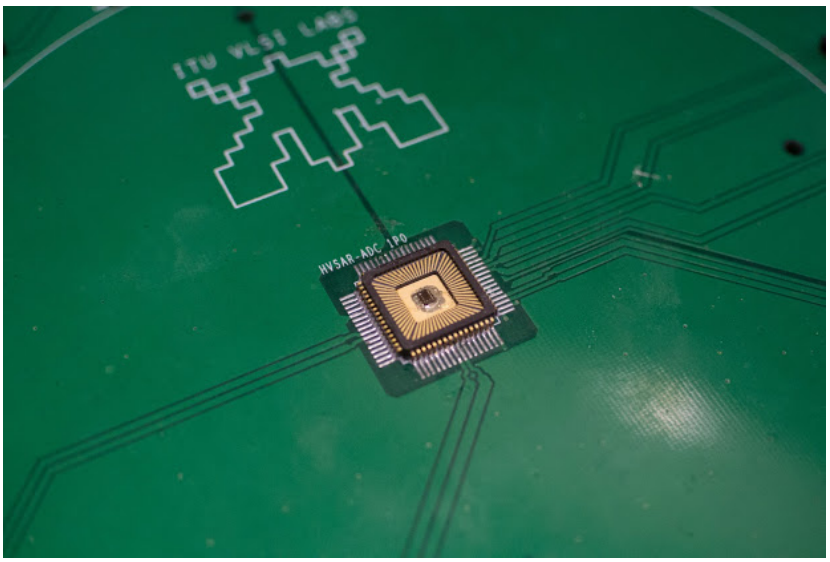


Figure 4.3: Printed circut board showing the uncapped SAR ADC in package

In the design of the board, the primary aim is to validate the chip in question and minimize any false errors caused by the board instead of the converter. To that end, several design recommendations were followed in designing the PCB. Ground pin connections for the analog and digital circuitry were separated at the board level, to separate the transient switching currents of the digital circuitry from the analog circuitry. The power connections, including the ground connections were sized to minimize voltage drops. SMA(SubMiniature version A) connectors and cables were used for analog pins, (differential input and output, voltage references) to connect

the equipment to the PCB to ensure lowest connection parasitics and highest signal integrity. The chip was given enough clearance so that the heat controlling device could sit on the board for temperature controlled measurements.

4.3 Measurement Methodology

4.3.1 Static Measurements

The aim of the static measurements of the ADC is to determine the quantization intervals, the analog range that gives rise to the specific digital codes. The test aims to determine the code edges, and from that, infer DNL and INL as the primary performance metrics.

Different methods are available when trying to detect the quantization intervals for an ADC. A very simplistic method is to find the input voltages where the probability of receiving either of the two codes is the same is to simply use a very precise source and find the probability distribution of codes for each input level. This method of measurement, as expected, is very time consuming for high precision converters, since multiple steps of measurement are required for every quantization interval. For a 16 bit converter, this would require 2^16 measurements, a unfeasible number. Another method is known as the servo loop method, where an integrator driven by a one bit DAC whose value is reversed when the digital output of the ADC crosses a code edge. While this method is faster than the first method offered, it requires forming a feedback loop and careful consideration of noise, since the input referred noise of the ADC might have a degrading effect on determining the precise location of the code edge.

A method superior to the previously mentioned measurement methods is the histogram method, and will be the method to be utilized in the ADC. This method relies on the statistical study of the input and output samples obtained from the ADC. The input of the ADC is a signal whose amplitude distribution, or probability density function, $p_i(x)$ is known beforehand. The output of the ADC has the probability distribution $p_o(x)$, related to the input distribution as in (**4.1**), where $V_{l,i}$ is the lower code edge, and $V_{u,i}$ is the upper code edge for code V_i.

$$p_o(x) = \int_{V_{l,i}}^{V_{u,i}} p_i(x)dx \tag{4.1}$$

The quantization interval for the code can be found as shown in (**4.2**).

$$\Delta_i = V_{u,i} - V_{l,i} \tag{4.2}$$

Assuming that the total number of samples, M, is large, the probability of the code can be approximated to the number of samples that give rise to the code V_i, as $p_o = M_i/M$. If the ideal code interval is defined as Δ, and the code interval for the code V_i is Δ_i, the DNL for the specific code can be calculated as shown in (**4.3**).

$$DNL(i) = \frac{\Delta_i - \Delta}{\Delta} = \frac{N \cdot M_i}{M} - 1 \tag{4.3}$$

When using the histogram method, there are different choices for the input signal to be used. One is to use a slow ramp signal, with an uniform distribution and a constant probability for every code. While this method would work ideally, any nonlinearity or noise effects that would alter the input signal and shifts its probability distribution from a uniform one would come up as errors in the DNL calculation. A better method is to use a sine wave; the probability distribution of a sine wave, $V = A \cdot sin(x) - V_o$ with amplitude A and offset V_o is equally well defined, as shown in (**4.4**).

$$p_i = \frac{1}{\pi\sqrt{A^2 - (V - V_o)^2}} \tag{4.4}$$

After obtaining the probability distribution, it can be normalized to the form of an uniform distribution by applying the reverse of the probability distribution shown in (**4.4**) [13]. From then on, the calculations to reach the DNL and INL figures are the same with the uniform distribution case.

There are multiple benefits of using the sine wave as the input to the histogram static testing method. The sine wave method can be used with less precise voltage sources, by using a bandpass filter on the frequency of interest to filter out any unwanted harmonic components. This effectively filters out any nonlinear effects that might degrade the input signal, and yields a pure sine wave for the input. The utilized Audio Precision AP

SYS-2722 signal source generates a sine-wave with THD at -110dB, which is sufficient for the measurement of the ADC with the expected 14Bit linearity.

The Matlab code for post-processing the static measurement results is appended to the thesis in Appendix A.3. The Labview environment for the static measurements are appended to the thesis in Appendix A.5 and Appendix A.6.

4.3.2 Dynamic Measurements

The method used in the dynamic measurements is simpler compared to the static measurement method. The input signal used is a sine wave, and the resulting output codes are captured. The Fast Fourier Transform of the resulting output is taken and specifications like the SNR, SNDR and SFDR are measured from the resulting power spectral density plot.

The critical element in measuring the performance parameters is to achieve as idealistic a Fourier transform as possible. This is made possible by utilizing coherent sampling. Coherent sampling allows for the removal of a windowing function at the output of the signal, and therefore removes any windowing artifacts. To utilize coherent sampling the frequency of the input sinusoidal, f_i is related to the sampling frequency of the ADC by (**4.5**), where f_s is the sampling frequency of the ADC, k is the number of periods to be captured and N_{fft} is the number of samples used in the FFT operation.

$$f_i = \frac{f_s \cdot k}{N_{fft}} \tag{4.5}$$

To achieve coherent sampling in the measurement setup, all the devices were synchronized to the 10MHz rubidium reference clock generator, so that precise frequency programming could be made to the signal source and the clock source to achieve the required frequency resolution.

After obtaining the power spectrum of the output signal, the fundamental component of the signal is extracted from the output, and its power is calculated. The resulting power spectrum is integrated to yield the total noise and harmonic distortion power, allowing the calculation of SNDR. The harmonics are then extracted and the power spectrum of the harmonics are calculated to allow the calculation of THD. Finally,

the noise power without the harmonic content is calculated to allow the calculation of SNR.

The Matlab code for postprocessing the dynamic measurement results is appended to the thesis in Appendix A.4. The Labview environment for the dynamic measurements are appended to the thesis in Appendix A.7.

4.4 Measurement Results

In this section, the measurement results from the setup described in the previous section is presented, and the obtained results are discussed.

4.4.1 DNL and INL

The measured DNL and INL plots for the ADC can be seen in 4.4. The figure shows that maximum DNL is $\pm$0.55LSB and maximum INL is $\pm$1.81LSB. As expected, the primary INL degradation stems from the split CDAC architecture, as the figure shows 16 distinct segments at every 256th code, resulting from the nonlinearity of the CDAC at the first conversion step. The source of this nonlinearity is the segmented architecture used, and the primary degrading effect is the parasitic capacitors at the segmented capacitors top and bottom plates as discussed previously.

4.4.2 SNDR and SFDR

Figure 4.5 shows the measured frequency spectrum of the ADC output for a 23kHz 50Vpp input signal on a 15V common-mode voltage at 250kS/s. The measured SFDR is 97.8dB and the measured SNR is 80.2dB. The measured SNR shows 13.02 effective number of bits, where the measured SFDR shows a 15.9 bit linear range. In all the measured samples, an SFDR figure better than 90dB were observed, without the need for any requirement for trimming, calibration or tuning.

Figure 4.6 shows the measured SNDR and SFDR figures for the ADC, with varying input amplitude values, with a sampling frequency of $f_s = 250kS/s$. The linear increase in both of these performance metrics show that the ADC is highly linear up to its fullscale range and linearity optimizations on the CDAC have been successful.

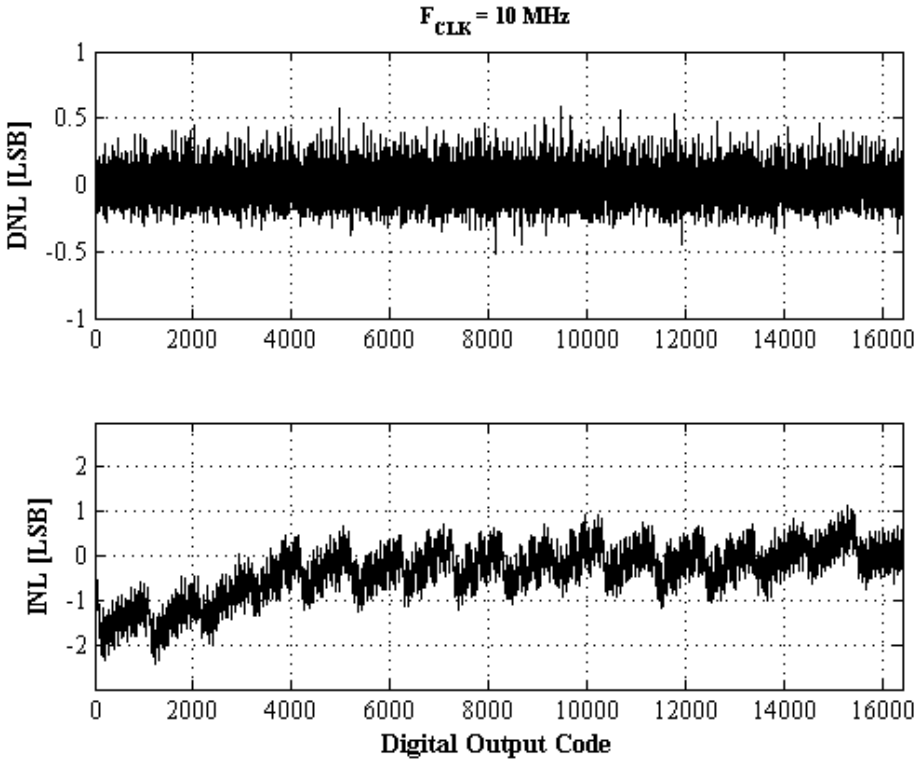


Figure 4.4: Differential and integrated nonlinearity measurement results

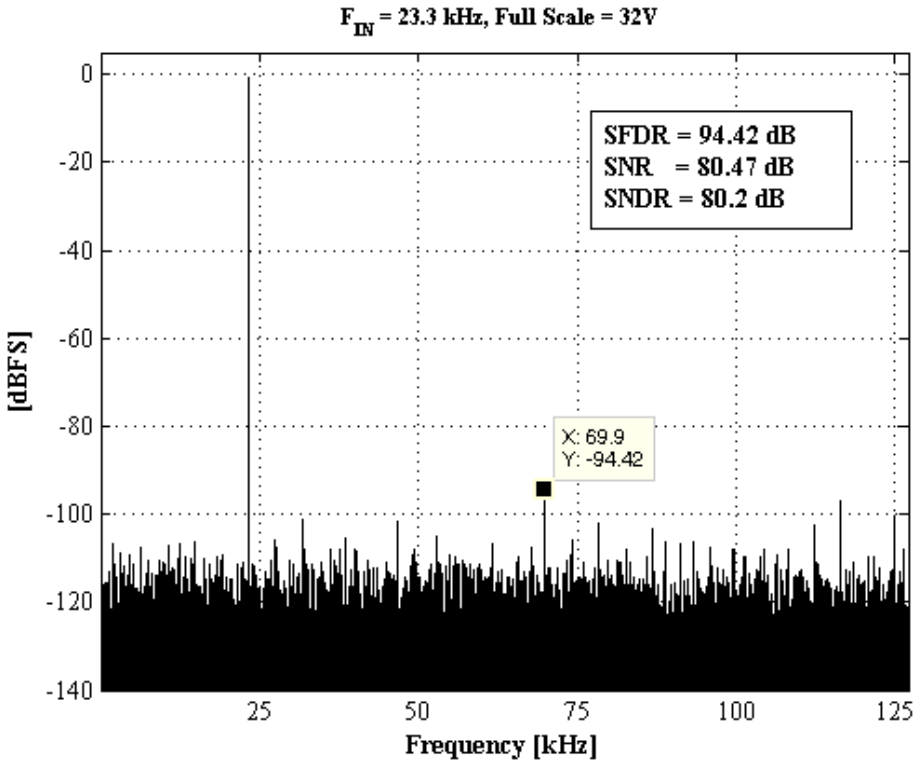


Figure 4.5: Power spectral density plot

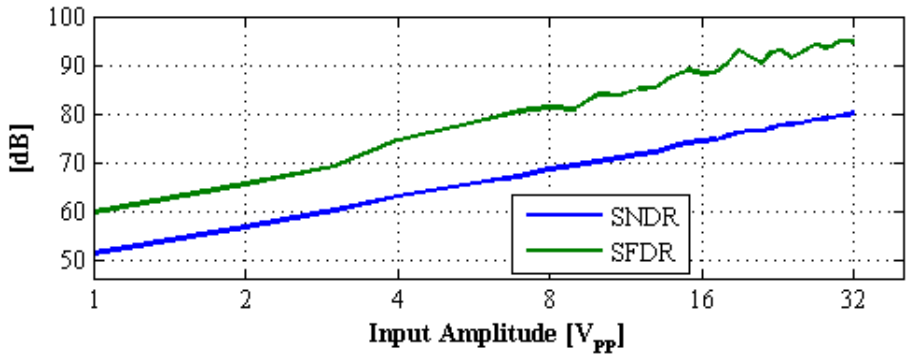


Figure 4.6: SNDR and SFDR versus input amplitude

Figure 4.7 shows the measured SNDR ad SFDR figures for the ADC, with increasing sampling frequencies. The ADC operates successfully up to 250kS/s, and starts showing SNDR and SFDR degradation at sampling frequencies beyond this value. This is attributed to the speed of the amplifier block. As the speed of the sampling frequency increases, the amplifier block does not have enough time to settle in the amplification and the resampling phases, and introduces nonlinear effects, dramatically reducing SFDR.

4.4.3 Noise and Zero-Input Code Distribution

To assess the noise characteristic of the ADC, the inputs were kept at a zero level (no sine wave applied) while the common mode voltage was varied and the code distribution of the output was assessed. Figure 4.8 shows the six-sigma deviation of the code distribution in terms of LSB, and the histogram for the zero input code distribution.

The zero-input output code distribution variance is measured to be less than 0.33LSB within the entire 50V common-mode range, which in turn means that the SNDR value of the ADC should be less than a 14 bit level. The conclusion reached here is that the static nonlinear characteristic of the ADC reflects upon its dynamic characteristics in a complex manner to increase the effective noise floor, thereby lowering the measured SNDR.

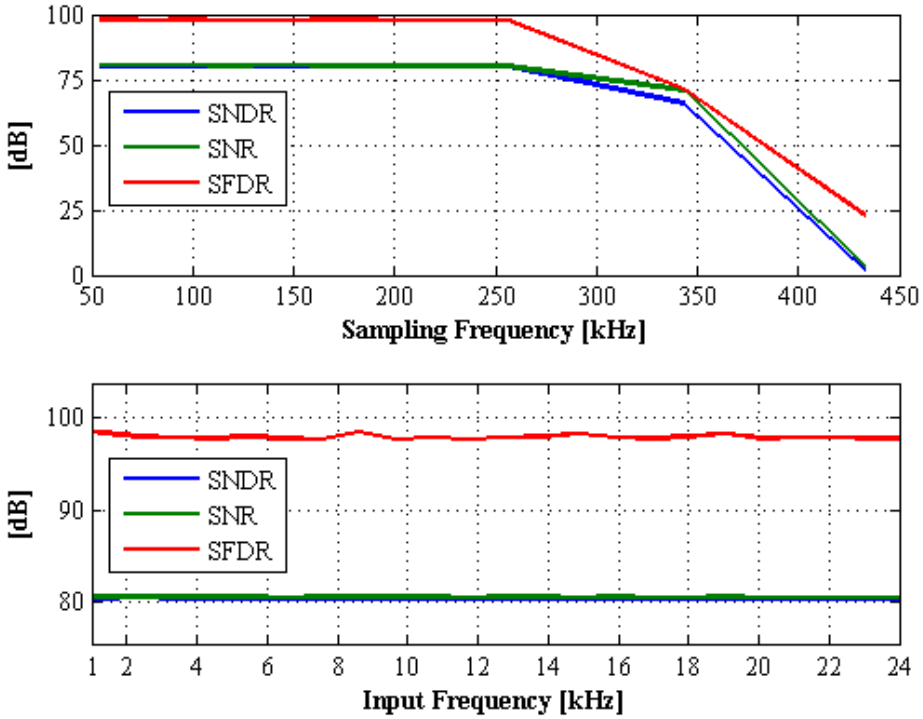


Figure 4.7: SNDR and SFDR versus ADC sampling frequency

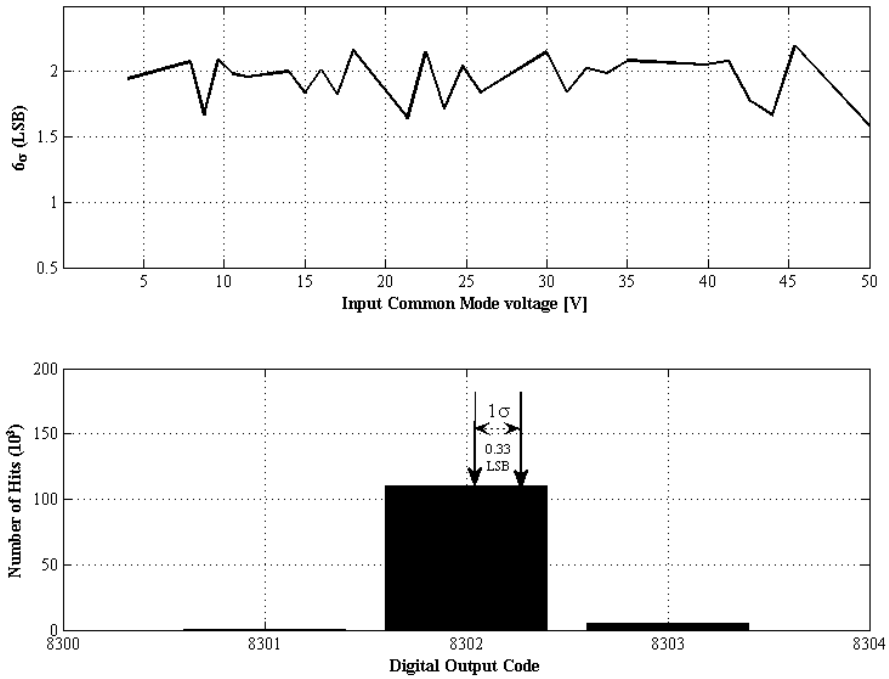


Figure 4.8: Zero input code distribution versus input common mode voltage

4.5 Masurement Results Conclusions

In this chapter, a test methodology for the SAR ADC chosen was developed and measurement results were presented. The ADC achieves 97.8dB SFDR and 80.2dB SNDR while operating at 250kS/s. Achieving the high linearity figure was possible through an analysis of the sources of nonlinearities in both the HVBS and the split CDAC, which was identified as the limiting factor for the 2-step architecture. The split CDAC nonlinearities were mitigated though systematic design and careful layout to overcome parasitic effects, random mismatches and process limitations. The 2-step SAR architecture was identified to be increasing the resolution of a low power converter with minimal silicon area increase, and the achieved a 2.05 pJ/conv-step FoM figure. The designed ADC offers a lower-power alternative for high voltage data acquisition systems, eliminating the need for off-chip or silicon expensive components and high voltage power supply integration.

5. CONCLUSIONS

In this thesis, a 50Vpp input range 14bit 250kS/s SAR ADC has been measured, characterized and analyzed. Methods for extracting primary performance parameters, static (INL-DNL) and dynamic (SNR-SFDR and SNDR) have been analyzed and implemented. The ADC showed 97.8dB SFDR and 80.2dB SNDR while operating at 250kS/s, and $\pm$0.55LSB DNL and $\pm$1.81LSB INL. Further methodology for extracting these parameters from the automated measurement setups were analyzed and developed. The measurement setup and the designed printed circuit board was driven with accuracy requirement in the measurement results in mind. The extracted parameters revealed the effects of nonidealities on the performance that were predicted in the analysis of the SAR ADC circuit components in the previous chapters. The analysis of the chosen SAR ADC was crucial in putting the results into context and constructing goals for the next iteration of the ADC. Similarly, this analysis was made possible by a mathematical definition of quantizer nonidealities The analysis on SAR ADC quantizer nonidealities were verified and were correlated with measurements, and the guidelines reached from the analytic approach and numeric verification. The work that was done in the thesis will serve to ease design work and enable further development of higher performance SAR ADCs.

REFERENCES

[1] **Murmann, B.**, `http://converterpassion.wordpress.com`, retrieved: 25 May 2015.

[2] **Hauser, M.**, 1991. Principles of Oversampling A/D Conversion, *Journal Audio Engineering Society*, **39**, 3–26.

[3] **Schelleng, J.**, 1948, Code modulation communication system, US Patent, No: 2453461 dated 09.11.48.

[4] **B.M., G. and R.P., T.**, 1963, Signal conversion Apparatus, US Patent, No: 3108266 dated 22.10.1963.

[5] **Murmann, B.**, 2008. A/D converter trends: Power dissipation, scaling and digitally assisted architectures, Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE, pp.105–112.

[6] **Maloberti, F.**, 2007. Data Converters, Springer.

[7] **Haenzsche, S.**, **Henker, S. and Schuffny, R.**, 2010. Modelling of capacitor mismatch and non-linearity effects ini charge redistribution SAR ADCs, Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference, pp.300–305.

[8] **Cong, Y. and Geiger, R.**, 2000. Switching sequence optimization for gradient error compensation in thermometer-decoded DAC arrays, *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, **47(7)**, 585–595.

[9] **Van Der Plas, G.**, **Vandenbussche, J.**, **Sansen, W.**, **Steyaert, M. and Gielen, G.**, 1999. A 14-bit intrinsic accuracy Q2 random walk CMOS DAC, *Solid-State Circuits, IEEE Journal of*, **34(12)**, 1708–1718.

[10] **Ozkaya, I.**, 2010. Wide Input Signal Range 14 Bits 1MSPS SAR ADC in 0.35um High Voltage CMOS Process, Ph.D. thesis, ITU.

[11] **Ozkaya, I.**, **Gurleyuk, C.**, **Ergul, A.**, **Akkaya, A. and Aksin, D.**, 2014. A 50V input range 14bit 250kS/s ADC with 97.8dB SFDR and 80.2dB SNR, European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014 - 40th, pp.71–74.

[12] **Mitros, J.**, **Tsai, C.Y.**, **Shichijo, H.**, **Kunz, M.**, **Morton, A.**, **Goodpaster, D.**, **Mosher, D. and Efland, T.**, 2001. High-voltage drain extended MOS

transistors for 0.18- mu;m logic CMOS process, *Electron Devices, IEEE Transactions on*, **48(8)**, 1751–1755.

[13] **Kuyel, T.**, 1999. Linearity testing issues of analog to digital converters, Test Conference, 1999. Proceedings. International, pp.747–756.

APPENDICES

APPENDIX A.1

Matlab code for generating quantization noise by n-bits

```
clear all
close all

n_bits_range = 1:20;

snr_q = zeros(size(n_bits_range));

for i = n_bits_range
    n_bits = i;
    n_codes = 2^n_bits;

    f_s = 1e6;
    t_s = 1/f_s;
    n_fft = 2^11;
    k_fft = 117;

    f_in = f_s/n_fft*k_fft;
    t = 0:t_s:((n_fft-1)*t_s);

    lsb = 2/n_codes;

    y = sin(2*pi*f_in*t);
    q_t = -1:lsb:1;
    q = quantiz(y, q_t(2:(size(q_t, 2)-1)))/(n_codes-1)*2-1;

    snr_q(i) = snr(q, f_in, f_s, n_fft);
end

plot(snr_q, '-d')
hold on
plot(n_bits_range, 6.02*n_bits_range+1.72, 'g-s')
hold off
grid on
legend('Simulated', 'Analytic', 'Location', 'NorthWest')
xlabel('Number of Bits')
ylabel('SNR [dB]')
title('Quantization Noise by Number of Bits')
print -depsc '../img/quantization-noise-by-n-bits.eps'
```

APPENDIX A.2

Matlab code for generating capacitor DAC mismatch

```
clear all
close all

n_runs = 10000;
dac_type = 'therm_dac';
n_bits = 14;
c_u = 30.94e-15;
c_sq = 2e-3;
v_ref = 3.3;

m_a = 1.63e-12;
m_l = 0.146e-12;
m_w = 0;

n_codes = 2^n_bits;

if strcmp(dac_type, 'bin_dac')
    a = c_u./c_sq;
    w = sqrt(a); l = sqrt(a);

    c_arr_mean = c_u*2.^[n_bits-1:-1:0];
    c_arr_a = c_arr_mean./c_sq;
    c_arr_stddev = c_sq./sqrt(2).*sqrt(c_arr_a.*(m_a+m_l+m_w))./300;

    c_arr = normrnd(repmat(c_arr_mean, [n_runs 1]), repmat(c_arr_stddev,[n_runs 1]));

    codes = fliplr(dec_to_bin(0:n_codes-1));
    v_codes = v_ref.*(c_arr*codes')./repmat(sum(c_arr,2), [1 n_codes]);
elseif strcmp(dac_type, 'therm_dac')
    a_c_u = c_u./c_sq;
    w_c_u = sqrt(a_c_u); l_c_u = sqrt(a_c_u);

    c_u_mean = c_u;
    c_u_stddev = c_sq./sqrt(2).*sqrt(a_c_u.*(m_a+m_l+m_w))./300;
    c_u_arr = normrnd(repmat(c_u_mean, [n_runs n_codes]), repmat(c_u_stddev, [n_runs n_codes]));

    c_arr = zeros(n_runs, n_bits);
    for i = 1:n_bits
        c_arr(:,n_bits-i+1) = sum(c_u_arr(:,2^(i-1):2^i-1),2);
    end

    codes = dec_to_therm(1:n_codes);
    v_codes = v_ref.*(c_u_arr*codes')./repmat(sum(c_arr,2), [1 n_codes]);
end

v_fs = max(v_codes,[],2);
v_off = min(v_codes,[],2);
v_lsb = (v_fs-v_off)./n_codes;
```

```
49
50  lsb_nl = 1;
51  codes_nl = (n_codes-1).*(v_codes - repmat(v_off, [1 n_codes])).//(repmat(v_fs - v_off, [1 n_codes]));
52  dnl = diff(codes_nl,1,2)-lsb_nl;
53  inl = cumsum(dnl,2);
54
55  std_dnl = std(dnl, 1);
56  std_inl = std(inl, 1);
57
58  if strcmp(dac_type, 'bin_dac')
59      data = csvread('data/sim_cdac_8bit_binary_mc.matlab',1,0);
60      dnl_meas = data(1:size(data,1), 2:2:size(data,2))';
61      inl_meas = cumsum(dnl_meas, 2);
62
63      std_dnl_meas = std(dnl_meas);
64      std_inl_meas = std(inl_meas);
65  end
66
67  if strcmp(dac_type, 'bin_dac')
68      s_title = sprintf('Binary CDAC, C_u = %e, n_{bits} = %d', c_u, n_bits);
69  elseif strcmp(dac_type, 'therm_dac')
70      s_title = sprintf('Thermometric CDAC, C_u = %e, n_{bits} = %d', c_u, n_bits);
71  end
72
73  figure;
74  subplot(2, 1, 1)
75  hold on
76  stairs(3*std_dnl)
77  if strcmp(dac_type, 'bin_dac')
78      stairs(3*std_dnl_meas, 'g')
79      legend('Analytic', 'Spice Monte-Carlo')
80  end
81  hold off
82  xlim([1 2^n_bits])
83  title(s_title)
84  ylabel('3\sigma(DNL)')
85  subplot(2, 1, 2)
86  hold on
87  stairs(3*std_inl)
88  if strcmp(dac_type, 'bin_dac')
89      stairs(3*std_inl_meas, 'g')
90  end
91  hold off
92  xlim([1 2^n_bits])
93  ylabel('3\sigma(INL)')
94
95  if strcmp(dac_type, 'bin_dac')
96      print -depsc '../img/cdac-capacitor-mismatch-bin.eps'
97  elseif strcmp(dac_type, 'therm_dac')
98      print -depsc '../img/cdac-capacitor-mismatch-therm.eps'
99  end
```

APPENDIX A.3

Matlab Code for postprocessing INL DNL measurement

```
clear all
% close all

inl_min = 0;
inl_max = 0;

A_i = 64; % Interstage Gain

F_clk = 999662.3;
F_s = F_clk/43;
N = 14;
%% Initialize Variables
y = zeros(0,0);
l_y = 0;
l_y_h = 0;
hits = zeros(1, 2^N);
hits_h = zeros(1, 2^8);
hits_cm = zeros(1, 2^N);

i_max = 1;
dirname='W:\\RHVSAR\Chip-2\INL-DNL\temp27\';
%% Read No-Input File
filename = strcat(dirname,'10Vcm0VppFin1.000kHzFs999.6623kHz2.1VrefP3.30Vdd1.csv');
file = csvread(filename);

data = file(:, 2);
time = file(:, 4);

dt = diff(time);
dt_i = find(dt > 2/F_clk);

data_col = vec2mat(data(dt_i(1)+1:dt_i(length(dt_i)-1)), 18);

pow2_8 = 2.^(7:-1:0);
data_h = data_col(:, 1:8)*pow2_8';
data_l = data_col(:, 9:16)*pow2_8';

y = data_h(1:length(data_h)-1).*A_i+data_l(2:length(data_l));
y = y./((A_i)*2^8).*2^14;
y = round(y);

B = mean(y);
SNR = std(y);

for j = 1:length(y)
    hits_cm(y(j)+1) = hits_cm(y(j)+1) + 1;
end

%% Read Files

for i = 1:i_max
    %% Load File with State Sampling
```

```
53          i
54          filename = sprintf('10Vcm0VppFin1.000kHzFs999.6623kHz2.1VrefP3.30Vdd%d.scv', i);
55          filename = strcat(dirname,filename);
56          file = csvread(filename);
57
58          data = file(:, 2);
59          time = file(:, 4);
60
61          dt = diff(time);
62          dt_i = find(dt > 2/F_clk);
63
64          data_col = vec2mat(data(dt_i(1)+1:dt_i(length(dt_i)-1)), 18);
65
66          pow2_8 = 2.^(7:-1:0);
67          data_h = data_col(:, 1:8)*pow2_8';
68          data_l = data_col(:, 9:16)*pow2_8';
69
70          y = data_h(1:length(data_h)-1).*A_i+data_l(2:length(data_l));
71          y = y./((A_i)*2^8).*2^14;
72          y = round(y);
73
74          %% Saturation & Coherence
75              sat_top = 2^14-1;
76              sat_bot = 0;
77
78              y = min(sat_top, max(sat_bot, y));
79          y = y(find(y >= sat_top, 1, 'first'):find(y >= sat_top, 1, 'last'));
80          y = y(find(y==2^(N-1), 1, 'first'):find(y==2^(N-1), 1, 'last'));
81
82          %% Histogram
83          l_y = l_y + length(y);
84
85          for j = 1:length(y)
86              hits(y(j)+1) = hits(y(j)+1) + 1;
87          end
88      end
89      %%
90      [A, B] = gain_offset(y);
91      hits_i = sine_ihpc(A, B, N, l_y);
92
93      iter_d = 4;
94      iter_n = 200;
95
96      inl_pp = zeros(iter_n);
97
98      A_d = A;
99      B_d = B;
100
101     % for i = 1:iter_d
102     % d = 10^(-i);
103     %
104     % for j = 1:iter_n
105     % A_i = A_d - iter_n/2*d + j*d;
106     % for k = 1:iter_n
107     % B_i = B_d - iter_n/2*d + k*d;
108     %
109     % hits_i = sine_ihpc(A_i, B_i, N, l_y);
```

```
110 %
111 % dnl = hits(2:2^N-1)./hits_i(1:2^N-2)'-1;
112 % dnl = dnl - mean(dnl);
113 % inl = cumsum(dnl);
114 %
115 % inl_pp(j, k) = max(inl) - min(inl);
116 % end
117 % end
118 %
119 % [j,k] = find(inl_pp == min(min(inl_pp)))
120 % figure;
121 % surf(inl_pp);
122 % A_d = A - iter_n/2*d + j*d;
123 % B_d = B - iter_n/2*d + k*d;
124 % end
125
126 hits_i = sine_ihpc(A_d, B_d, N, l_y);
127
128 dnl = hits(2:2^N-1)./hits_i(1:2^N-2)'-1;
129 dnl = dnl - mean(dnl);
130 inl = cumsum(dnl);
131
132 %% Plots
133 %close all
134
135 % figure;
136 % plot(hits(2:2^N-1));
137 % hold on
138 % plot(hits_i, 'Color', 'Red');
139 % title('Histogram, N = 14');
140 % xlabel('Code');
141 % ylabel('Number of Hits');
142 % hold off
143 figure;
144 plot(dnl);
145 title('DNL');
146 xlabel('OUTPUT␣CODE');
147 ylabel('DNL␣[LSB]');
148 grid on
149 figure;
150 plot(inl);
151 title('INL');
152 xlabel('OUTPUT␣CODE');
153 ylabel('INL␣[LSB]');
154 grid on
155
156 % str = {'hist', 'dnl', 'inl'};
157 %
158 % h = get(0,'children');
159 % h = sort(h);
160 % for i=1:length(h)
161 % saveas(h(i), str{i}, 'png');
162 % end
```

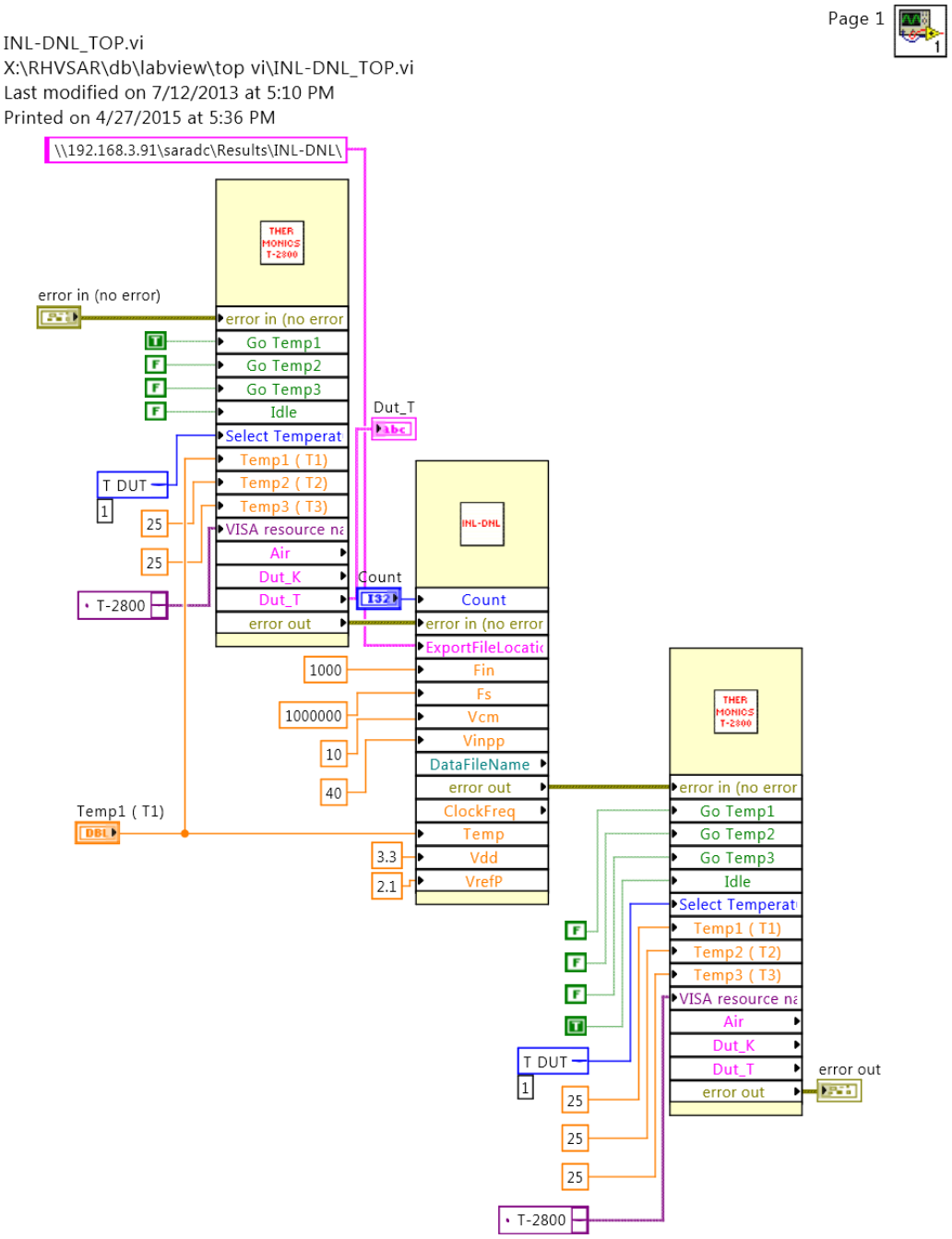
APPENDIX A.4

Matlab Code for postprocessing SNDR SFDR measurement

```
clear all
close all

inl_min = 0;
inl_max = 0;

A_i = 64; % Interstage Gain

F_clk = 623513.7;
F_s = F_clk/43;
T_s = 1/F_s;
N = 16;
L = 2^N;

%% Initialize Variables

%% Read Input File
filename = '\\192.168.3.91\saradc\Results\vsVinpp\8Vcm7VppFin1.000kHzFs624.4290kHz2.1VrefP3.30Vdd1.csv';
file = csvread(filename);

data = file(:, 2);
time = file(:, 4);

dt = diff(time);
dt_i = find(dt > 2/F_clk);

data_col = vec2mat(data(dt_i(1)+1:dt_i(length(dt_i)-1)), 18);

pow2_8 = 2.^(7:-1:0);
data_h = data_col(:, 1:8)*pow2_8';
data_l = data_col(:, 9:16)*pow2_8';

%%
close all;

y = data_h(1:length(data_h)-1).*A_i+data_l(2:length(data_l));
y = y./((A_i)*2^8).*2^14;
y = round(y);

Y = fft((y(1:L)-mean(y(1:L)))./2^13, L)/L;
f = F_s/2*linspace(0, 1, L/2+1);

Y_log = 20*log10(2*abs(Y(1:L/2+1)));
Y_log_s = sort(Y_log);

F_in = f(find(Y_log == max(Y_log)));
y = (y(1:L)-mean(y(1:L)))./2^13;

w = rectwin(L);
[SNR,ptotdB,psigdB,pnoisedB] = calcSNR(y', F_in./F_s, L/2, w', L, 1);
SNR
SFDR = Y_log_s(length(Y_log_s))-Y_log_s(length(Y_log_s)-1)
```

```
53
54  figure(1);
55  stairs(ptotdB(1:(L/2-1))+3.02);
56  figure(2);
57  stairs(psigdB(1:(L/2-1))+3.02);
58  figure(3);
59  stairs(pnoisedB(1:(L/2-1))+3.02);
60
61  str = {'total', 'signal', 'noise'};
62
63  h = get(0,'children');
64  h = sort(h);
65  for i=1:length(h)
66     saveas(h(i), str{i}, 'png');
67  end
```

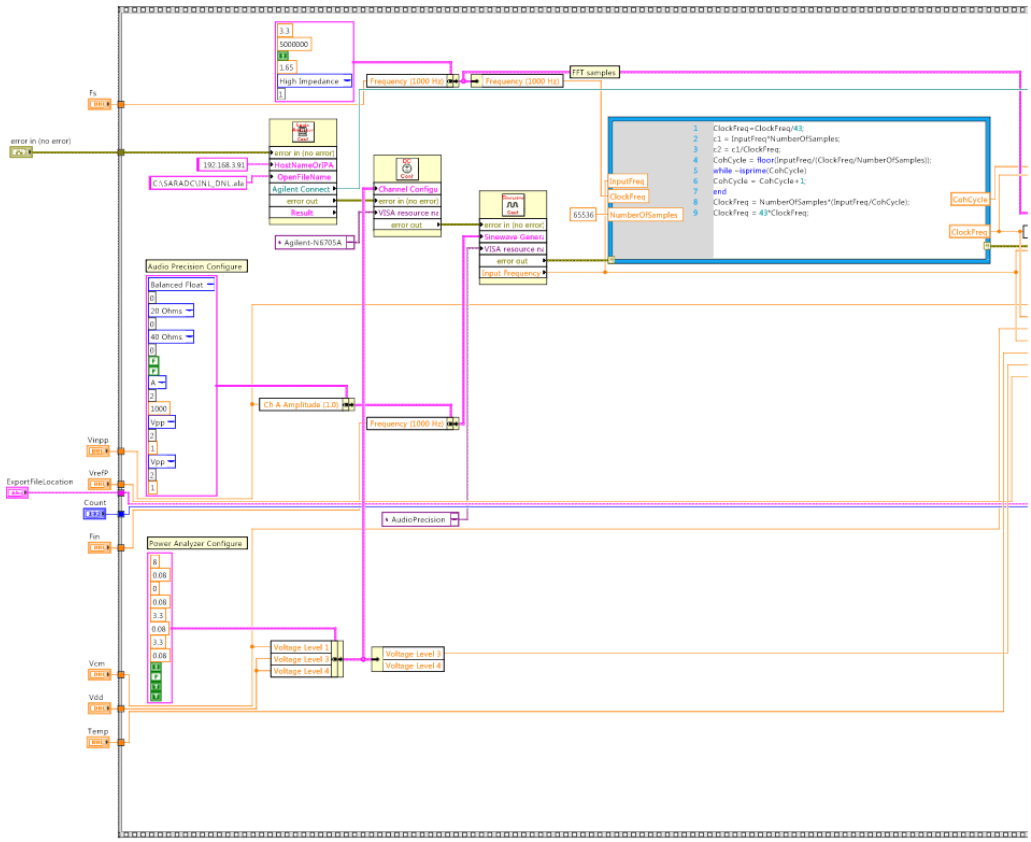
APPENDIX A.5


Page 1
INL-DNL_TOP.vi
X:\RHVSAR\db\labview\top vi\INL-DNL_TOP.vi
Last modified on 7/12/2013 at 5:10 PM
Printed on 4/27/2015 at 5:36 PM
\\192.168.3.91\saradc\Results\INL-DNL\
THERMONICS T-2800
error in (no error)
Go Temp1
Go Temp2
Go Temp3
Idle
Select Temperat
Temp1 (T1)
Temp2 (T2)
Temp3 (T3)
VISA resource na
Air
Dut_K
Dut_T
error out
Dut_T
T DUT
1
25
25
T-2800
Count
INL-DNL
Count
error in (no error
ExportFileLocatio
Fin
Fs
Vcm
Vinpp
DataFileName
error out
ClockFreq
Temp
Vdd
VrefP
1000
1000000
10
40
3.3
2.1
Temp1 (T1)
THERMONICS T-2800
error in (no error
Go Temp1
Go Temp2
Go Temp3
Idle
Select Temperat
Temp1 (T1)
Temp2 (T2)
Temp3 (T3)
VISA resource na
Air
Dut_K
Dut_T
error out
error out
T DUT
1
25
25
25
T-2800

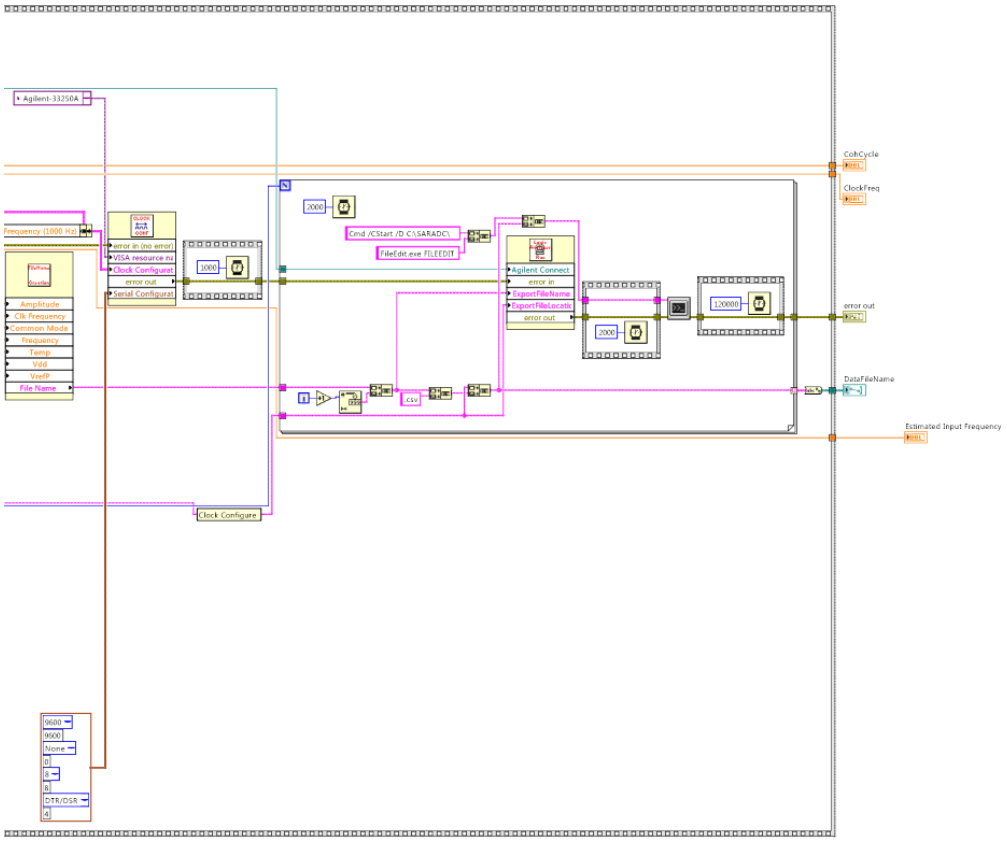
APPENDIX A.6

INL-DNL

INL-DNL.vi
X:\RHVSAR\db\labview\test setup\INL-DNL.vi
Last modified on 7/12/2013 at 4:44 PM
Printed on 4/27/2015 at 5:35 PM

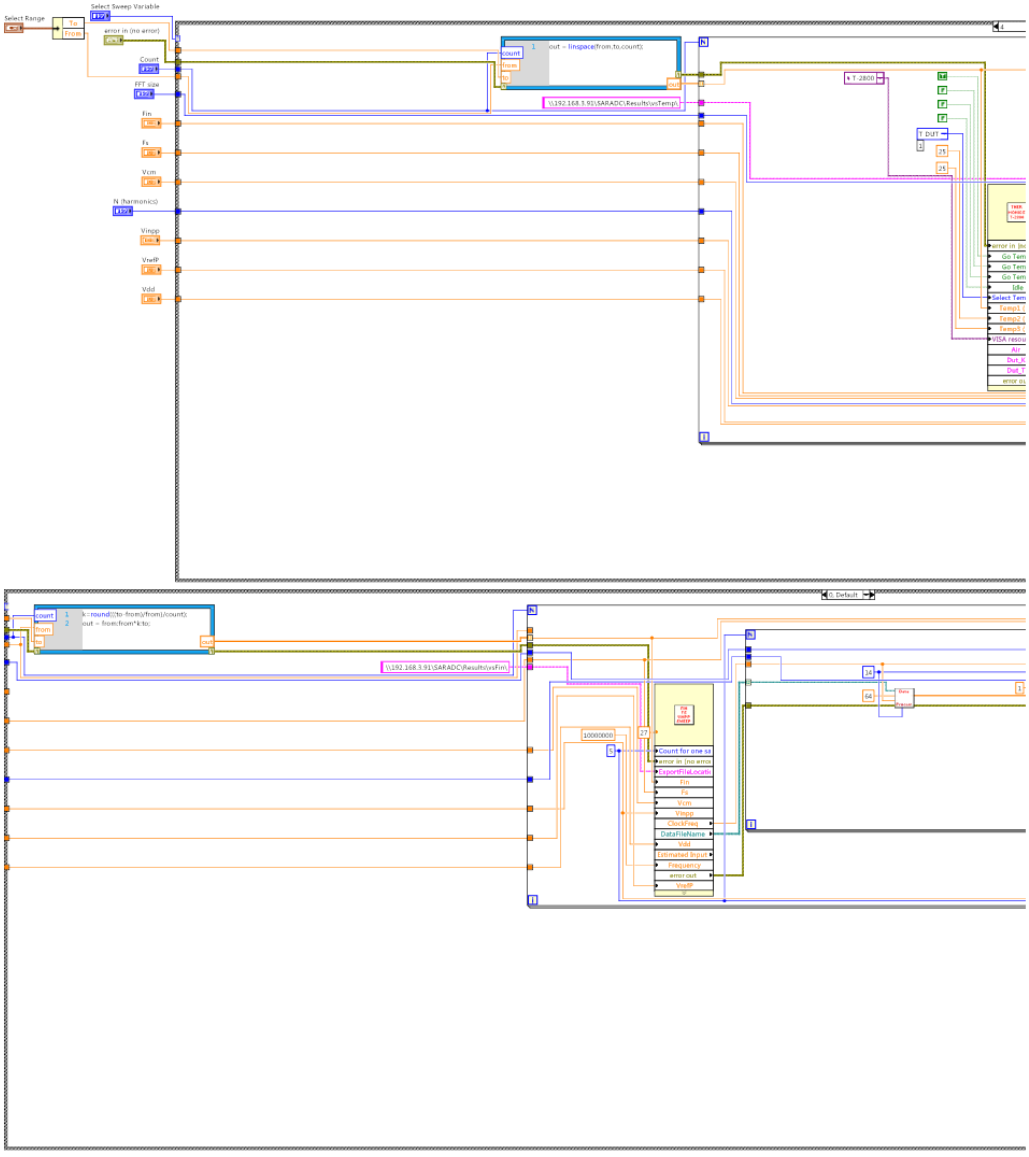


INL-DNL.vi
X:\RHVSAR\db\labview\test setup\INL-DNL.vi
Last modified on 7/12/2013 at 4:44 PM
Printed on 4/27/2015 at 5:35 PM

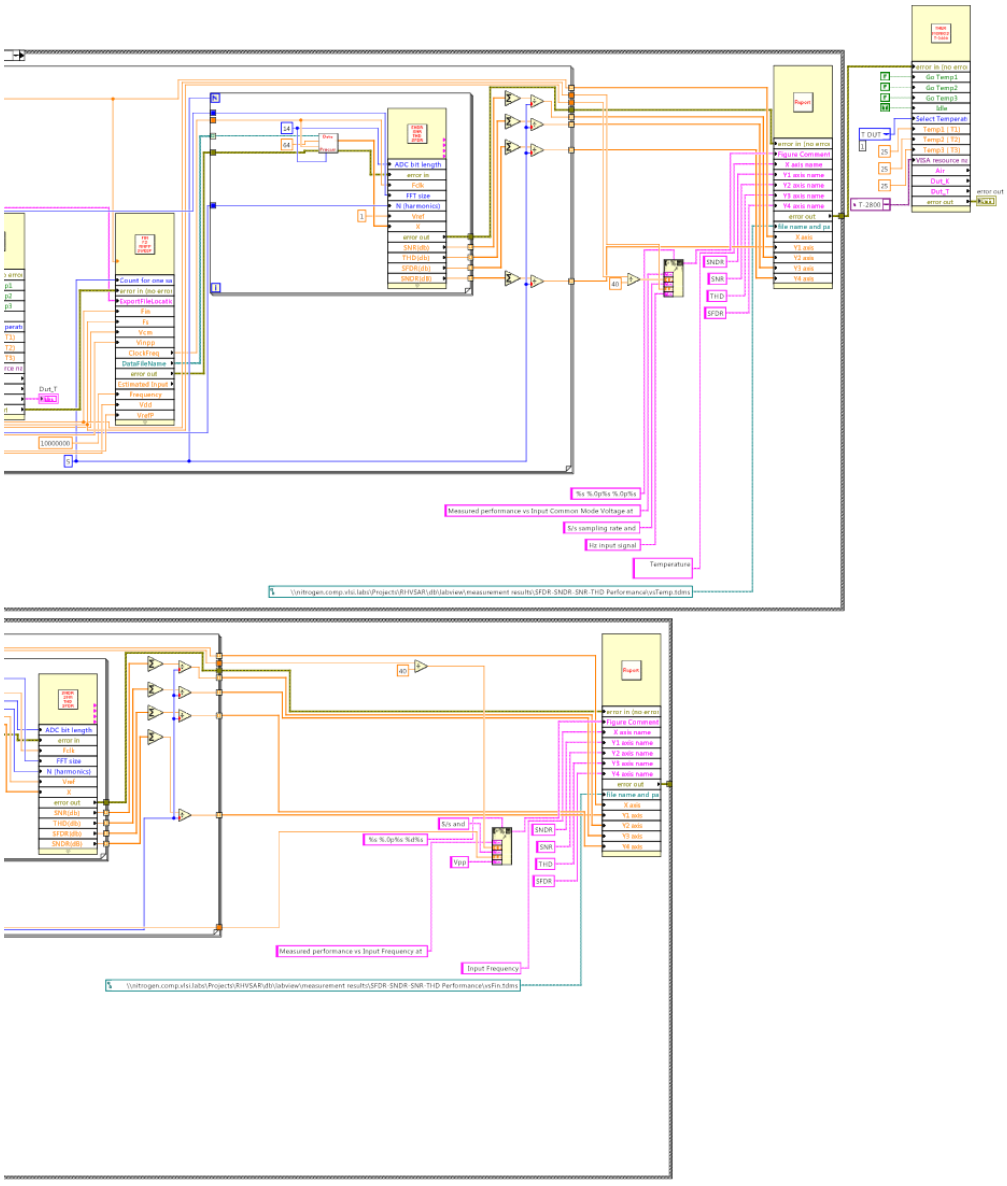


APPENDIX A.7

SFDRversusFsFinVinPPVcm_TOP.vi
X:\RHVSAR\db\labview\top vi\SFDRversusFsFinVinPPVcm_TOP.vi
Last modified on 7/8/2013 at 10:50 AM
Printed on 4/27/2015 at 5:36 PM



SFDRversusFsFinVinPPVcm_TOP.vi
X:\RHVSAR\db\labview\top vi\SFDRversusFsFinVinPPVcm_TOP.vi
Last modified on 7/8/2013 at 10:50 AM
Printed on 4/27/2015 at 5:36 PM



CURRICULUM VITAE

Name Surname:
Çağrı Gürleyük

Place and Date of Birth:
Istanbul, 22.01.1987

Adress:
Bahçeköy Yolu Üstü Sırma Sok. Özden Sitesi No:13
B3 D4 Sarıyer/Istanbul

E-Mail:
cgurleyuk@gmail.com

B.Sc.:
Electronics Engineering, Istanbul Technical University, 2012

M.Sc.:
Electronics Engineering, Istanbul Technical University, 2015

Professional Experience and Rewards:
Design Engineer at Analog Devices, Istanbul

PUBLICATIONS/PRESENTATIONS ON THE THESIS

▪ Ozkaya, I.; **Gurleyuk, C.**; Ergul, A.; Akkaya, A.; Aksin, D.Y., "A 50V input range 14bit 250kS/s ADC with 97.8dB SFDR and 80.2dB SNR," European Solid State Circuits Conference (ESSCIRC), ESSCIRC 2014 - 40th , vol., no., pp.71,74, 22-26 Sept. 2014