ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

ENGINEERING AND TECHNOLOGY

**BEHAVIORAL MODELLING OF
SAR ADC**

M.Sc. THESIS

Mehmet Arda AKKAYA

Department of Electronics and Communications Engineering

Electronic Engineering Programme

MAY 2015

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

**BEHAVIORAL MODELLING OF
SAR ADC**

M.Sc. THESIS

**Mehmet Arda AKKAYA
(504121369)**

Department of Electronics and Communications Engineering

Electronic Engineering Programme

Thesis Advisor: Prof. Dr. Ali Toker

MAY 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SAR ADC'NİN
DAVRANIŞSAL MODELLENMESİ**

YÜKSEK LİSANS TEZİ

**Mehmet Arda AKKAYA
(504121369)**

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Ali Toker

MAYIS 2015

Mehmet Arda AKKAYA, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504121369 successfully defended the thesis entitled **"BEHAVIORAL MODELLING OF 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 :	**Yrd. Doç. Dr. Metin YAZGI**	....
	Istanbul Technical University	
	Prof. Dr. Osman PALAMUTÇUOĞULLARI	....
	Beykent University	

Date of Submission :   04 May 2015
Date of Defense :   26 May 2015

To my family,

FOREWORD

First of all, I would like to express my appreciation and thanks to Prof. Dr. Ali Toker for accepting me as his student and sharing his wisdom and guidance with me.

I would like to thank my dear friends Atilim Ergul, Mert Ozdemir for their support, help and discussions during my research life. This thesis would not have been possible without their contributions.

I owe my deepest gratitude to Rengin Haksal for motivating and believing in me without questioning my manner. I will be forever thankful for her love and patience.

Last but not least, I would like to express my sincere gratitude my family for their unconditional support.

May 2015   Mehmet Arda AKKAYA

x

TABLE OF CONTENTS

Page

ABBREVIATIONS

AC	**:**	Alternating Current
ADC	**:**	Analog to Digital Converter
DAC	**:**	Digital to Analog Converter
DC	**:**	Direct Current
IC	**:**	Integrated Circuit
HVBS	**:**	High Voltage Bootstrapped Switch
LSB	**:**	Least Significant Bit
MIM	**:**	Metal-Insulator-Metal
MSB	**:**	Most Significant Bit
MSps	**:**	Mega Samples per Second
OPAMP	**:**	Operational Amplifier
SAR	**:**	Successive Approximation Register
SPI	**:**	Serial Peripheral Interface
SoC	**:**	System on Chip

LIST OF TABLES

LIST OF FIGURES

BEHAVIORAL MODELLING OF
SAR ADC

SUMMARY

During the last two decades, the usage of integrated circuits have been increased in modern electronic systems due to development of the process technology while offering low cost and high performance. Also it leads to powerful digital signal processing techniques that require an analog-to-digital converter (ADC) to communicate with analog world.

Nowadays, ADCs (Analog to Digital Converters) become very popular with the increase of mixed signal applications in electronic. Moreover, energy consumption have been the most significant topic in the last decade. Amongst various ADC architectures, SAR (Successive Approximation Register) ADC becomes most common ADC architecture as it is suitable for low power, medium to high resolution and medium speed applications.

Previously, Capacitive DAC array was the most critical block of SAR ADC, due to the matching of the elements in the DAC that determines the total noise contribution and linearity of SAR ADC. Developing technology gives an opportunity to implement the DAC with smaller area and better matching and as a result it is possible to reach higher resolution with SAR ADC. However other non-idealities of the blocks in SAR ADC must be taking into account to attain higher resolution together with high speed.

The increase of transistor number in the circuit due to the development of process technology cause long transistor-level simulation time that can be take days to complete. However, Matlab or Simulink models offer fast and accurate results that can be used to determine design specifications or to investigate exist architecture.

In this thesis, complete behavioral model of a two-step SAR ADC architecture is studied in Matlab Simulink environment to analyze non-idealities of capacitive DAC, comparator, opamp, clock signal and reference voltages by using charge equations that are derived to find transfer functions of the SAR-ADC. Since all block parameters are set as a variable, it is possible to use model for any *N*bit SAR ADC architecture with small changes. Moreover the model shows promising results with respect to measurement results of the SAR-ADC that is also studied in the thesis. As a consequence, accomplished model is a powerful tool that helps to understand basic and complex behaviour of the SAR ADC and provides information about limiting factors for ADC in a short time period compared to transistor-level simulations.

SAR ADC'NİN
DAVRANIŞSAL MODELLENMESİ

ÖZET

Son yirmi yılda, elektronik sistemlerde kullanılan tümdevre sayısı giderek artmıştır. Bunun sebebi, CMOS teknolojisinindeki gelişmeler sonucunda daha yüksek performasa sahip tümdevrelerin daha ucuz fiyata daha küçük alanda üretilmesine olanak sağlamasıdır. Ayrıca, bu gelişmeler karmaşık sayısal işaret işleme tekniklerinin de ortaya çıkmasına yardım etmiştir. Böylece analog ve sayısal sistemler iç içe girmeye başlamışlardır.

Tüm işaretlerin analog işaretler olduğu dünyamızla sayısal işaretlerin işlendiği dünya arasındaki bağlantı analog-dijital çeviriciler ile sağlanmaktadır. Analog-dijital çeviriciler (ADC), dijital işaret işleme tekniklerinin gelişmesi ile birlikte oldukça yaygın hale gelmiştir. Analog bir işaret analog-dijital çevirici sayesinde dijital koda dönüştürülmekte ve işlenmek üzere dijital sinyal işleme bloğuna gönderilmektedir. Analog-dijital çevirici dönüştürme işlemi sırasında sırasıyla örtüşme önleyici süzgeç, örnekleme, kuantalama ve kodlama fonksiyonlarını gerçekleştirmektedir. Öncelikle giriş işareti dışındaki tüm işaretleri süzmek için örtüşme önleyici süzgeç kullanılır. Daha sonra bu işaret belli bir periyotta işlenmke üzere örneklenir. Örneklenen bu işaretin hangi quantalama aralığına düştüğünü quantalama bloğu belirler. Son olarak da bu işaret sayısal bir kod olarak karşımıza çıkar

Başlıca ADC yapıları incelendiğinde: SAR ADC, Delta Sigma ADC, Flash ADC karşımıza çıkmaktadır. Her bir ADC yapısı belli bir mühendislik problemini çözebilmek adına oluşturulmuşlardır. Diğer ADC türleri ile kıyaslandığında, SAR ADC orta ve yüksek çözünürlük ile beraber düşük güç tüketimi sağlamayabilmesi, tercih sebebi olmasını sağlamıştır. Ayrıca SAR algoritmasının basitliği sebebiyle yapı kolaylıkla anlaşılabilmektedir.

Basit bir SAR ADC yapısında örnekleme bloğu, karşılaştırıcı bloğu, SAR lojiğini gerçekleştirecek sayılsal devre ve en önemlisi dahili bir dijital-analog çevirici (DAC) bulunmaktadır. İlk olarak, analog giriş işareti örneklme bloğunda örneklenmektedir. Daha sonra örneklemiş giriş işareti karşılaştırıcı bloğunda dahili DAC çıkışı ile karşılaştırılır. Bu ilk karşılatırmada DAC orta seviyedeki kod değerindedir. Buna göre en önemli bit değeri hesaplanır ve DAC yeni değerine yerleştirilir. DAC'ın sahip olduğu eleman sayısı kadar bu işlem devam eder. Tüm elemanlar için karşılaştırma yapıldığında çıkış işareti hazırlanmış olmaktadır. İki aşamalı SAR ADC yapısı da aynı bloklardan oluşmaktadır. Bu bloklara ek olarak bir de kuvvetlendirici bloğu bulunmaktadır. İlk SAR dönüşümü tamamlandığı zaman, giriş işareti ve nihai DAC değeri arasındaki fark bize quantalama hatasını vermektedir. Bu hata kuvvetlendirici bloğu sayesinde kuvvetlendirilerek tekrardan bir SAR dönüşümü yapmak üzere örnekleme bloğuna gönderilir. İkinci SAR dönüşümü de tamamlanmasıyla beraber, ilk ve ikinci SAR dönüşümlerinde elde edilen sayısal değerler birleştirmek üzere sayısal hata bloğuna yollanır. Bu blok bize nihai sayısal çıkış işaretini verecektir.

Dahili DAC içerisinde kullanılan kapasite elemanlarının uyumu, SAR ADC performansını belirleyen en önemli etkendir. Tüm işlemler iki kapasite arasındaki orana göre gerçekleştirildikleri için eleman değerlerindeki saçılma doğrudan performası düşürmektedir. CMOS teknlojisindeki gelişmeler sonucu daha küçük alanda uyumu daha yüksek kapasiteler geçeklenebilmektedir. Böylece SAR ADC ile daha yüksek çözünürlük değerlerine gidilebilmektedir.

Proses teknolojisinin gelişmesi ile beraber kapasitif DAC daha küçük alanda daha uyumlu elemanlarla gerçeklenebilmektedir. Böylece SAR ADC ile daha yüksek çözünürlük değerlerine gidilebilinmektedir. Fakat daha yüksek çözünürlükle beraber daha yüksek hızlara gidilmek istenildiğinde SAR ADC yapısındaki diğer blokların da performansa etkileri ortaya çıkmaktadır. Bu etkilerin nelerden kaynaklandığı ve performansı ne kadar etkiledikleri önemli bir soru haline gelmektedir.

Gelişen teknolojiyle beraber, tümdevre yapılarını incelemek için kullanılan programlar da oldukça ilerlemiştir. Fakat günümüzün karmaşık elektronik devleri incelerken tasarımın her aşamasında bu programları kullanmak oldukça fazla zaman kaybına yol açmaktadır. Tümdevre yapılarındaki transistor sayısının artması, tümdevreyi analiz etmek için kullanılan bu programların daha yavaş çalışmasına sebep olmaktadır. Günümüzde kullanılan en basit bir tümdevremin bile çok sayıda transistor içerdiği düşünülürse, karmaşık bir tümdevre yapısının transistor seviyesindeki analizinin daha yavaş olacağı açıktır. Bir devrenin simulasyon süresi günler mertebesinde sürebilmektedir.

Analog-dijital çeviriciler karmaşık tümdevre yapılarından biridir ve bu yüzden tasarıma transistor seviyesi yerine blok seviyesinde başlamak akıllıca bir başlangıç olmaktadır. Böylece istenilen performası sağlayabilmek için tümdevreyi oluşturan blokların ihtiyaç duyduğu tasarım parametreleri belirlenmiş olur. Tasarım sırasında tasarlanan bloklara ait parametreler, model içerisine yerleştirilerek istenilen hedefe ne kadar yaklaşıldığı da araştıralabilir. Son olarak tasarımı bitmiş bir tümdevrenin parametreleri yine model içine eklenerek performansı en çok etkileyen blokların hangileri olduğu hakkında fikir sahibi olunabilir. Matlab ve Simulink ortamında kurulan modeller hızlı ve doğru sonuçlar vererek, tasarım öncesinde ve sonrasında tasarım parametrelerinin belirlenmesinde ve varolan yapı hakkında bilgi edinmede kullanılabilirler.

Bu çalışmada, iki adımlı SAR ADC mimarisinin davranışsal modellenmesi, devredeki idealsizlikler dahil edilerek incelenmiştir. Tüm bloklar Simulink ortamında tasarlanmıştır ve bloklara ait tüm parametreler değişken olarak kabul edilmiştir. Böylece oluşturulan model herhangi bir hızdaki ve herhangi bir çözünürlükteki SAR ADC yapısını gerçekleştirmek için kullanılabilmektedir.

SAR ADC yapısındaki en önemli blok olan kapasitif DAC analizi için yoğun bir çaba harcanmıştır. Ek olarak daha düşük alanda aynı fonksiyonu gerçeklemek için kullanılan zayıflatıcı kapasitenin etkisi de incelenmiştir. Her bir kapasite elemanın saçılması ve parazitik kapasitelerin etkisi de bu analize dahil edilmiştir. Bir sonraki önemli blok olan karşılaştırıcı içinse yükselme ve düşme zamanı, girişte alıglanabilecek en küçük gerilim ve dengesizlik gerilimi modellenmiştir. İki aşamalı SAR yapısı kuvvetlendiriciye ihtiyaç duyduğu için bu blok da modellenmiştir. Ayrıca tüm işlemler saat işaretine göre gerçekleştirildikleri için saat işaretinin sahip olabileceği idealsizlikler de modellenmiştir. Son olarak her bir referans geriliminin etkisini modellemek için değişkenler kullanılmıştır.

Oluşturalan model sonuçları ile ölçüm sonuçlarının tutarlı olduğu gözlenmiştir. Laboratuvar ortamında tamamlanması saatler süren ölçümler kurulan model ile hızlı bir şekilde elde edilebilmektedir. Sonuç olarak, gerçekleştirilen davranışsal model ile SAR ADC performansı kısa zaman diliminde doğru bir şekilde inceleme fırsatı sunmaktadır.

1. INTRODUCTION

1.1 Introduction

The evolution of integrated circuits(ICs) dominates modern electronic systems during the last two decades while offering low cost, high performance (Razavi, 2001). Furthermore, it leads to a rapid improvement on implementation of digital signal processing techniques. Since every signal in real world are analog continuous signals, analog to digital converters (ADCs) and digital to analog converters (DACs) are indispensable blocks of electronic systems to establish connection between the physical world and the electronic system. Today, complex digital signal processing methods are applicable in a consequence of developments in ADCs due to scaling of process technology that leads to more demanding specifications for ADCs in the sense of sampling rate and conversion accuracy (P. Carbone, 2014).

While supply voltage and input signal level reduce dramatically due to scaling of process technology, it is harder to process high voltage signals where high voltage data acquisition is necessary. High voltage data acquisition is generally utilized within ADCs in power management System-on-Chips (SoCs) where input signal voltage exceeds power supply voltage (Aksin & Ozkaya, 2009). The traditional methods for high voltage data acquisition are attenuating input signal by using resistive divider or increasing system supply voltage as stated in (Sadkowski, 2013). However, the first method, using resistive divider, loads the input, results signal degradation and consumes higher silicon area, the second method increase power consumption of system. An ADC that directly samples high voltage input signal without including any additional circuit while using low supply voltage solve both problems.

Generally, starting design of ADC from transistor-level is an unsuccessful attempt without knowing theoretical limits of the selected architecture. Also, transistor-level simulation of the whole system dramatically takes a long time due to increasing number of transistors in ICs year by year. As a result, simulation time has been becoming dominant time factor for integrated circuit design period. However

behavioral modeling gives a chance to analyze low-level effect that may arise in real circuits during the design period (Osipov & Bocharov, 2012). In addition, behavior modeling provides an approximation about performance of the system that will be designed while it offers much shorter calculation time against transistor-level simulation. Therefore, in the beginning of data converter design, behavior modeling becomes the most important part as it provides the information that how non-idealities affects the whole system performance.

Each ADC architecture require own behavioral model since they use different conversion algorithm to convert analog signal. Delta-Sigma ADCs are the most suitable architecture for creating a behavioral model that can give very accurate results according to actual design because it can easily analysed mathematically. So, there are many studies about Delta-Sigma ADC modelling in the literature (Jarman, 1995). On the other hand, behavioral modeling solutions are insufficient for SAR ADC architectures that are widely use in SoC (Lin, Yang, Zhong, Sun, & Xia, 2005). In addition, there is less studies on a 2step SAR architecture that offers to examine design trade-off between different factors. As a result, designers cannot analyze the effects of non-idealities in SAR ADC deeply without using a behavioral model by the reason of long transistor-level simulation times and lack of complete model of the SAR ADC.

There are several studies which proposes behavior models for SAR ADC characterization. Since capacitive DAC array is the most limiting part that determines noise and linearity performance of SAR ADC directly, the studies lay on analysis of it. The effect of parasitic in a split capacitor DAC array is examined in (Osipov & Bocharov, 2012) but it does not take account effects of comparator, reference voltage etc. . Also, in (Fredenburg & Flynn, 2012) and (Lin et al., 2005) analysis of element mismatch is investigated deeply but other effects are eliminated during the anlysis. Another paper is proposed to model a charge redistribution SAR ADC in 2010 (Haenzsche, Henker, & Schuffny, 2010). However it does not investigate amplifier and comparator non-idealities also. As seen from proposed papers up-to today, they all only focused on the capacitor array modeling but ignore comparator and amplifier and clock signal non-idealites that are becoming an important parameter while resolution and speed of ADC are increased. Therefore it is important to generate a 2-step SAR architecture model including the effects of all non-idealties in the system.

In this work, a 14bit 2-step SAR ADC architecture is measured, analyzed and then a

complete SAR model is built in Matlab Simulink by using derived transfer functions of the ADC. A model of split capacitor array is included the model to investigate effects of parasitic and capacitance mismatch on non-linearity. Also, a comparator model is introduced to determine performance requirement of it for N-bit resolution. Another important point of 2-step SAR architecture is inter-stage gain that is used to obtain 14-bit resolution from two 8-bit conversion. Because the variation of inter-stage gain causes SNR degradation even both 8-bit conversions are completed without any mistakes. Since every step of ADC is determined with respect to clock signal, non-idealities of clock signal are also included the model. Finally, measurement results of the SAR-ADC are examined in the established model to identify design problems which cause performance degradation. In order to offer a solution for any N-bit SAR ADC architecture, every parameters in the model are kept as variable that can alter by designer. While using the SAR model, circuit design considerations about comparator, opamp and DAC can also be determined. The model total simulation time is just 5 minutes with a standard computer. Therefore it is useful to start the design with the model instead of transistor level design. In addition, transient simulation time of transistor level system requires enormous data to make proper calculation and bigger data result as larger simulation time. As a consequence, the model grants shorter simulation time with accuracy close to transistor-level simulation by including many non-idealties.

1.2 Organization of Thesis

This thesis is organized in 6 chapters.

In this chapter, a brief summary of the thesis is introduced.

In chapter 2, general information about ADC is given. Then, performance metrics of ADC including static and dynamic parameters are explained in detailed.

In chapter 3, selected conversion method and 2-step SAR ADC architecture is examined deeply. Also measurement results of the presented ADC and selection of the measurement instruments while characterizing the SAR-ADC are expressed.

In chapter 4, transfer functions of the measured SAR-ADC architecture is calculated for every phase of the ADC by using charge equations. At the end of the

chapter, important parameters as inter-stage gain , full-scale are investigated to satisfy demanded conversion resolution and full-scale.

In chapter 5, behavioral model of the SAR-ADC is presented. First a simple voltage model is described to explain two-step SAR architecture. Following, established charge model and each block in the model are introduced. In addition effects of non-idealities for each block is included to the model.

Finally, chapter 6 concludes the thesis and discuss the future works.

2. BASIC PRINCIPLES OF ADCs

2.1 Introduction To Analog Digital Converters

An Analog to Digital Converter (ADC) is an electronic device that transform continuous analog signals to digital equivalent for signal processing. It is the interface between the analog and digital world. A basic ADC operation can be consist of four functions: anti-aliasing filtering, sampling, quantization and data coding. The block diagram of an ADC that satisfy mentioned functions is shown in Figure 2.1 (Maloberti, 2007).
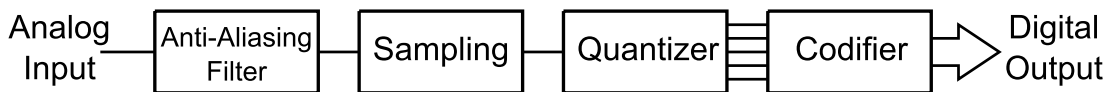
Figure 2.1 : Block diagram of an ideal ADC.

The sampling function is used to transform continuous analog signal to its sampled data equivalent which is comprised of the same discrete time intervals. During sampling time period, T_S, input signal is captured with exact amplitude value in exact time. Figure 2.2 shows an analog signal that is sampled at frequency f_S (Zhang, 2009). Obviously, more sampled data results more precise representation of analog signal. In contrast, less sampled data results less accurate representation of analog signal. This deduction leads us to common fact which is known as Niyquist's criteria ($f_S > 2f_B$).
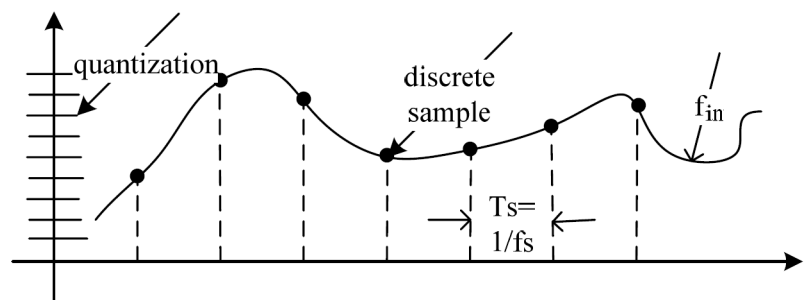
Figure 2.2 : Continuous time signal and its sampled data

In a sampled data system, sampling rate, f_S, must be bigger than twice of an analog signal bandwidth f_B. If sampling rate is smaller than two times f_B ($f_S < 2f_B$) then aliasing will occur and it causes loss of information. Therefore, an anti-aliasing filter is placed in front of the sampler in order to avoid aliasing. It restricts the band of the signal, so out of band interferes are removed. The anti-aliasing filter selection is related to both the sampling rate f_S and the demanded dynamic range. The filter order is calculated from rule of thumb formula as $A_{SB}|_{dB}/20log_{10}[(f_S - 2f_B)/f_B]$ where A_{SB} is stop band attenuation(Maloberti, 2007). When transition band is narrow, filter order must be increased to obtain a desired attenuation value but a higher filter order cause more complexity in filter structure. On the contrary, larger transition band reduce the filter order and is simplified filter complexity. However, larger bandwidth requires high speed circuit to realize main building blocks of ADC.
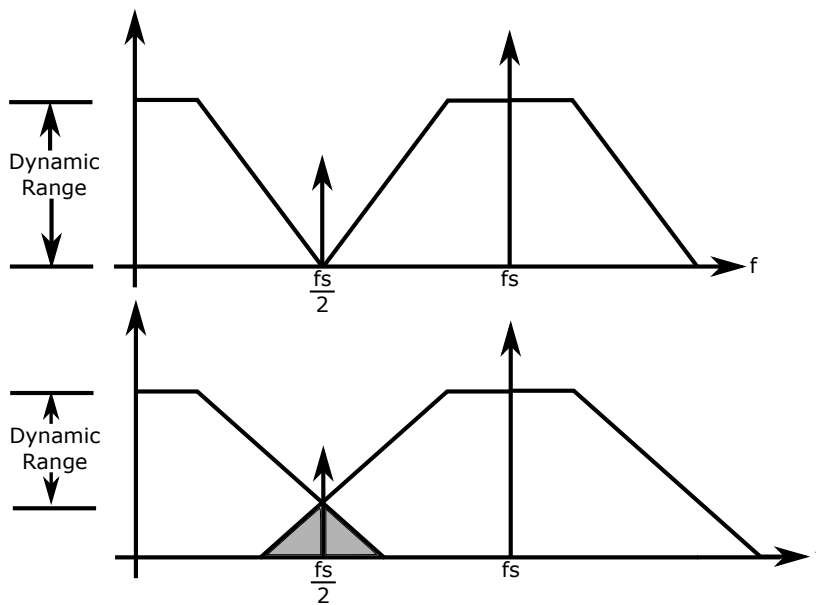
Figure 2.3 : Spectrum of the signal when aliasing occurs

Figure 2.3 shows the effects of aliasing on the dynamic range of sampled signal (Kester, 2007). In the top part of the figure, the sampling frequency is bigger than two times f_B and so that aliased component do not interfere with the input signal. In the bottom part of the figure, the sampling frequency is less than two times f_B and so that aliased component overlap and modify the input signal by reducing the dynamic range.

After the sampling operation, sampled-data signal is transformed to discrete level by a quantizer which decides amplitude of the sampled-data signal by using equal level

6

intervals. The amplitude of the interval step is ratio between the dynamic range of the interval and the number of the quantization intervals. If the full scale of is X_{FS} and for N-bit resolution the number of quantization intervals is 2^N, the quantization step Δ becomes

$$\Delta = \frac{X_{FS}}{2^N} = 1LSB \tag{2.1}$$

In ideal quantizer, the number of the intervals goes to infinity and the quantization error becomes zero but in real, the quantizer generates an reluctant error because of the limited interval numbers and it limits the signal-to-noise ratio (Lundberg, 2002). Assume that, the quantizer output is Y for input signal X_{in}

$$Y = X_{in} + \varepsilon_Q \tag{2.2}$$

where ε_Q is the quantization error. For input signal X_{in} larger than Δ, the quantization error ε_Q is a random quantity that ranges from $-\Delta/2$ to $\Delta/2$. Also, the probability distribution function $p(\varepsilon_Q)$ can be modeled as a uniform distribution in Figure 2.4 (Zhang, 2009).
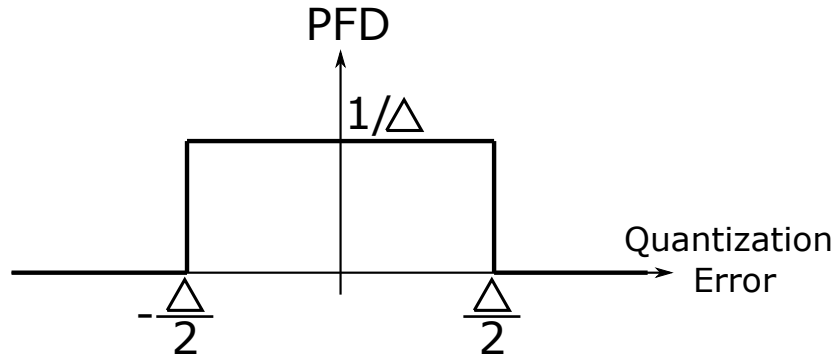
Figure 2.4 : Probability density function of quantizer error

The time average power of ε_Q is

$$P_Q = \int_{-\infty}^{\infty} \varepsilon_Q^2 \cdot p(\varepsilon_Q) d\varepsilon_Q = \int_{-\Delta/2}^{\Delta/2} \frac{\varepsilon_Q^2}{\Delta} d\varepsilon_Q = \frac{\Delta^2}{12} \tag{2.3}$$

As it stated before, equation (2.3) proves that the quantization power decreases as the number of the quantization intervals increases. Also, equation (2.3) leads to the relation between SNR and resolution. Let a sine wave input has peak voltage X_{FS} and the power of the input signal is calculated as:

$$P_{sin} = \frac{1}{T} \int_0^T \frac{X_{FS}^2}{4} sin^2(2\pi f t) dt = \frac{X_{FS}^2}{8} \tag{2.4}$$

Once input signal and quantization noise power is calculated, maximum achievable SNR is

$$SNR = \frac{P_{sin}}{P_Q} = \frac{X_{FS}^2}{8} \cdot \frac{12}{\Delta^2} \tag{2.5}$$

by using equation (2.1)

$$SNR = \frac{(\Delta \cdot 2^n)^2}{8} \cdot \frac{12}{\Delta^2} = \frac{3}{2} 2^{2n} \tag{2.6}$$

Then SNR in dB is written as

$$SNR|db = 10 log_{10} = (6.02 \cdot n + 1.78) dB \tag{2.7}$$

Equation (2.7), which is first examined in (Bennett, 1948), is a rule of thumb formula for data converters. It determines maximum achievable SNR value for an ideal *N*-bit ADC.

The last function of the ADC is coding the quantized signal. At the end of the coding, ADC generates bunch of binary data for given analog input signal. The coding is related to how output binary data is defined. There are several coding schemes as unipolar straight binary, complementary straight binary etc. and all of them are have different usage of area according to ADC architecture.

There are 4 main ADC architecture: Sigma-Delta (ΣΔ) ADC, Successive Approximation Register (SAR) ADC, Pipeline ADC, Flash ADC. Each ADC structure has different types of sampling rate and resolution specifications that cover particular application areas. Figure 2.5 shows sampling rate and resolution specifications comparison of ADC structures together with some application areas (Kester, 2007).

Amongst to other types, the successive approximation register ADC is the most famous ADC architecture. For medium to high resolution (from 8 to 20 bits) with sampling rates up to several MHz (10 MHz) SAR ADCs are commonly used architecture with very low power consumption (Maxim Integrated, 2002a). Digital output data is mainly transferred via a standard serial interface (I^2C or SPI).

2.2 ADC Characterization

Nowadays, there are many various types of data converters that are designed to respond to requirement fro different categories of application area at the market. All of them offer solution for distinctive engineering problems, and the resolution number is not
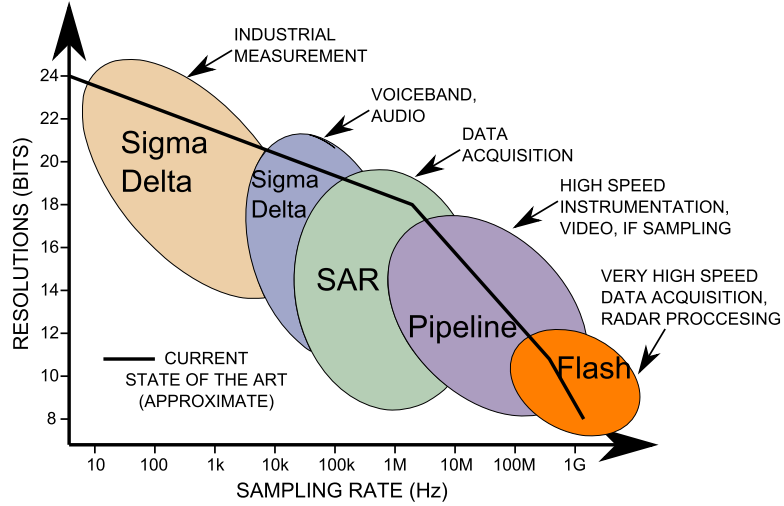
Figure 2.5 : Analog to digital converter types and specifications.

only their distinguished parameter. A data converter has static and dynamic parameters that are use to describe performance of it.

Static parameters are related to the input-output characteristic of a data converter and they can be determined by applying constant voltage to the input terminal and observing the output. Dynamic parameters are related to frequency response and speed of circuits that are used in data converter architecture. In the next section, static and dynamic parameters of data converters are explained in detail.

2.2.1 Static parameters

The relation between the input and the output defines the static behavior of a data converter. In order to determine static behavior of a data converter, digital data at the output is examined in return for constant input voltage. An ideal data converter satisfy the equation;

$$V_{IN} = \frac{V_{FS}}{2^N} \sum_{k=0}^{N-1} b_k 2^k + \varepsilon \tag{2.8}$$

where V_{IN} is the analog input voltage, V_{FS} is the full scale voltage, N the resolution, b_k are the output bits and ε is quantization error. The equation leads a staircase characteristic with uniform steps over entire full scale. Fig. 2.6 shows the ideal characteristic of a 3 bit data converter. The difference between the straight line and the staircase characteristic is defined as quantization error.

Figure 2.7 shows quantization error for a 3bit ADC (Maloberti, 2007). It is varying between $\Delta/2$ and $-\Delta/2$ inside the dynamic range. When the input signal exceed

9
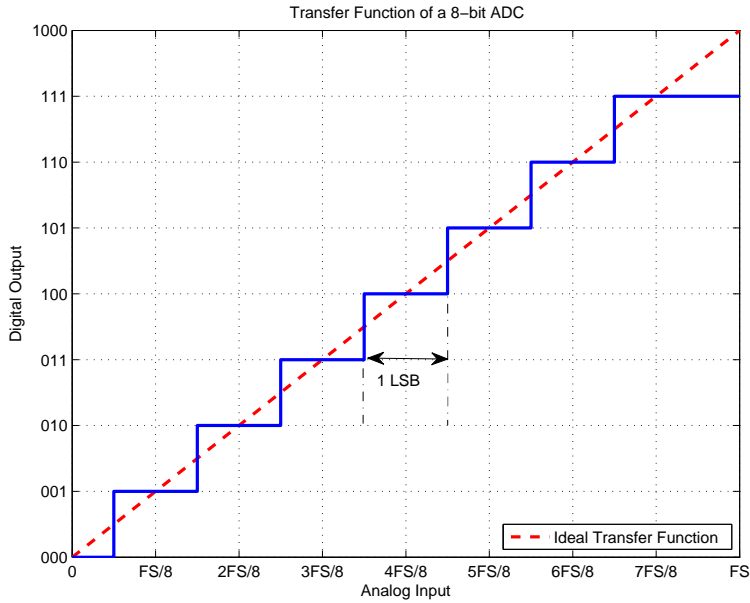
Figure 2.6 : Ideal input-output characteristic of a data converter for 3 bit
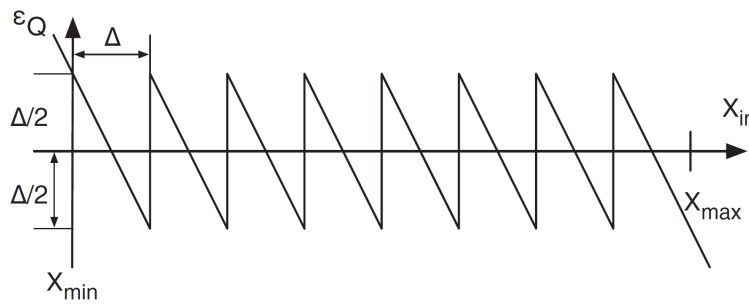
Figure 2.7 : Quantization error for a 3bit converter.

dynamic range, quantizer reaches saturation limits and quantization error increase linearly according to the limit value.

Static parameters of an ADC are listed and defined below.

- Resolution: is the number of bits that are obtained at the output of an ADC. An ADC with higher resolution results more accurate representation of analog signal at the output.

- Analog Resolution: is the smallest perceptible voltage at the input which is also called as LSB voltage. For a 10-bit ADC with a full scale voltage $1V$ results, $1LSB = (1V/2^{10}) = 976\mu$V, LSB voltage.

10
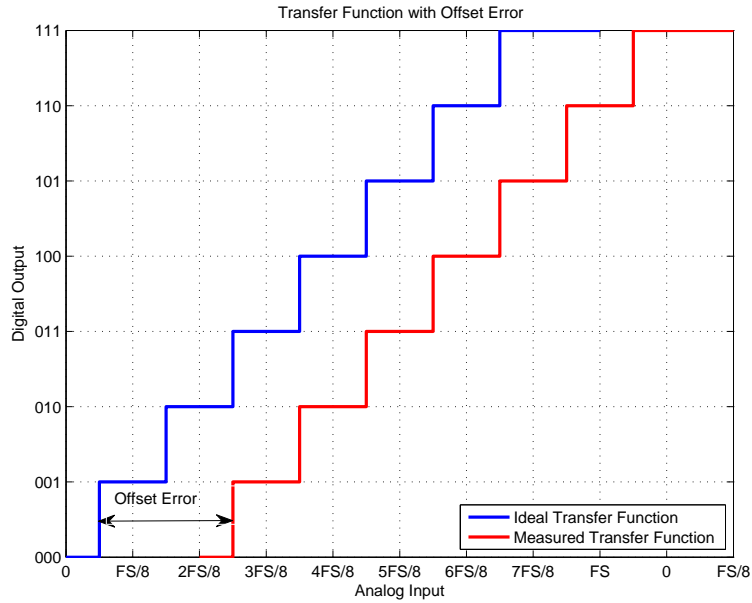
Figure 2.8 : ADC Characteristic with Offset Error

- Offset Error: is a shift of the ADC transfer function over entire full scale for zero. So that, all quantization steps are shifted by offset value and it can be expressed in *LSB* or voltage. Figure 2.8 shows how offset error alter the ideal characteristic.

- Gain Error: is the deviation in the slope of the straight line on the transfer curve. The slope is ratio between D_{FS} and X_{FS} which are full-scale digital code and full-scale analog range respectively. For an ideal ADC, the slope is equal to 1. Figure 2.9 shows how gain error effects the ideal characteristic.

- Full Scale Error: is the difference between the last code transition and ideal top transition which can be determined as the sum of the gain error and the offset error equals to the full scale error.

- Differential Non-linearity(DNL) Error : is the deviation of quantization step for any two successive codes. In ideal, any two successive codes should vary by one LSB width and in this case DNL error is zero. Assume that V_D is the analog voltages which leads us the digital output code D and N is the resolution. DNL is calculated as:

$$DNL = (V_{D+1} - V_D)/V_{LSB-IDEAL} - 1 \qquad \textbf{(2.9)}$$

11
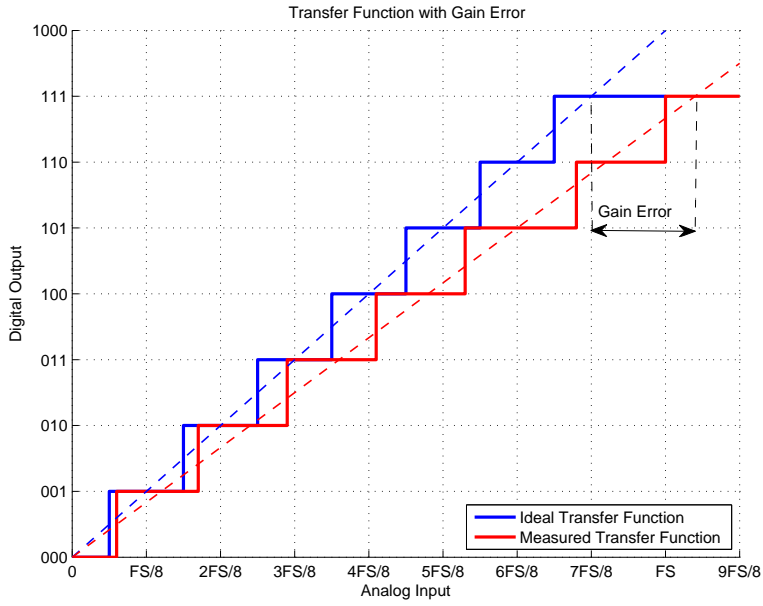
Figure 2.9 : ADC Characteristic with Gain Error

where *DNL* is between 0 and $2^N - 2$. Figure 2.10 shows DNL error for an ADC. Generally, maximum DNL over entire dynamic range is indicated as DNL error. It limits the ADC's signal to noise ratio and spurious free dynamic range.
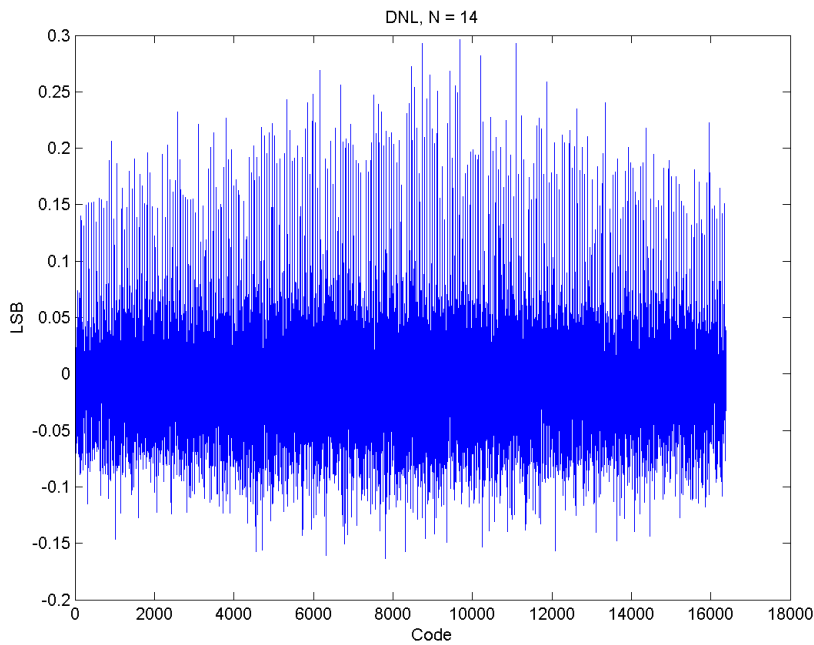
Figure 2.10 : DNL Error of an ADC

12

- Monotonicity: is the correlation between the input analog signal and the digital output as the same direction. If the DNL error is always smaller than $\pm 1 LSB$, a converter is assured monotonic.

- Missing Codes: are output digital codes that are not appear at the ADC transfer function for any analog input voltages. If an ADC has missing code at the output, the corresponding quantization step will be zero. Thus, DNL error for missing code will be -1.

- Integral Non-linearity: is the deviation of the actual transfer function from the ideal interpolating line (Baker, 2006). Also, INL is calculated by excluding effects of the gain and the offset error on the transfer function. The second approach is known as the endpoint-fit line and generally it is revealed in data-sheets. Assume that V_D is the analog voltages which leads us the digital output code D, V_{OFFSET} is offset voltages and N is the resolution. INL is calculated as:

$$INL = [(V_D - V_{OFFSET})/V_{LSB-IDEAL}] - D \qquad \textbf{(2.10)}$$

where D is between 0 and $2^N - 1$. Figure 2.11 shows INL error for an ADC. Generally, maximum INL over entire dynamic range is indicated as INL error.
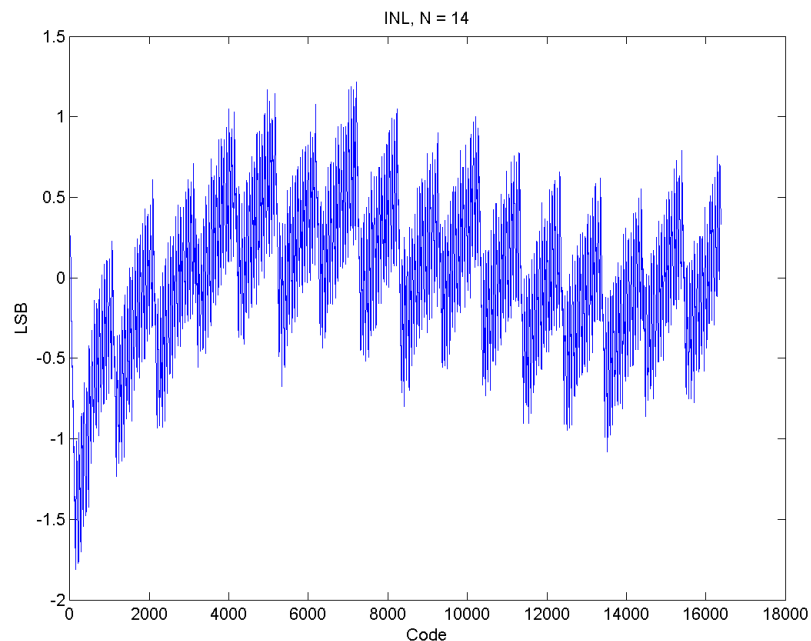
Figure 2.11 : INL Error of an ADC

13

2.2.2 Dynamic parameters

An ideal data converter accepts input signal bandwidth from 0 Hz to infinity and converts that input signal simultaneously. However the frequency response and the speed of analog parts of a data converter are limited and that limitation determines its dynamic performance. If an ADC is designed to operate at high conversion rate or to accept input signal with high bandwidth, dynamic parameters become vital. In addition, a dynamic parameter should remain constant for certain frequency interval. The dynamic parameters of an ADC are listed and defined below.

- Analog Input Bandwidth: is the frequency value that use to determine input signal bandwidth limitation. After the analog input bandwidth, digital output data has decreased by more than 3 dB.

- Aperture Delay : is the delay time between the trigger time of the sampling clock and the time when the input voltage is captured by data converter.

- Aparture Jitter: is the deviation of the sampling time for every clock cycle which results to capture incorrect sample.

- Dynamic Range: is ratio between the highest signal level that the converter can deal with and the noise level. The dynamic range unit is *dB* and it determines the maximum achievable SNR. If the input signal level is higher than the dynamic range, the ADC output is saturated. If the input signal level is very low, it does not detect by ADC.

- Signal to Noise Ratio (SNR: is the ratio between desired signal power and total noise power generated by quantization noise and circuit noise. Total noise power is calculated for the noise within entire Nyquist interval. For an ideal ADC quantization noise is only and unavoidable noise generator. So achievable maximum SNR value is the ratio of main signal power and quantization noise. For a *N*-bit ADC, theoretical maximum SNR is

$$SNR_{dB} = 6.02 \cdot N + 1.76 \tag{2.11}$$

where input signal is a sine wave. Table 2.1 that is generated by using equation (2.11), shows ideal SNR values for different resolution numbers. However an actual

14

ADC has many noise contributor as thermal noise, clock jitter, kT/C noise etc. and therefore, it is not possible to reach theoretical SNR value.

Resolution	SNR
8 bits	49.92 dB
10 bits	61.96 dB
12 bits	74.00 dB
14 bits	86.04 dB
16 bits	98.08 dB

Table 2.1 : Performance summary and comparison table

- Signal to Noise and Distortion (SINAD or SNDR): is the ratio of desired signal power to total noise power including harmonic components that are generated by the input sine wave (excluding dc). Any non-linearity caused by one of the static or dynamic parameter effects the SINAD. That leads us the SINAD is dependent on both the amplitude and the frequency of the input sine wave.

- Effective-Number-of-Bits (ENOB): is another representation of the SINAD using bits. The relation between SINAD and ENOB is

$$ENOB = \frac{SINAD_{dB} - 1.76}{6.02} \tag{2.12}$$

which is obtained from equation (2.11).

- Harmonic Distortion (HD): is the ratio between the root-mean square(rms) of the signal and the rms sum of harmonic components. Generally, first ten harmonic components are used for calculations. After the tenth harmonic terms, harmonic components become insignificant. Assume that f_{in} and f_s are the input signal frequency and the sampling frequency respectively. N-th harmonic component appears at the $|\pm f_{in} \pm kf_s|$ where k is selected as an appropriate number to put the harmonic component into the first Nyquist zone.

- Spurious Free Dynamic Range (SFDR): is the ratio of rms value of the fundamental signal amplitude to the next highest peak rms value in the first Nyquist zone. The highest spur can be generated at the harmonic components frequency due to non-idealities in the ADC or clock reference. Figure 2.12 shows SFDR measurement for 3Hz signal

15

Figure 2.12 : SFDR definition on FTT graph
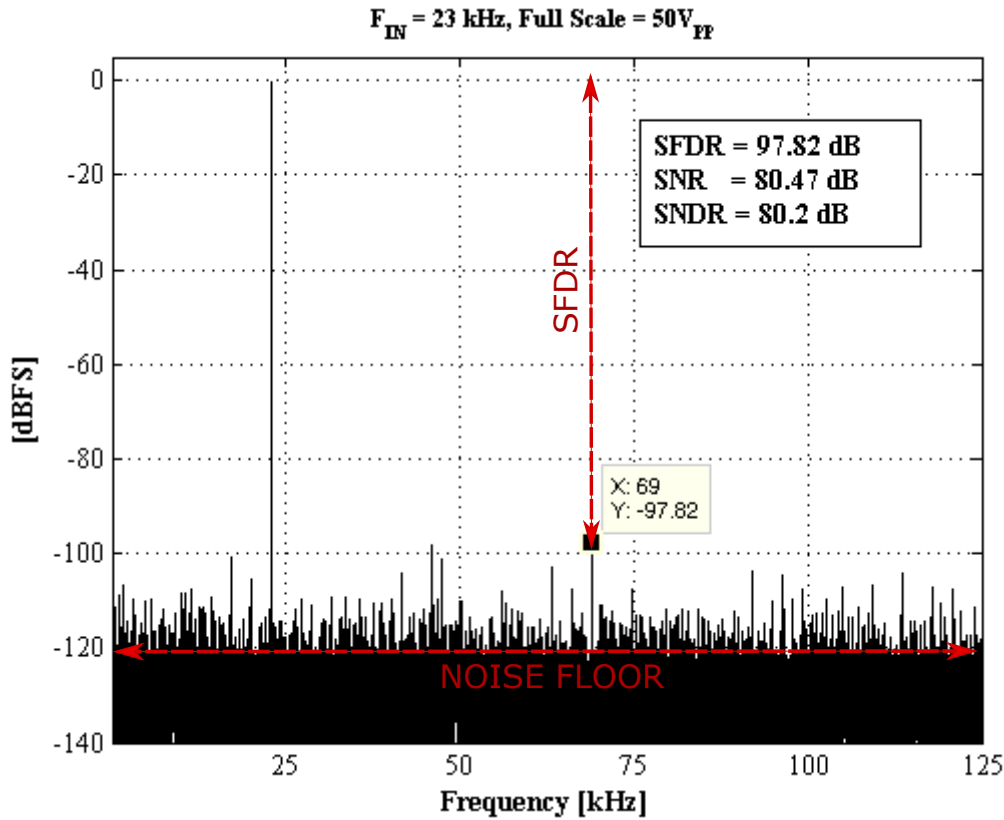
- Inter-modulation Distortion (IMD): is occurred because of non-linearity in the ADC or when the input signal is a multi-tone signal. The ADC produces new component on the spectrum which has frequency different than input signal. So, it is the ratio between picked inter modulated components to the fundamental signal power of f_1 and f_2.

- Effective Resolution Bandwidth (ERBW): is a parameter that determines the maximum signal frequency that ADC can process. The analog input signal is increased to the frequency at which SINAD has decreased 3dB.

$$FoM = \frac{P_{Tot}}{2^{ENB} \cdot 2BW}$$

(2.13)

- Figure of Merit (FoM): is a parameter that have used to evaluate and compare ADC performance. There are many figure of merit definitions in literature and data-sheets. However, equation (2.13) is one of the most common definition, also accepted by IEEE, that obtains total power consumption (P_{Tot}), effective number of bits (ENB) and bandwidth of converted signal (BW) as parameters.

16

Static and dynamic parameters are defined to determine the performance of an ADC. Each of parameters are important for different application area to designate another requirement. However, a single parameter is not sufficient to characterize the performance of the ADC. As a result, static and dynamic parameters must be handle as a whole while characterizing the ADC.

3. MEASUREMENT OF THE SAR-ADC

3.1 SAR-ADC Arcitecture

The basic block diagram of SAR-ADC is shown in Figure 3.1 (Maxim Integrated, 2001c).
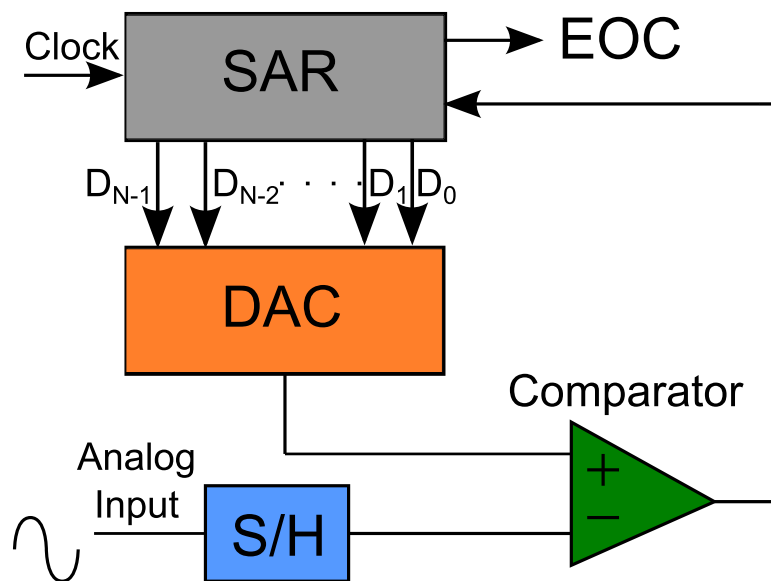
Figure 3.1 : Block diagram of generic SAR-ADC architecture.

It consists of 4 block as sample and hold circuit (S/H), comparator, SAR logic and DAC. At the beginning, an analog input signal is sampled by a sample and hold circuit in order to keep the signal unchanged during the conversion phase. At beginning of the conversion, the internal DAC is set to mid-scale code. Then, the comparator decides whether the sampled input signal is greater or less than the DAC output, and the first approximation is preserved in the successive approximation register. In the next step, the DAC is set to new value depending on the first approximation and SAR block determines next bit value according to comparator output. The operation maintains until the all bit values have been acquired. At the end of the conversion (EOC) the DAC output almost reaches the sampled input signal. Figure 3.2 shows how DAC output changes according to the sampled input signal where FS is full scale voltage

19

and t is clock period.
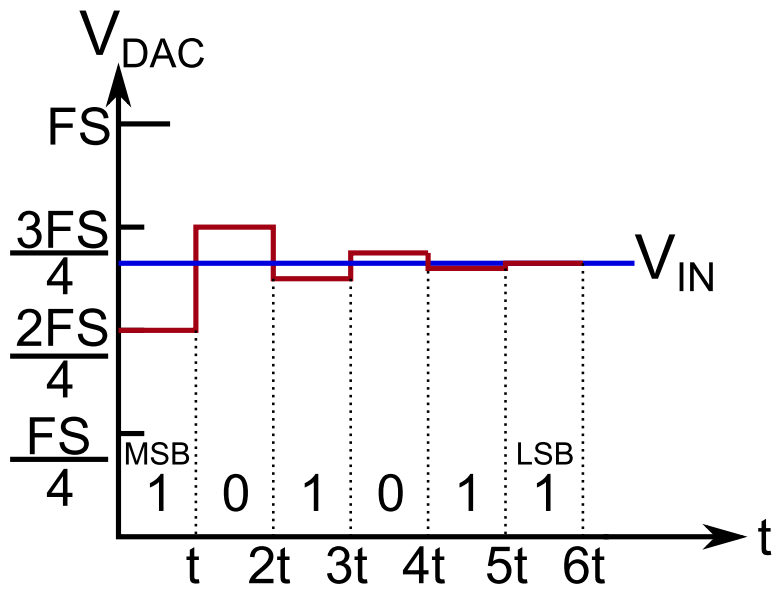
Figure 3.2 : Binary search algorithm

Notice that, first determined bit is the most significant bit and last determined bit is the least significant bit. Also, a single conversion requires N clock period to make N approximation.

3.2 Two-Step SAR-ADC Architecture

The fully-differential two-step SAR-ADC structure is shown in Figure 3.3.
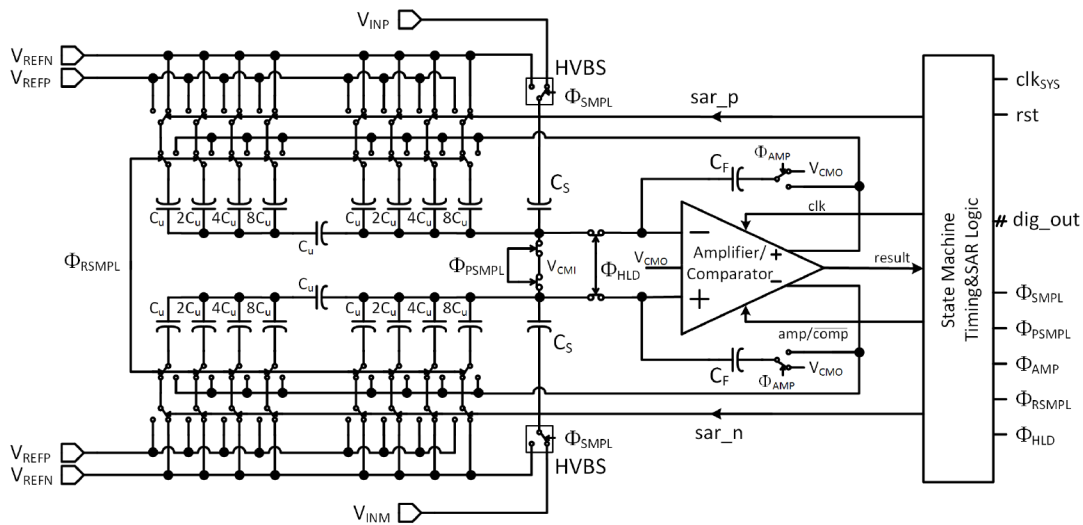
Figure 3.3 : Fully-differential two-step SAR-ADC architecture

20

There are 5 operation state of two-step SAR-ADC as input sampling, first 8bit SAR conversion, residue amplification, residue re-sampling and second 8bit SAR conversion. The operation starts with high voltage input sampling onto the metal-insulator-metal (MIM) sampling capacitor C_S by using the high voltage bootstrapped switch (HVBS). Since total charge on C_S is redistributed on the capacitive DAC (CDAC), the internal circuit is protect from being exposed to high voltage signal (Ozkaya, Gurleyuk, Ergul, Akkaya, & Aksin, 2014). Next, the first 8bit conversion is take part to generate the most significant 8bit. During the first 8bit conversion the amplifier/comparator is in comparator configuration so that it can perform SAR algorithm. At the end of the first 8bit conversion, the residual charge, which is smaller than 1LSB voltage if the conversion is realized without any mistakes, is left on the CDAC array. In the amplification state, the residual charge is amplified by the inter-stage gain of 64, once the feedback capacitor C_F provides a feedback path between the input and the output of the amplifier. Following the amplification state, in the resampling state, the amplified residue signal at the amplifier output is re-sampled on the CDAC array where the amplifier is reconfigured as a buffer for driving the amplified residue signal (Ozkaya et al., 2014). Finally, the least significant 8bit is generated in the second 8bit conversion where the amplifier switches to comparator mode again. After all, the digital error correction block combines the most significant 8bit and the least significant 8bit according to the inter-stage gain value and extracts the 14bit digital data, while the next input data is sampled on C_S in order to repeat all process again. According to these five states, the flow diagram of the ADC is drawn in Figure 3.4.

The states of flow diagram are explained below.

WAIT : The state machine does not switch condition until the "START" signal is up.

SAMPLE : The state machine does not switch condition until the counter counts down from "Stime" to zero. The input signal is sampled on sampling capacitor as a charge.

SAR 1 : The state machine does not switch condition until the counter counts down from "Ctime" which is 8 to zero. The most significant 8 bits are acquired at the end of the first SAR conversion.

AMPLIFIY : The state machine does not switch condition until the counter counts down from "Atime" to zero. The voltage that is originated from residual charges after the first SAR conversion is amplified during the state.

21
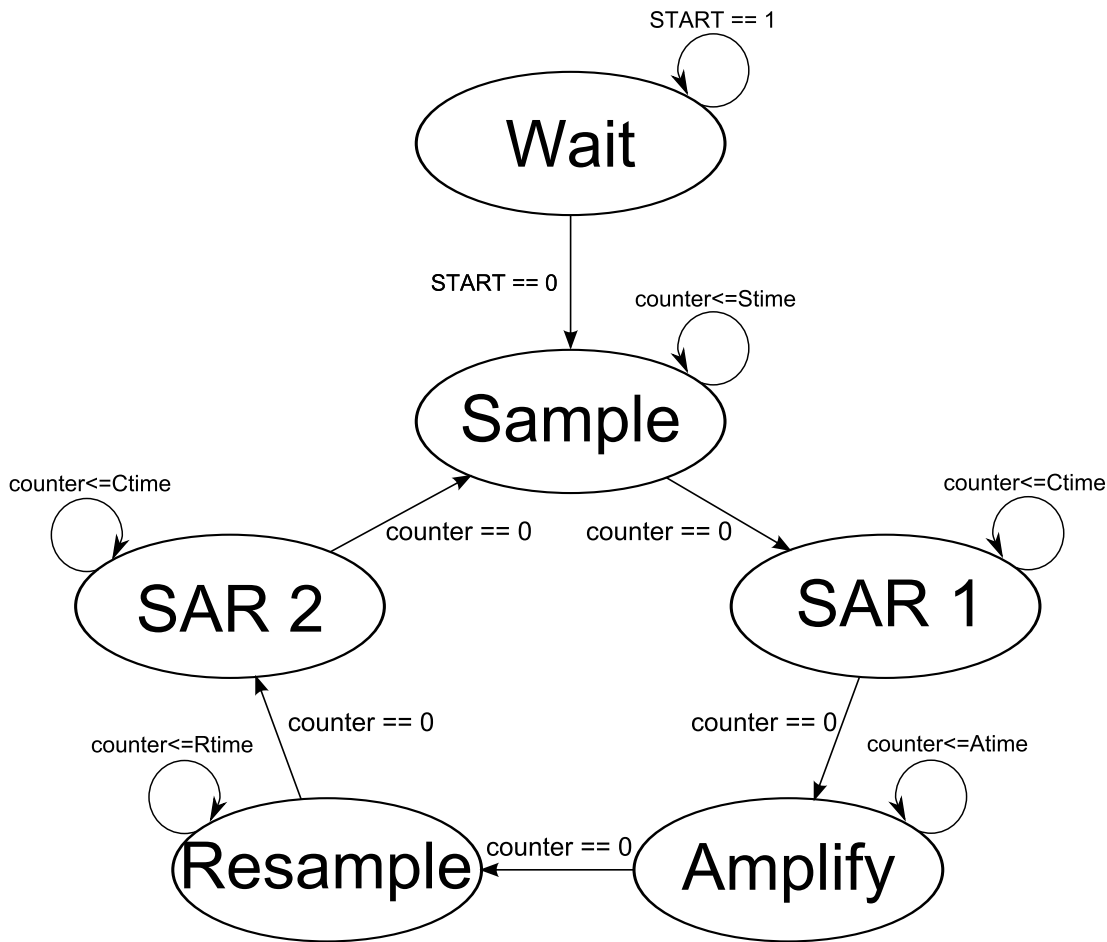
Figure 3.4 : Flow diagram of SAR ADC

RESAMPLE : The state machine does not switch condition until the counter counts down from "Rtime" to zero. The amplified residual voltage is sampled on capacitive DAC.

SAR 2 : The state machine does not switch condition until the counter counts down from "Ctime" to zero. The least significant 8 bits are acquired at the end of the second SAR conversion.

Each state in the flow diagram has a specific time length. A counter counts down clock signal cycle at each state until it reaches zero. In every transition, new value of time length is loaded from a register to the counter and the counter starts to count down all again. The time length of the states can be set through IC2 except conversion time, "Ctime". Each conversion time ("SAR1" or "SAR2" state) is accomplished exactly in every 8 clock cycle. Because it is deterministic and the value of conversion time is the same as number of bits that is obtained during conversion states. The rest of
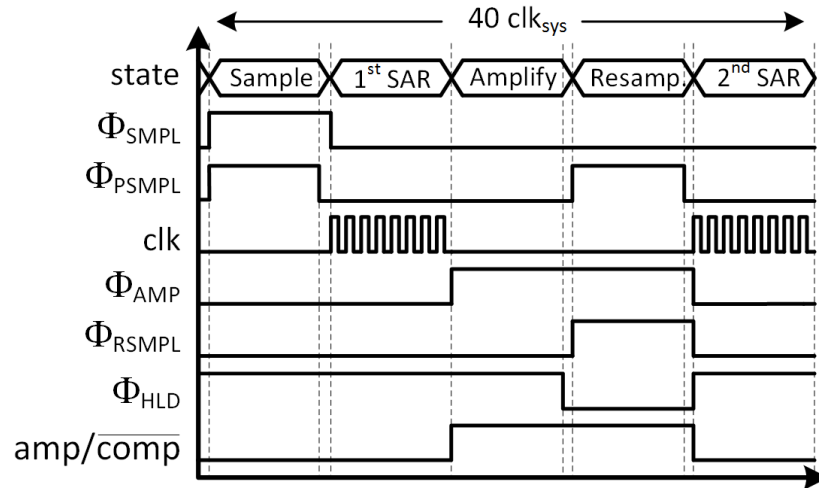
Figure 3.5 : Timing diagram of SAR ADC

the time lengths are selected in order to optimize the ADC performance. Assume that the settling time of the amplifier is 50 ns and clock signal frequency is 100 MHz, the amplification state requires 5 clock cycles to satisfy the settling time of the amplifier. Then "Atime" is set to 5 via IC2. However, the amplification time is not exact same for every chip by the reason of process drifts. If the amplification time is much longer or much shorter than its nominal value, the amplification state requires different clock cycle value than its nominal value which is 5 clock cycles in this assumption. In order to optimize the settling time, "Atime" is set to a new value via IC2. The timing diagram of the two step SAR-ADC is shown in Figure 3.5. According to the diagram, a single conversion demands 40 clock cycle, 24 for settling time of the sampling, amplification and resampling states and 16 for the two SAR conversion.

The flow diagram of SAR algorithm is shown in Figure 3.6. During the SAR1 and SAR2 states, successive approximation register (SAR) block is enabled to generate 8 bit digital data. At the initial condition, 8 bits digital data applied to the capacitive DAC array is '10000000' which is mid-code for the 8 bit DAC array. When the SAR conversion is begun, the comparator decides the 1st bit value of the digital data by comparing the input signal voltage and the capacitive DAC array voltage. If the first approximation is positive (Comp=1), the first bit, the most significant bit in this case, is set to 1 and next significant bit is set to 1 resulting a new approximation as $'11000000'$. If not (Comp=0), first bit is set to 0 and next significant bit is again set to 1 resulting
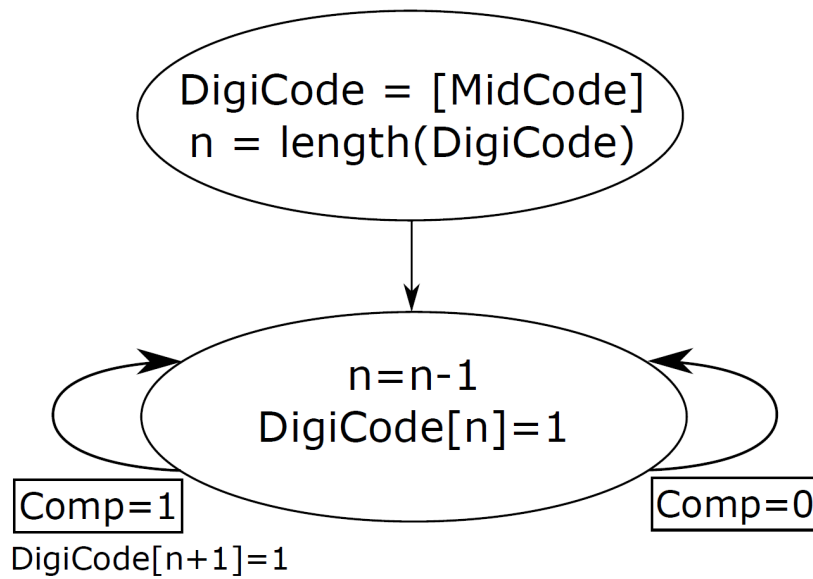
Figure 3.6 : Flow diagram of SAR alogrithm

a new approximation as $'01000000'$. The next digital bit value is determined at each clock cycle according to flow diagram until 8 approximation is made for 8 bits. 3.6.

3.3 Measurements

3.3.1 Evaluation board of the SAR-ADC

The evaluation board of SAR-ADC is shown in Figure 3.7.

The terminals of the board are marked with a red square on the figure 3.7. Since the ADC structure is symmetrical, the trace of related pins(as V_{INP}-V_{INN}) are drawn as a pair. In this way, noise coming from the other pins affects similarly on these pairs. Also, the analog and digital sides are isolated from each other to prevent noise coupling from digital blocks to analog blocks. The digital terminals are placed on the upper right side of the board and the analog terminals are placed on rest of the board. All of the connections on the board and their description are summarized in table 3.1.

3.3.2 Instrument selection

The performance of the ADC is limited to measurement instruments that will be used to characterize ADC performance. In order to determine characteristic of the ADC properly and accurately, every instrument in the measurement setup must be chosen carefully. The selection of analog input signal, reference voltage, clock signal source
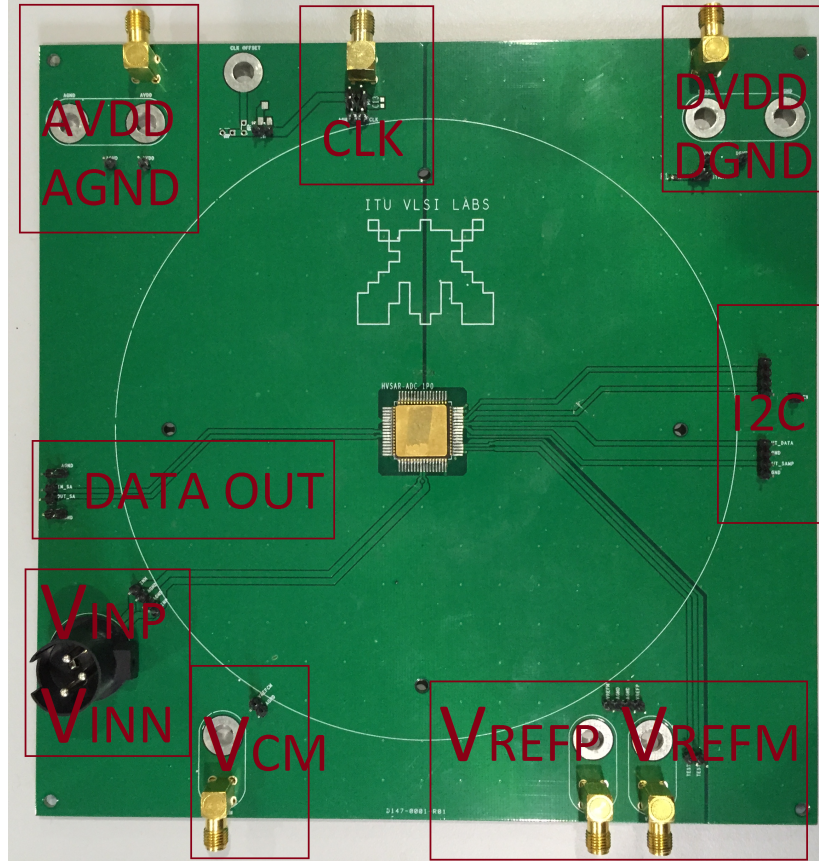
Figure 3.7 : The evaluation board of SAR-ADC

Table 3.1 : Pin description

Pin Name	Description
AVDD	Analog supply voltage
AGND	Analog ground
DVVD	Digital supply voltage
DGND	Digital ground
VINP	Analog input voltage
VINN	Analog input voltage
VCM	Input common mode voltage
VREFP	Positive reference voltage
VREFN	Negative reference voltage
I2C	I2C connection
CLK	System clock signal
DATA OUT	Serial data output

and the relation between input signal frequency and clock frequency are important facts about measurement setup.

The ADC performance is mainly correlated with the analog input signal quality. Therefore, the most sensitive pins on the evaluation board are V_{INP} and V_{INN} where analog input signal is applied. The resolution and linearity of input signal must be as

high as possible, so that the ADC performance is not limited by input signal source. There are many signal sources that can provide a signal with high resolution and linearity at lower amplitude but it is not easy to generate a $50V_{pp}$ sine wave with high resolution and linearity. By the reason of that, a highly linear input signal source, Audio Precision SYS-2722, is used as a signal source to generate desired analog input signal for the ADC. It is possible to generate a 24bit signal from 100Hz to 24kHz by Audio Precision SYS-2722. However, the mentioned SNR value is defined for a $1V_{p}p$ signal on 0V common mode voltage. Therefore resolution of the signal will be less than 24bit. The frequency spectrum of $32V_{p}p$ analog input signal at 20kHz generated by using Audio Precision SYS-2722 is shown in Figure 3.8.
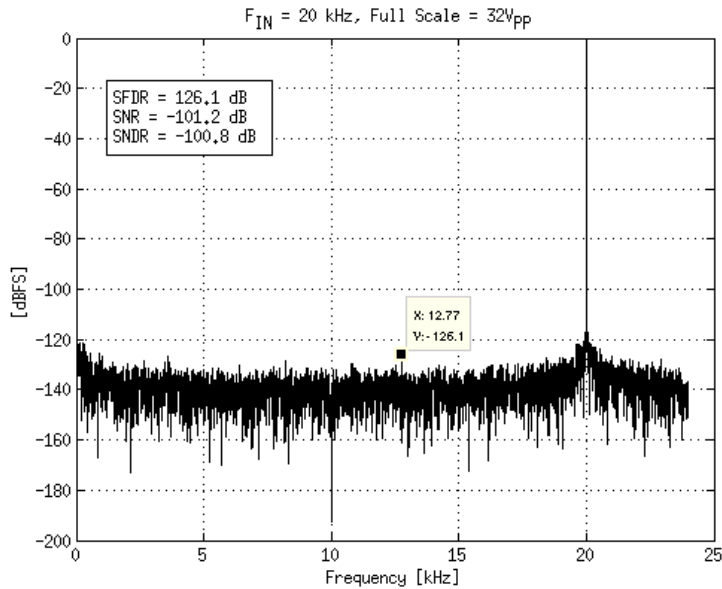
Figure 3.8 : Output frequency spectrum of Audio Precision for $32V_{pp}$

As it seen from figure 3.8, the noise floor of the signal is at the level of -140dB and SFDR value is at the level of 126.1dB. According to the measurement results, Audio Precision SYS-2722 is sufficient to characterize a 16bit ADC. Even though, differential value of the signal is at the $32V_{pp}$, common mode voltage of the signal is at 0V. Since, the minimum signal voltage that is accepted by ADC is 0V, the input signal lower than 0V is not valid input. Therefore, the input signal that is generated by the instrument must have a common mode voltage, ensures that minimum signal level is always higher than 0V. If the common voltage is set to $V_{pp}/2$, the above condition is satisfied. Generally, the performance of an ADC is also limited by a voltage reference in some way (Kester, 2007). The differential input signal is exposed the noise contribution of

the input common mode voltage in the same direction. Hence, the differential input signal does not cancel common mode voltage noise and the effect of noise on the input signal directly appears on frequency spectrum. As a result, Agilent B2962A low noise power supply is used to generate the input common mode voltage. Moreover, reference voltages of the ADC, V_{REFP}-V_{REFN}, are also sensitive to noise coming from the power supply because in every SAR conversion bottom terminal of the DAC array is set to V_{REFP} or V_{REFN} with respect to digital code. So that, DAC linearity is affected by reference voltage directly. As a consequence, Agilent B2962A low noise power supply is also used to generate reference voltages V_{REFP} and V_{REFN}.

Apart from reference voltages, clock signal of the ADC is one of the most critical source in the system because dynamic performance of the ADC is affected by the timing accuracy of the clock signal (King, 2011). Thus, it is very important to use system clock with low jitter or phase noise. The total jitter requirement of the system is calculated from the below formula

$$T_{jitter(rms)} = \frac{V_{IN}}{V_{FS}} \times \frac{1}{2^{(N+1)} \times \pi F_{IN}} \qquad (3.1)$$

where V_{IN} peak to peak input voltage, V_{FS} ADC full scale voltage, F_{IN} input signal frequency and N ADC resolution. If input signal voltage is selected to equal the full scale range, then total jitter requirement become factor of the resolution bits number and the input frequency. For a 14bit ADC that has 20kHz input signal, calculated rms jitter is 485ps. It is possible to calculated maximum achievable SNR value for different jitter values by using equation (5.2). Figure 3.9 illustrates SNR and ENOB alteration by jitter for full scale input voltages at different frequency (Kester, 2007).

Agilent 33250A Function/Arbitrary Waveform Generator is used to generate clock signal with jitter lower than 75pS. According to figure 3.9, it is possible to reach 83dB SNR for a full-scale input signal at 1MHz.

Coherent sampling is a technique which greatly increases the spectral resolution of an FFT and eliminates the requirement for windowing when the proper ratio between input frequency and sampling frequency is satisfied (Maxim Integrated, 2002b). It is the most common method for characterizing the dynamic performance of ADCs. For a sine wave that is formed 2^N number of data points requires the use of input signal
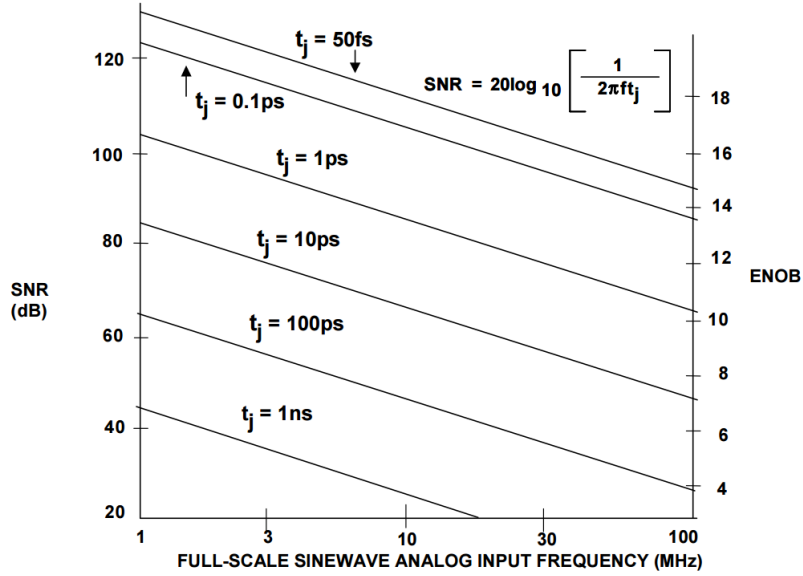
Figure 3.9 : Maximum achievable SNR and ENOB due to jitter vs. Input Frequency

frequency f_{in}

$$f_{in} = \frac{k}{2^N} f_s \qquad (3.2)$$

where f_s is the sampling or clock frequency, k is an integer number of clock cycles and 2^N is the number of data points in FFT analysis.
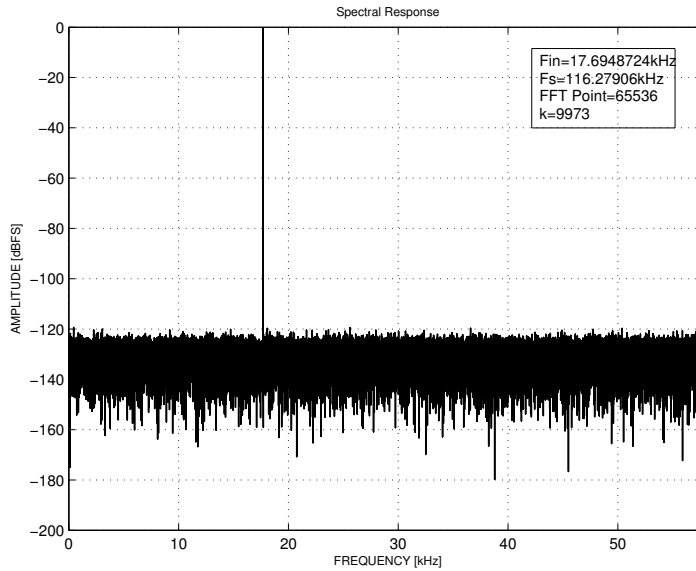
Figure 3.10 : FFT results of coherently sampled data stream

Figure 3.10 shows FFT results of an ADC output that is generated by using coherent sampling method. The circumstance for coherent sampling were satisfied by selecting $f_{in} = 17.6948724\text{kHz}$, $f_s = 116.27906\text{kHz}$, $k = 9973$ and $N = 16$. If number of the

28

clock cycles is a prime number, repetitive patterns in the output stream is prevented.
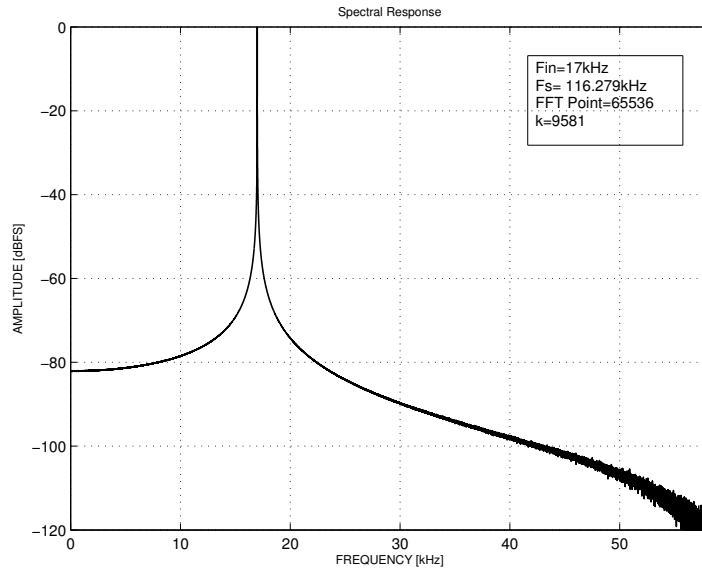
Figure 3.11 : FFT results of non-coherently sampled data stream

Figure 3.11 shows FFT results of an ADC output that is generated by using non-coherent sampling method. This time input frequency is changed to 17kHz and as a result the effects of spectral leakage that is caused by non-coherent sampling is observed.

It's hard to achieve coherent sampling in real life with two different instruments because both of the instruments generate output signal with respect to own internal crystal oscillator. Thus, as an external reference Rubidium Frequency Reference that can generate very accurate reference signal is used for locking the input signal source and the clock signal source output to the same reference. By this way, it is possible to achieve coherent sampling for the measured ADC.

3.3.3 Measurement results

Figure 3.12 shows the measured frequency spectrum of the ADC output for a $50V_{pp}$ input signal on a 15V common mode voltage at 23kHz and sampling frequency of the ADC is 10MHz. The input signal frequency is limited by the selected input signal source and the peak-to-peak input signal voltage is limited by HVBS. The determined SNR and SFDR values are 80.2dB and 97.8dB respectively.

Figure 3.13 shows the variation of the SFDR, SNDR and SNR with respect to the
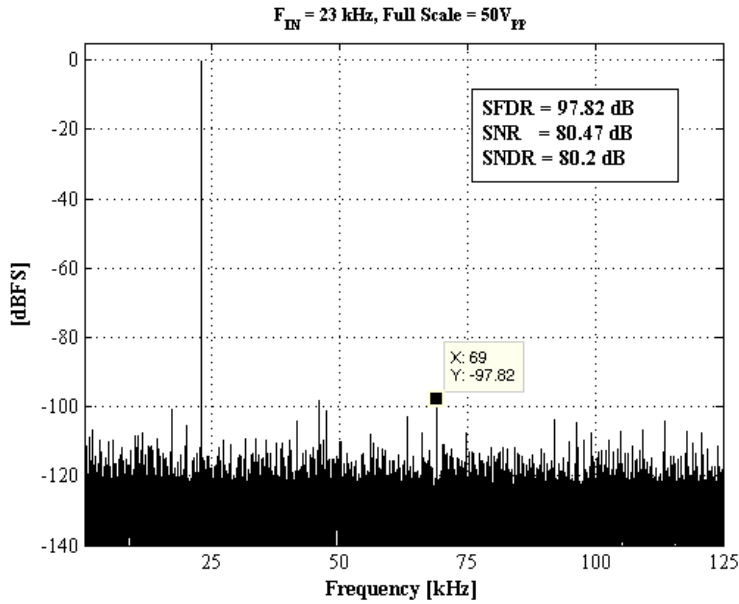
Figure 3.12 : Output frequency spectrum of Audio Precision for $32V_{pp}$

sampling rate and input frequency. First, sampling rate is increased from 50kS/s up to 450kS/s with 10kS/s intervals and SFDR, SNDR and SNR are calculated for every single measurement. Up to 250kS/s sampling rate, variaton of the SFDR, SNDR and SNR are negligible. Beyond 250kS/s sampling rate, the dynamic performance of the ADC starts to decrease but it is not crucial until 350kS/s sampling rate. To obtain other plot in figure 3.13, the analog input signal frequency is swept from 1kHz up to 24kHz and SFDR, SNDR and SNR are calculated again. Note that, input signal frequency is limited by input signal source. As it seen from figure 3.13, dynamic performance of the ADC is almost independent from analog input frequency. There are some slight degradations (smaller than 0.5dB) by the reason of the frequency response of the selected instruments.

Figure 3.14 shows variation of the SFDR, SNDR and SNR with respect to the input signal amplitude. In order to realize measurement in figure 3.14, input signal amplitude is increased from $1V_{pp}$ up to $50V_{pp}$ while sampling rate is fixed at 50kS/s and SFDR, SNDR and SNR are calculated for every single measurement. As expected, SFDR, SNDR and SNR are increased with respect to the input signal amplitude. Recall that from chapter 2, all three parameters SFDR, SNDR and SNR are defined according to signal power at the ADC output. Therefore, if the signal power at the ADC input is increased then the signal level at the ADC output is increase and SFDR, SNDR and SNR are also increased with respect to the signal power at the ADC output. However,
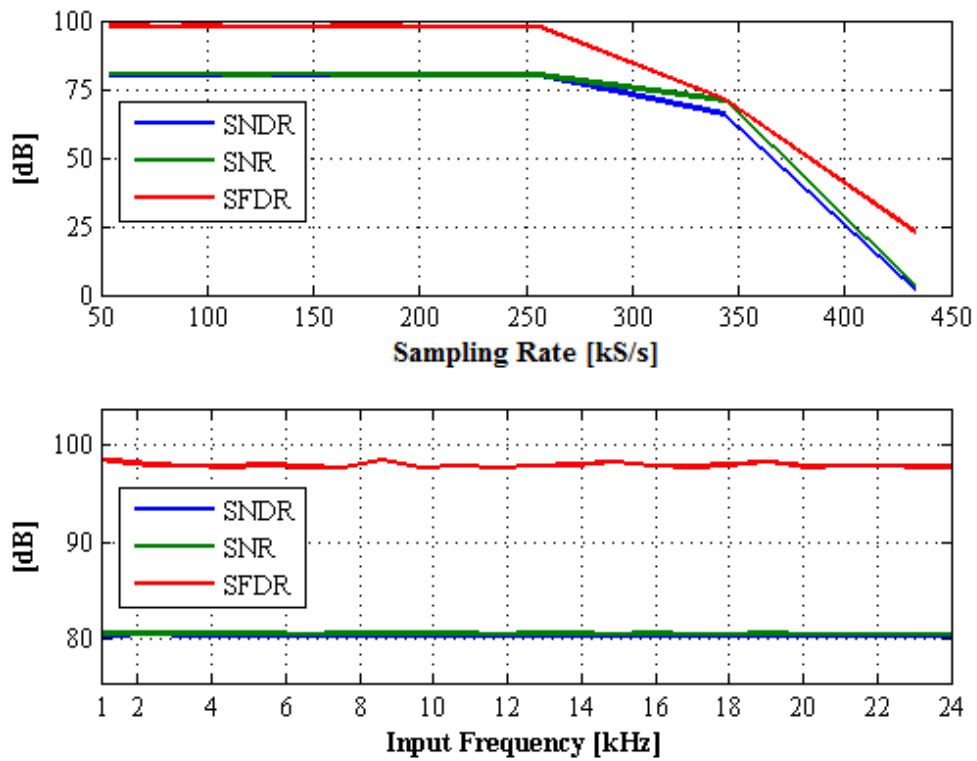
Figure 3.13 : SNR, SNDR and SFDR measurements vs. sampling rate and input frequency

the linearity of the HVBS and the CDAC are varying with input amplitude, so that there is no linear relation between input signal amplitude and SFDR, SNDR and SNR. As the input signal amplitude is increased, linearity of the HVBS and CDAC dominate the ADC output.

A generic setup for testing INL and DNL is applying quasi-DC voltage ramp or low-frequency sine wave as the input signal (Maxim Integrated, 2001b). Both way purpose to generate transfer function of the ADC by hitting every single output code. However it is hard to achieve a signal with that accuracy. Therefore, the histogram method which is explained in (Maxim Integrated, 2001a) is used to verify INL and DNL specifications of the SAR-ADC. In order to hit every digital code, coherency between the input signal and the clock signal is neglected. With sufficient sample size INL and DNL characteristic of the ADC is determined as shown in Figure 3.15 and 3.16. Figure 3.15 show INL and DNL measurement for first 8bit data that is obtained at the end of the first conversion. Maximum DNL is ±0.07LSB and maximum INL is ±0.05LSB. The dominant error occurs from the most significant capacitor mismatch.

31
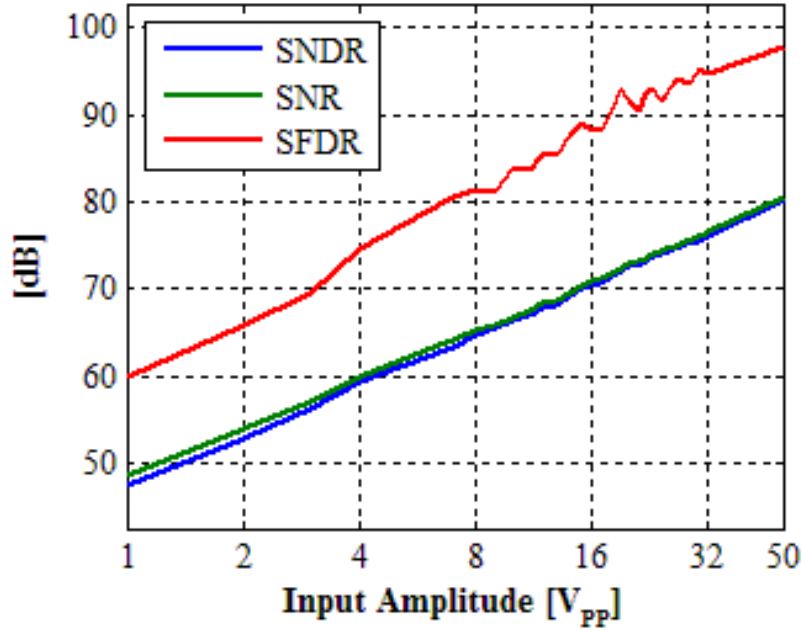
Figure 3.14 : SNR, SNDR and SFDR measurements vs. input amplitude

Figure 3.16 shpws INL and DNL measurement for 14bit data that is obtained at the end of the full SAR conversion. As it expected, primary INL variation results by the reason of the split CDAC structure (Ozkaya et al., 2014). There are 16 specific segments at every 256th code that is related to first 8bit DNL error.

Table 3.2 shows a comparison of the measured ADC with its counterparts by using measurement results. The implemented ADC offers a solution for high voltage data acquisition without using a resistive divider or an additional supply. Input range of the ADC measured as $50V_{pp}$ which is 15.1 times of its supply voltage. As a result the ADC has the best FoM because of low voltage and wide input range.

Specifications	This Work	AD7612[1]	ADS8505[2]	MAX1132[3]	LTC1609[4]
Sampling Architecture	D.S.[5]	D.S.[5]	R.D.[6]	R.D.[6]	R.D.[6]
Input Range (V)	$50V_{pp}$	$40V_{pp}$	$20V_{pp}$	$24V_{pp}$	$20V_{pp}$
Input Load	2pF	12pF	11.5kΩ ‖ 50pF	7.9kΩ ‖ 32pF	13.3kΩ ‖ 10pF
Power Supply (V)	3.3	5& ±15	5	5	5
Resolution (bits)	14	16	16	16	16
Sampling Rate (kS/s)	250	750	250	200	200
SNDR (dB)	80.2	93.5	88	85	87
SFDR (dB)	97.8	107	98	96	94
DNL (LSB)	0.55	1.5	2	1	2
INL (LSB)	1.81	1.5	4	1.5	2
Power (mW)	4.29	190	70	55	65
FOM (pJ/conv-step)	2.05	6.55	13.6	15.1	16.8

Table 3.2 : Performance summary and comparison table
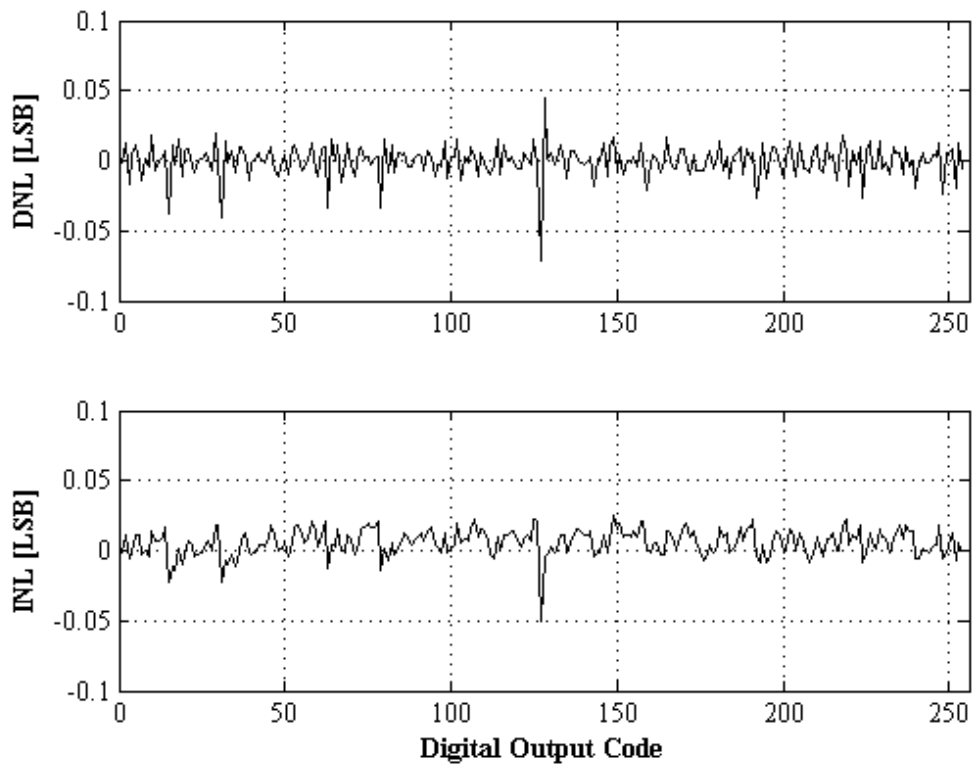
Figure 3.15 : DNL-INL measurement for 8bit output
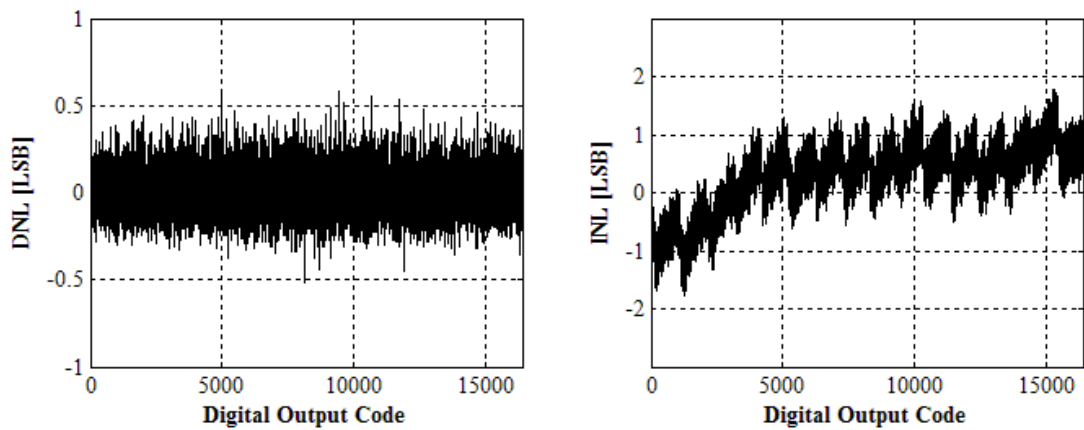
Figure 3.16 : DNL-INL measurement for 14bit output

As consequence, two-step SAR-ADC architecture is measured and characterized with respect to static and dynamic parameters. In order to measure static and dynamic parameters properly, selection of the instruments is critical process. After selecting correct instruments, all measurement are accomplished in LabView environment automatically since every measurement is repetitive.

33

4. TRANSFER FUNCTIONS OF SAR ADC

4.1 Transfer Functions of SAR-ADC

Obtaining transfer functions of ADC is an useful and important action to understand behavior of the implemented architecture in detail. In this section, the mathematical analysis of the selected analog to digital conversion method will be examined. The circuit that is used for to obtain the transfer function is shown in Figure 4.1 (Ilter, 2010). The parasitic capacitors and their effects on the transfer function will also be considered during calculations. In order to simplify mathematical equations, all elements in the Figure 4.1 are accepted as ideal element. Once all of the transfer functions are obtained, ideal elements will be switched into the real element step by step.
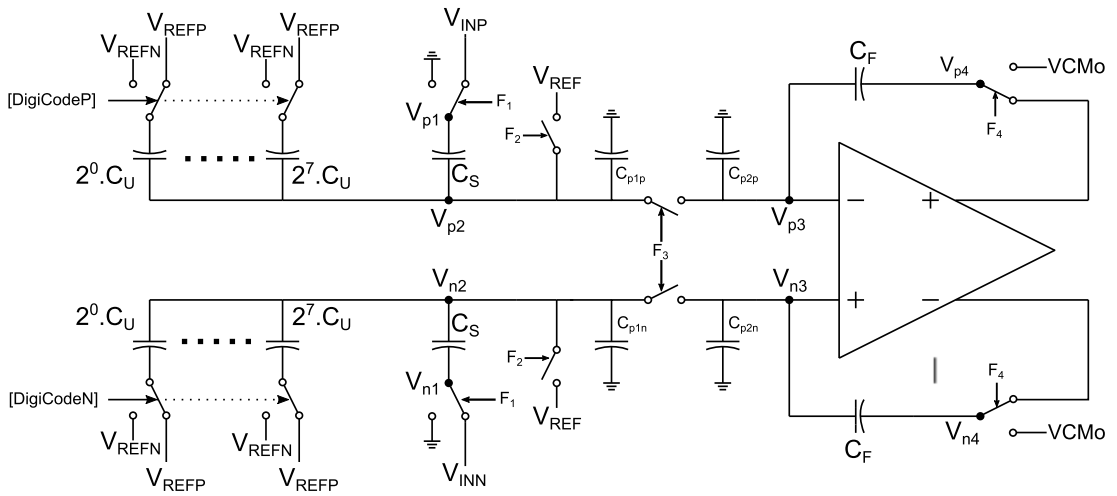
Figure 4.1 : The simplified SAR-ADC architecture

There are four main switches as F_1, F_2, F_3 and F_4 in the Figure 4.1. In every phase of the SAR-ADC, a particular combination of the switches are utilized to provide a different configuration. A single switch is used to implement functionality of F_2 and F_3 switches, however F_1 and F_4 switches are implemented by using two switches to realize represented MUX operation. Additionally, the control signals of the switches F_1,F_2,F_3 and F_4 are indicated as ϕ_1, ϕ_2, ϕ_3 and ϕ_4 respectively.

35

The signals V_{INP} and V_{INN} are high voltage differential analog input signals that are applied to the bottom terminal of the sampling capacitance C_S according to condition of F_1. V_{REFP} and V_{REFN} are DC voltages constituting positive reference and negative reference that are applied to binary weighted capacitive DAC structure with respect to applied digital code. In addition V_{REF} and V_{CMo} are input common mode reference and output common reference voltages.

Also, the circuit that will be analyzed, contains a differential amplifier block which acts as comparator or operational amplifier whether it is in the open loop or closed loop configuration.

The capacitances in the circuit are the sampling capacitance C_S and the feedback capacitance C_F. According to the switches condition, the bottom terminal of the sampling capacitance (C_S) is connected to analog input signal or ground. If the bottom terminal of the feedback capacitance (C_F) is connected to the output of the differential amplifier, the amplifier is used in amplifier configuration else the amplifier is used in the comparator configuration. The other capacitances in the circuit are C_{p1p}, C_{p1n}, C_{p2p} and C_{p2n} which are the parasitic capacitances on the high impedance nodes. Other capacitances are not important at all since they are connected to the low impedance nodes.

For simplicity, the CDAC structure is modeled as a binary weighted capacitor matrix. Every capacitance in the capacitor matrix are constituted from a unit capacitance C_U. For integrity, only the least significant and the most significant capacitor in the CDAC is shown in the Figure 4.1 and they are generated by a single unit capacitor and 2^7 piece of capacitors, respectively. In the real circuit the DAC structure is a binary weighted capacitor matrix with an attenuator capacitor in the middle. Moreover, the effect of using different CDAC structure will be analyzed at the next section.

[DigiCodeP] and [DigiCodeN] are 8 bit long binary input vectors that are applied to the switches connected to the bottom terminal of the capacity matrix. The vector [DigiCodeN] is the ones' complement of the vector [DigiCodeP]. Assume that [DigiCodeP] is equal to [1000 0000], then [DigiCodeN] is equal to [0111 1111]. The most significant bit in the input vectors are applied to the maximum valued capacitor in the capacity matrix. According to digital bit value, positive reference

36

voltage (V_{REFP}) or negative reference voltage (V_{REFN}) is applied to the bottom terminal of the related capacitor in the capacity matrix. When the bit value is 0, negative reference voltage is applied to the bottom of capacitor. Note that if the negative reference voltage does not equal to 0, a single input vector is not sufficient for calculations since the input vector contains zeros for negative reference. Therefore, $[DigiCode] \cdot V_{REFP} + [Digi\bar{C}ode] \cdot V_{REFP}$ equation is used in calculations for a single capacitor matrix in order to get effect of non-zero negative reference voltages.

4.1.1 Sampling phase

The circuit configuration of the sampling state is shown in Figure 4.2.
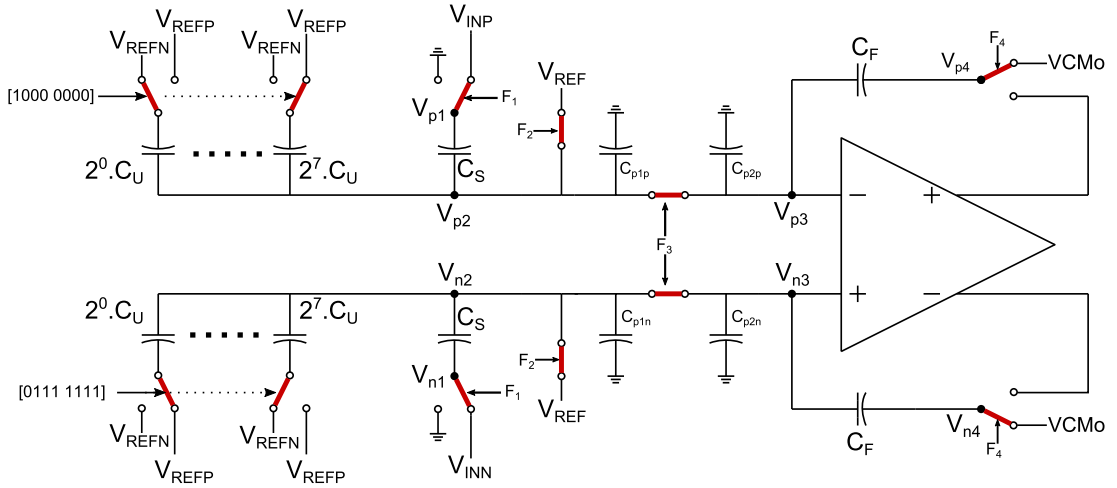
Figure 4.2 : The circuit configuration during sampling phase

In this state, the analog input signals are sampled on the sampling capacitors so that the comparator can make decision in the next phase. At the end of the sampling state, F_2 switch goes to OFF position so that the nodes V_{p3} and V_{n3} are left floating and the sampling state is completed at that moment.

In the sampling phase, the control signals are:

$$
\begin{aligned}
\phi_1 &= 1\,(\text{on})\\
\phi_2 &= 1\,(\text{on})\\
\phi_3 &= 1\,(\text{on})\\
\phi_4 &= 1\,(\text{on})
\end{aligned}
$$

The initial node voltages are:

$$
\begin{aligned}
V_{1p} &= V_{INP} & V_{3p} &= V_{REF}\\
V_{1n} &= V_{INN} & V_{3n} &= V_{REF}\\
V_{2p} &= V_{REF} & V_{4p} &= V_{CMo}\\
V_{2n} &= V_{REF} & V_{4n} &= V_{CMo}
\end{aligned}
$$

37

And the digital code that are applied to the capacitive DAC arrays are:

$$
\begin{aligned}
\text{DigiCodeP} &= [1000\,0000] \\
\text{DigiCodeN} &= [0111\,1111] \\
\text{DigiCodeP} &= \overline{\text{DigiCodeN}}
\end{aligned}
$$

If the total charges on the node V_{2p} and V_{2n} are calculated:

$$
\begin{aligned}
Q_{Sp} =& (V_{2p} - V_{1p}) \cdot C_S + (V_{2p} - [DigiCodeP] \cdot V_{REFN} - [DigiCodeN] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3p} - V_{4p}) \cdot C_F + (V_{2p} - 0) \cdot (C_{p1p} + C_{p2p}) \\
Q_{Sp} =& (V_{REF} - V_{INP}) \cdot C_S + (V_{REF} - [1000\ 0000] \cdot V_{REFP} - [0111\ 1111] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{REF} - V_{CMo}) \cdot C_F + (V_{REF} - 0) \cdot (C_{p1p} + C_{p2p})
\end{aligned}
$$

$$(4.1)$$

$$
\begin{aligned}
Q_{Sn} =& (V_{2n} - V_{1n}) \cdot C_S + (V_{2p} - [DigiCodeN] \cdot V_{REFP} - [DigiCodeP] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3n} - V_{4n}) \cdot C_F + (V_{2n} - 0) \cdot (C_{p1n} + C_{p2n}) \\
Q_{Sn} =& (V_{REF} - V_{INN}) \cdot C_S + (V_{REF} - [0111\ 1111] \cdot V_{REFP} - [1000\ 0000] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{REF} - V_{CMo}) \cdot C_F + (V_{REF} - 0) \cdot (C_{p1n} + C_{p2n})
\end{aligned}
$$

$$(4.2)$$

where Q_{Sp} and Q_{Sn} are total sampled charge on the common node pairs V_{2p}-V_{3p} and V_{2n}-V_{3n}. Also, Bin is a vertical weighted vector that represent the CDAC as follow:

$$
Bin = \begin{bmatrix} 2^7 \\ 2^6 \\ 2^5 \\ 2^4 \\ 2^3 \\ 2^2 \\ 2^1 \\ 2^0 \end{bmatrix}
$$

And the subtraction of a variable from a matrix is calculated as below example:

$$
V_{2p} - [01011001] = [(V_{2p} - 0 V_{2p} - 1)(V_{2p} - 0)(V_{2p} - 1)(V_{2p} - 1)(V_{2p} - 0)(V_{2p} - 0)(V_{2p} - 1)
$$

4.1.2 1st conversion phase

The circuit configuration of the 1st conversion state is shown in Figure 4.3.
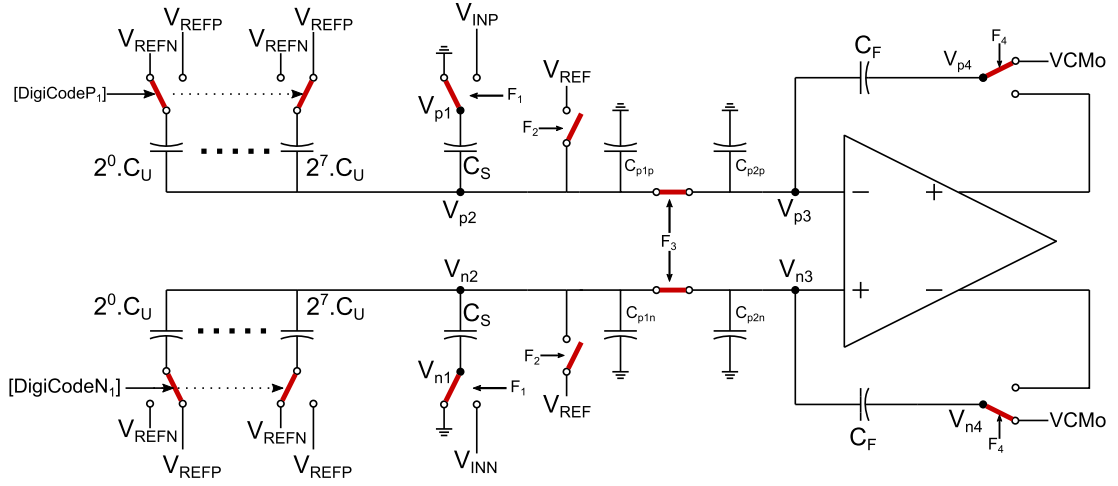
Figure 4.3 : The circuit configuration during 1st conversion phase

In this state, the differential amplifier is in open-loop configuration, so, it is used as a comparator. According to the comparator output, the digital code is acquired by using SAR algorithm. At the end of the first conversion state, 8 bit long digital data is determined.

In the first conversion phase the control signals are:

$$
\begin{aligned}
\phi_1 &= 0(off) \\
\phi_2 &= 0(off) \\
\phi_3 &= 1(on) \\
\phi_4 &= 1(on)
\end{aligned}
$$

If the first conversion is completed without any mistakes, the node voltages are

$$
\begin{array}{ll}
V_{1p} = 0 & V_{3p} = V_{RESP} \\
V_{1n} = 0 & V_{3n} = V_{RESN} \\
V_{2p} = V_{RESP} & V_{4p} = V_{CMo} \\
V_{2n} = V_{RESN} & V_{4n} = V_{CMo}
\end{array}
$$

And the digital codes that are applied to the capacitive DAC arrays are:

$$
\begin{aligned}
DigiCodeP &= DigiCodeP_1 \\
DigiCodeN &= DigiCodeN_1 \\
DigiCodeP_1 &= \overline{DigiCodeN_1}
\end{aligned}
$$

where $[DigiCodeP_1]$ and $[DigiCodeN_1]$ indicate digital codes that are acquired at the end of the first SAR conversion. If the total charges on the nodes V_{2p}-V_{3p} and V_{2n}-V_{2p} are calculated:

$$
\begin{aligned}
Q_{RESp} =& (V_{2p} - V_{1p}) \cdot C_S + (V_{2p} - [DigiCodeP] \cdot V_{REFP} - [DigiCodeN] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3p} - V_{4p}) \cdot C_F + (V_{2p} - 0) \cdot (C_{p1p} + C_{p2p}) \\
Q_{RESp} =& (V_{RESP} - 0) \cdot C_S + (V_{RESP} - [DigiCodeP_1] \cdot V_{REFP} - [DigiCodeN_1] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{RESP} - V_{CMo}) \cdot C_F + (V_{RESP} - 0) \cdot (C_{p1p} + C_{p2p})
\end{aligned}
$$

$$\tag{4.3}$$

$$Q_{RESn} = (V_{2n} - V_{1n}) \cdot C_S + (V_{2p} - [DigiCodeN] \cdot V_{REFP} - [DigiCodeP] \cdot V_{REFN}) \cdot Bin \cdot C_U$$
$$+ (V_{3n} - V_{4n}) \cdot C_F + (V_{2n} - 0) \cdot (C_{p1n} + C_{p2n})$$
$$Q_{RESn} = (V_{RESN} - 0) \cdot C_S + (V_{RESN} - [DigiCodeN_1] \cdot V_{REFP} - [DigiCodeP_1] \cdot V_{REFN}) Bin \cdot C_U$$
$$+ (V_{RESN} - V_{CMo}) \cdot C_F + (V_{RESN} - 0) \cdot (C_{p1n} + C_{p2n})$$

$$(4.4)$$

where V_{RESP} and V_{RESN} are residue voltages at the end of the first conversion.

4.1.3 Amplification phase

The circuit configuration of the amplification state is shown in Figure 4.4.
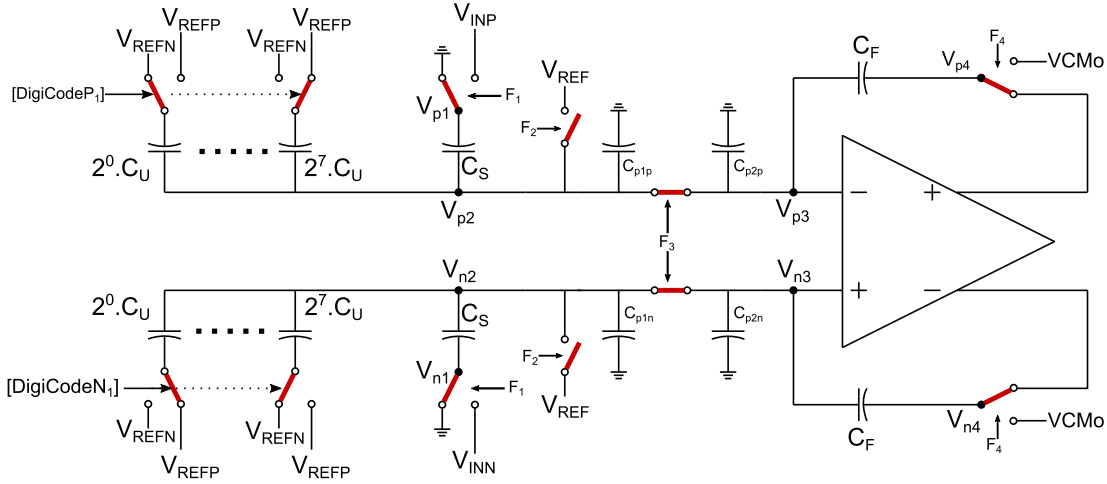
Figure 4.4 : The circuit configuration during amplification phase

The digital code that is determined at the end of the first SAR conversion is kept constant during the amplification state. In the amplification phase, the differential amplifier is used in close-loop configuration so that it can amplify the residue voltage. Moreover, the capacitor C_F provides a feedback path between the input and the output of the differential amplifier and determines the interstage gain.

In the amplification phase, the control signals are:

$$
\begin{aligned}
\phi_1 &= 0(off) \\
\phi_2 &= 0(off) \\
\phi_3 &= 1(on) \\
\phi_4 &= 0(off)
\end{aligned}
$$

The node voltages at the end of the phase are:

$$
\begin{aligned}
V_{1p} &= 0 & V_{3p} &= V_X \\
V_{1n} &= 0 & V_{3n} &= V_X \\
V_{2p} &= V_X & V_{4p} &= V_{AMPP} \\
V_{2n} &= V_X & V_{4n} &= V_{AMPN}
\end{aligned}
$$

And the digital codes that are applied to the capacitive DAC arrays are still:

$$
\begin{aligned}
DigiCodeP &= DigiCodeP_1 \\
DigiCodeN &= DigiCodeN_1 \\
DigiCodeP_1 &= \overline{DigiCodeN_1}
\end{aligned}
$$

In that case the total charges on the common nodes V_{2p}-V_{3p} and V_{2n}-V_{3n} are:

$$
\begin{aligned}
Q_{AMPp} =& (V_{2p} - V_{1p}) \cdot C_S + (V_{2p} - [DigiCodeP] \cdot V_{REFP} - [DigiCodeN] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3p} - V_{4p}) \cdot C_F + (V_{2p} - 0) \cdot (C_{p1p} + C_{p2p}) \\
Q_{AMPp} =& (V_X - 0) \cdot C_S + (V_X - [DigiCodeP_1] \cdot V_{REFP} - [DigiCodeN_1] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_X - V_{AMPP}) \cdot C_F + (V_X - 0) \cdot (C_{p1p} + C_{p2p})
\end{aligned}
$$
(4.5)

$$
\begin{aligned}
Q_{AMPn} =& (V_{2n} - V_{1n}) \cdot C_S + (V_{2p} - [DigiCodeN] \cdot V_{REFP} - [DigiCodeP] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3n} - V_{4n}) \cdot C_F + (V_{2n} - 0) \cdot (C_{p1n} + C_{p2n}) \\
Q_{AMPn} =& (V_X - 0) \cdot C_S + (V_X - [DigiCodeN_1] \cdot V_{REFP} - [DigiCodeP] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_X - V_{AMPN}) \cdot C_F + (V_X - 0) \cdot (C_{p1n} + C_{p2n})
\end{aligned}
$$
(4.6)

where V_{AMPP} and V_{AMPN} are amplified residue voltages at the outputs of the amplifier.

At the end of the sampling phase the switches F_1 and F_2 are turned OFF position and the nodes V_{2p}-V_{2p} and V_{3p}-V_{3n} are left floating. By the charge conservation rule total charge in the sampling state must be equal to total charge in the amplification state. Hence, the charges equation at the (4.1)-(4.5) and (4.2)-(4.6) are equal.

$$
\begin{aligned}
Q_{Sp} &= Q_{AMPp} \\
Q_{Sn} &= Q_{AMPn} \\
\Rightarrow \quad & Q_{Sp} - Q_{Sn} = Q_{AMPp} - Q_{AMPn}
\end{aligned}
$$
(4.7)

If the equations (4.1), (4.5), (4.2), (4.6) are inserted to (4.7):

$$
\begin{aligned}
& (V_{REF} - V_{INP}) \cdot C_S + (V_{REF} - [1000\ 0000] \cdot V_{REFP} - [0111\ 1111] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{REF} - V_{CMo}) \cdot C_F + (V_{REF} - 0) \cdot (C_{p1p} + C_{p2p}) \\
& - (V_{REF} - V_{INN}) \cdot C_S - (V_{REF} - [0111\ 1111] \cdot V_{REFP} - [1000\ 0000] \cdot V_{REFN}) Bin \cdot C_U \\
& - (V_{REF} - V_{CMo}) \cdot C_F - (V_{REF} - 0) \cdot (C_{p1n} + C_{p2n}) \\
& = (V_X - 0) \cdot C_S + (V_X - [DigiCodeP_1] \cdot V_{REFP} - [DigiCodeN_1] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_X - V_{AMPP}) \cdot C_F + (V_X - 0) \cdot (C_{p1p} + C_{p2p}) \\
& - (V_X - 0) \cdot C_S - (V_X - [DigiCodeN_1] \cdot V_{REFP} - [DigiCodeP_1] \cdot V_{REFN}) Bin \cdot C_U \\
& - (V_X - V_{AMPN}) \cdot C_F - (V_X - 0) \cdot (C_{p1n} + C_{p2n})
\end{aligned}
$$
(4.8)

And if proper operations are made, the equation is simlified to:

$$
\begin{aligned}
& (V_{INN} - V_{INP}) \cdot C_S - ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + ((C_{p1p} + C_{p2p}) - (C_{p1n} + C_{p2n})) \cdot V_{REF} \\
& = (V_{AMPN} - V_{AMPP}) \cdot C_F + ((C_{p1p} + C_{p2p}) - (C_{p1n} + C_{p2n})) \cdot V_X \\
& + (([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} + ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN}) \cdot Bin \cdot C_U
\end{aligned}
$$
(4.9)

From the equation (4.9) the amplified signal can be written:

$$(V_{AMPP} - V_{AMPN})$$

$$= (V_{INP} - V_{INN}) \cdot \frac{C_S}{C_F} + (V_X - V_{REF}) \cdot \frac{(C_{p1p} + C_{p2p}) - (C_{p1n} + C_{p2n})}{C_F}$$

$$+ (([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} + ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F}$$

$$+ ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F}$$

$$(4.10)$$

At the begging of the analysis, the input common mode voltage appears at the transfer function. After all, the term with the input common mode voltage will be zero and transfer function will be independent from the input common mode voltage at the end of the analysis.

4.1.4 Resampling phase

The circuit configuration of the resampling state is shown in Figure 4.5.
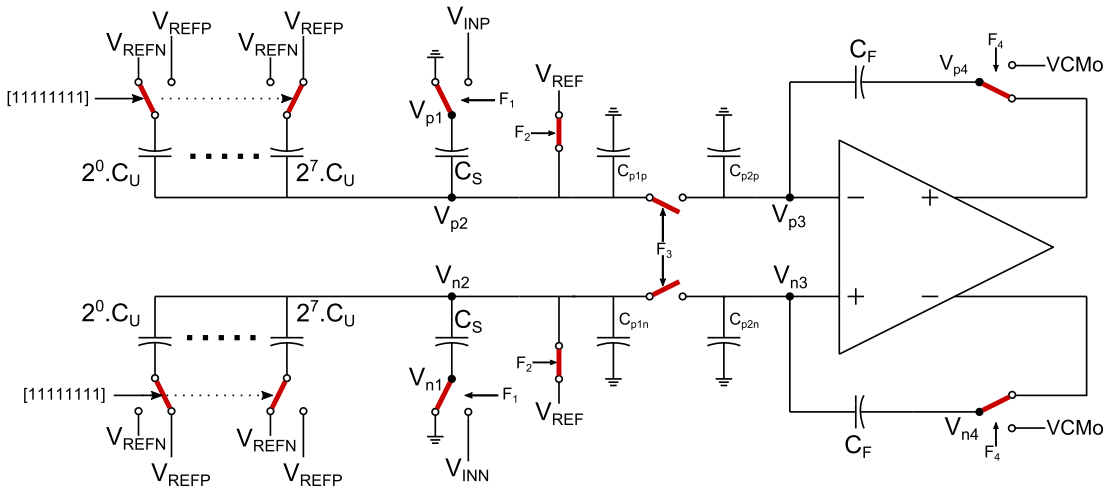
Figure 4.5 : The circuit configuration during resampling phase

During the resampling phase, the inputs of the differential amplifier are separated from the capacitive DAC arrays by the switch F_3 so that the differential amplifier can hold the amplified signal constant at the outputs. Also, the amplified signals are resampled on the capacitive DAC arrays in order to repeat another SAR conversion.

In the resampling phase the control signals are:

$$\begin{aligned}
\phi_1 &= 0(off) \\
\phi_2 &= 1(on) \\
\phi_3 &= 0(off) \\
\phi_4 &= 0(off)
\end{aligned}$$

The node voltages are:

$$\begin{aligned}
V_{1p} &= 0 & V_{3p} &= V_X \\
V_{1n} &= 0 & V_{3n} &= V_X \\
V_{2p} &= V_{REF} & V_{4p} &= V_{AMPP} \\
V_{2n} &= V_{REF} & V_{4n} &= V_{AMPN}
\end{aligned}$$

And instead of the digital code, the outputs of the differential amplifier which are V_{AMPP} and V_{AMPN} are applied to the capacitive DAC arrays all along the resampling state. For the sake of mathematical analysis, the digital codes that are applied to the capacitive DAC arrays are accepted as:

$$\text{DigiCodeP} = [1111\,1111]$$
$$\text{DigiCodeN} = [1111\,1111]$$

In that case, node pairs V_{2p}-V_{3p} and V_{3n}-V_{3n} are disconnected by F_3. Hence, charge equations will be solved for each node independently.

$$Q_{RSp2} = (V_{2p} - V_{1p}) \cdot C_S + (V_{2p} - [1111\ 1111] \cdot V_{4p}) \cdot Bin \cdot C_U + (V_{2p} - 0) \cdot C_{p1p}$$
$$Q_{RSp2} = (V_{REF} - 0) \cdot C_S + (V_{REF} - [1111\ 1111] \cdot V_{AMPP}) \cdot Bin \cdot C_U + (V_{REF} - 0) \cdot C_{p1p}$$

$$(4.11)$$

$$Q_{RSn2} = (V_{2n} - V_{1n}) \cdot C_S + (V_{2n} - [1111\ 1111] \cdot V_{4n}) \cdot Bin \cdot C_U + (V_{2n} - 0) \cdot C_{p1n}$$
$$Q_{RSn2} = (V_{REF} - 0) \cdot C_S + (V_{REF} - [1111\ 1111] \cdot V_{AMPN}) \cdot Bin \cdot C_U + (V_{REF} - 0) \cdot C_{p1n}$$

$$(4.12)$$

$$Q_{RSp3} = (V_{3p} - 0) \cdot C_{p2p} + (V_{3p} - V_{4p}) \cdot C_F$$
$$Q_{RSp3} = (V_X - 0) \cdot C_{p2p} + (V_X - V_{AMPP}) \cdot C_F$$

$$(4.13)$$

$$Q_{RSn3} = (V_{3n} - 0) \cdot C_{p2n} + (V_{3n} - V_{4n}) \cdot C_F$$
$$Q_{RSn3} = (V_X - 0) \cdot C_{p2n} + (V_X - V_{AMPN}) \cdot C_F$$

$$(4.14)$$

4.1.5 2nd conversion phase

The circuit configuration of the 2st conversion state is shown in Figure 4.6.

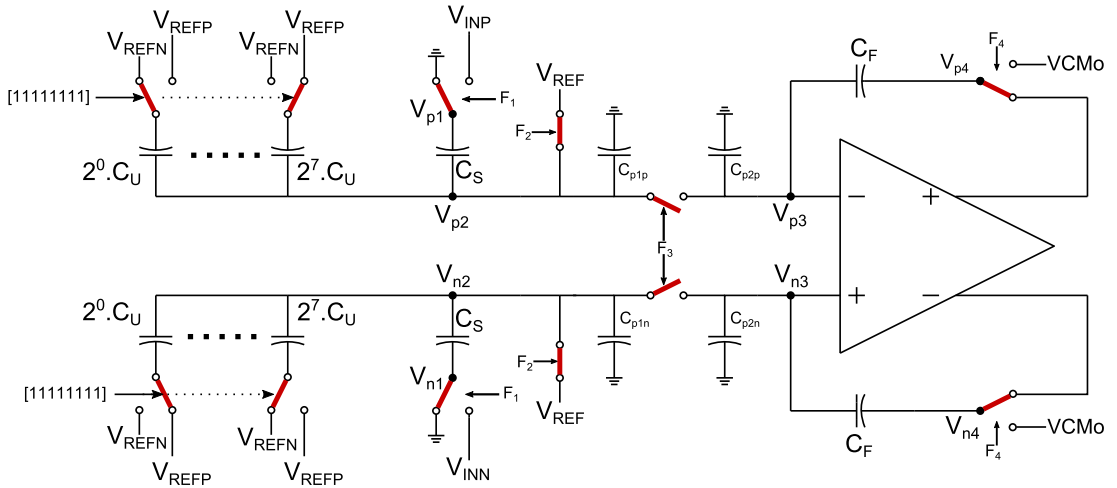**Figure 4.6** : The circuit configuration during 2nd conversion phase

In this state, the differential amplifier is in open loop configuration and acts as a comparator. According to the comparator output, another 8bit digital data is acquired by using SAR algorithm.

If the second conversion is completed without any mistakes, the node voltages are

$$
\begin{aligned}
V_{1p} &= 0 & V_{3p} &= V_{RESP} \\
V_{1n} &= 0 & V_{3n} &= V_{RESN} \\
V_{2p} &= V_{RESP2} & V_{4p} &= V_{CMo} \\
V_{2n} &= V_{RESN2} & V_{4n} &= V_{CMo}
\end{aligned}
$$

And the digital codes that are applied to the capacitive DAC arrays are:

$$
\begin{aligned}
DigiCodeP &= DigiCodeP_2 \\
DigiCodeN &= DigiCodeN_2 \\
DigiCodeP_2 &= \overline{DigiCodeN_2}
\end{aligned}
$$

where $[DigiCodeP_2]$ and $[DigiCodeN_2]$ indicate digital codes that are acquired at the end of the second SAR conversion.

The total charges on the common nodes V_{2p}-V_{3p} nad V_{2n}-V_{3n} can be determined as follow:

$$
\begin{aligned}
Q_{RESp2} =& (V_{2p} - V_{1p}) \cdot C_S + (V_{2p} - [DigiCodeP] \cdot V_{REFP} - [DigiCodeN] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3p} - V_{4p}) \cdot C_F + (V_{2p} - 0) \cdot (C_{p1p} + C_{p2p}) \\
Q_{RESp2} =& (V_{RESP2} - 0) \cdot C_S + (V_{RESP2} - [DigiCodeP_2] \cdot V_{REFP} - [DigiCodeN_2] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{RESP2} - V_{CMo}) \cdot C_F + (V_{RESP2} - 0) \cdot (C_{p1p} + C_{p2p})
\end{aligned}
$$

$$\tag{4.15}$$

$$
\begin{aligned}
Q_{RESn2} =& (V_{2n} - V_{1n}) \cdot C_S + (V_{2p} - [DigiCodeN] \cdot V_{REFP} - [DigiCodeP] \cdot V_{REFN}) \cdot Bin \cdot C_U \\
& + (V_{3n} - V_{4n}) \cdot C_F + (V_{2n} - 0) \cdot (C_{p1n} + C_{p2n}) \\
Q_{RESn2} =& (V_{RESN2} - 0) \cdot C_S + (V_{RESN2} - [DigiCodeN_2] \cdot V_{REFP} - [DigiCodeP_2] \cdot V_{REFN}) Bin \cdot C_U \\
& + (V_{RESN2} - V_{CMo}) \cdot C_F + (V_{RESN2} - 0) \cdot (C_{p1n} + C_{p2n})
\end{aligned}
$$

$$\tag{4.16}$$

From the begging of the resampling phase to the end of the 2nd conversion phase, total charge in the circuit is preseved.

$$
\begin{aligned}
Q_{RSp2} + Q_{RSp3} &= Q_{RESp2} \\
Q_{RSn2} + Q_{RSn3} &= Q_{RESn2}
\end{aligned}
$$

$$\tag{4.17}$$

Subtracting above equations yield

$$
\begin{aligned}
(Q_{RSp2} + Q_{RSp3}) - (Q_{RSn2} + Q_{RSn3}) &= Q_{RESp2} - Q_{RESn2} \\
(Q_{RSp2} - Q_{RSn2}) + (Q_{RSp3} - Q_{RSn3}) &= Q_{RESp2} - Q_{RESn2}
\end{aligned}
$$

$$\tag{4.18}$$

And inserting equations from (4.11) to (4.16)

$$
\begin{aligned}
& \binom{((V_{REF} - 0) \cdot C_S + (V_{REF} - [1111\ 1111] \cdot V_{AMPP}) \cdot Bin \cdot C_U + (V_{REF} - 0) \cdot C_{p1p})}{-((V_{REF} - 0) \cdot C_S + (V_{REF} - [1111\ 1111] \cdot V_{AMPN}) \cdot Bin \cdot C_U + (V_{REF} - 0) \cdot C_{p1n})} \\
& + ((V_X - 0) \cdot C_{p2p} + (V_X - V_{AMPP}) \cdot C_F - ((V_X - 0) \cdot C_{p2n} + (V_X - V_{AMPN}) \cdot C_F)) \\
& = \binom{(V_{RESP2} - 0) \cdot C_S + (V_{RESP2} - [DigiCodeP_2] \cdot V_{REFP} - [DigiCodeN_2] \cdot V_{REFN}) Bin \cdot C_U}{+ (V_{RESP2} - V_{CMo}) \cdot C_F + (V_{RESP2} - 0) \cdot (C_{p1p} + C_{p2p})} \\
& - \binom{(V_{RESN2} - 0) \cdot C_S + (V_{RESN2} - [DigiCodeN_2] \cdot V_{REFP} - [DigiCodeP_2] \cdot V_{REFN}) Bin \cdot C_U}{+ (V_{RESN2} - V_{CMo}) \cdot C_F + (V_{RESN2} - 0) \cdot (C_{p1n} + C_{p2n})}
\end{aligned}
$$

$$\tag{4.19}$$

It follows that

$$
\begin{aligned}
&(V_{AMPN} - V_{AMPP}) \cdot C_F + V_{REF} \cdot (C_{p1p} - C_{p1n}) + V_X \cdot (C_{p2p} - C_{p2n}) \\
&+ (V_{AMPN} - V_{AMPP}) \cdot ([1111\ 1111] \cdot Bin \cdot C_U \\
&= (V_{RESP2} - V_{RESN2}) \cdot C_S + (V_{RESP2} - V_{RESN2}) \cdot C_F \\
&+ V_{RESP2} \cdot (C_{p1p} + C_{p2p}) - V_{RESN2} \cdot (C_{p1n} + C_{p2n}) \\
&+ \left(\begin{array}{c} (V_{RESP2} - V_{RESN2}) - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} \\ + ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot C_U
\end{aligned}
\tag{4.20}
$$

Also, simplification related to *Bin*

$$
[1111\ 1111] \cdot Bin = 2^n - 1
\tag{4.21}
$$

where *n* is 8 which is the length of $Bin = [11111111]$ and it is kept as a variable in order to obtain the transfer function also valid for different values of *n*.

$$
\begin{aligned}
&(V_{AMPN} - V_{AMPP}) \cdot C_F + V_{REF} \cdot (C_{p1p} - C_{p1n}) + V_X \cdot (C_{p2p} - C_{p2n}) \\
&+ (V_{AMPN} - V_{AMPP}) \cdot (2^n - 1) \cdot C_U \\
&= (V_{RESP2} - V_{RESN2}) \cdot C_S + (V_{RESP2} - V_{RESN2}) \cdot C_F \\
&+ V_{RESP2} \cdot (C_{p1p} + C_{p2p}) - V_{RESN2} \cdot (C_{p1n} + C_{p2n}) \\
&+ \left(\begin{array}{c} (V_{RESP2} - V_{RESN2}) - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} \\ + ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot C_U
\end{aligned}
\tag{4.22}
$$

Since $(V_{AMPN} - V_{AMPP})$ equals to equation (4.10)

$$
\begin{aligned}
&- \left(\begin{array}{c} (V_{INP} - V_{INN}) \cdot \frac{C_S}{C_F} + (V_X - V_{REF}) \cdot \frac{(C_{p1p} + C_{p2p}) - (C_{p1n} + C_{p2n})}{C_F} \\ + ([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F} \end{array} \right) \cdot ((2^n - 1) \cdot \cdot C_U + C_F) \\
&+ V_{REF} \cdot (C_{p1p} - C_{p1n}) + V_X \cdot (C_{p2p} - C_{p2n}) \\
&= (V_{RESP2} - V_{RESN2}) \cdot C_S + (V_{RESP2} - V_{RESN2}) \cdot C_F \\
&+ V_{RESP2} \cdot (C_{p1p} + C_{p2p}) - V_{RESN2} \cdot (C_{p1n} + C_{p2n}) \\
&+ \left(\begin{array}{c} (V_{RESP2} - V_{RESN2}) - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} \\ + ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot C_U
\end{aligned}
\tag{4.23}
$$

Now, extracting $(V_{INP} - V_{INN})$ from the equation leads to

$$
\begin{aligned}
&- (V_{INP} - V_{INN}) \cdot \frac{C_S}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&= (V_{RESP2} - V_{RESN2}) \cdot C_S + (V_{RESP2} - V_{RESN2}) \cdot C_F \\
&+ V_{RESP2} \cdot (C_{p1p} + C_{p2p}) - V_{RESN2} \cdot (C_{p1n} + C_{p2n}) \\
&+ \left(\begin{array}{c} ([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F} \end{array} \right) \cdot ((2^n - 1) \cdot C_U + C_F) \\
&+ \left(\begin{array}{c} (V_{RESP2} - V_{RESN2}) - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} \\ + ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot C_U \\
&+ ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&+ (V_X - V_{REF}) \cdot \frac{(C_{p1p} + C_{p2p}) - (C_{p1n} + C_{p2n})}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&+ V_{REF} \cdot (C_{p1p} - C_{p1n}) + V_X \cdot (C_{p2p} - C_{p2n})
\end{aligned}
\tag{4.24}
$$

45

The parasitic capacitances in the circuit are small and because of the symmetrical circuit structure. Also, the difference between these parasitic capacitances are negligible compared to the other capacitance in the circuit. As a result

$$C_{p1} = C_{p1p} \cong C_{p1p} \quad and \quad C_{p2} = C_{p2p} \cong C_{p2p} \tag{4.25}$$

The above assumption reduce the number of the parasitic capacitance in the transfer function and leads to simplification on the equation (4.24)

$$
\begin{aligned}
&- (V_{INP} - V_{INN}) \cdot \frac{C_S}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&= (V_{RESP2} - V_{RESN2}) \cdot C_S + (V_{RESP2} - V_{RESN2}) \cdot C_F \\
&+ (V_{RESP2} - V_{RESN2}) \cdot (C_{p1} + C_{p2}) \\
&+ \left(\begin{array}{c} ([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN} \cdot Bin \cdot \frac{C_U}{C_F} \\ + ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F} \end{array} \right) \cdot ((2^n - 1) \cdot C_U + C_F) \\
&+ \left(\begin{array}{c} (V_{RESP2} - V_{RESN2}) - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} \\ - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot C_U
\end{aligned}
\tag{4.26}
$$

Also, at the end of the conversion phase the node voltages V_{2p} and V_{2n} approach each other in consequence of the conversion algorithm.

$$V_{RESP2} \cong V_{RESN2} \cong V_{RES2} \tag{4.27}$$

Therefore, the equation (4.26) is simplified to

$$
\begin{aligned}
&- (V_{INP} - V_{INN}) \cdot \frac{C_S}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&= ([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFP} \cdot Bin \cdot \frac{C_U}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&+ ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFN} \cdot Bin \cdot \frac{C_U}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F) \\
&- ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP} - ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \cdot Bin \cdot C_U \\
&+ ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_F} \cdot ((2^n - 1) \cdot C_U + C_F)
\end{aligned}
\tag{4.28}
$$

$$
\begin{aligned}
&(V_{INP} - V_{INN}) \\
&= ([DigiCodeP_1] - [DigiCodeN_1]) \cdot V_{REFP}) \cdot Bin \cdot \frac{C_U}{C_S} \\
&+ ([DigiCodeN_1] - [DigiCodeP_1]) \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_S} \\
&+ \left(\begin{array}{c} ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFP}) \\ ([DigiCodeP_2] - [DigiCodeN_2]) \cdot V_{REFN}) \end{array} \right) \cdot Bin \cdot \frac{C_U \cdot C_F}{((2^n - 1) \cdot C_U + C_F) \cdot C_S} \\
&- ([0000\ 0001] \cdot V_{REFP} - [0000\ 0001] \cdot V_{REFN}) \cdot Bin \cdot \frac{C_U}{C_S}
\end{aligned}
\tag{4.29}
$$

In order to simplify the transfer function, V_{REFN} is accepted as 0. Actually, in the real circuit, V_{REFN} is connected to ground. Therefore

$$
\begin{aligned}
&V_{INP} - V_{INN} \\
&= \left(\begin{array}{c} ([DigiCodeP_2] - [DigiCodeN_2]) \cdot Bin \cdot \frac{C_F}{((2^n - 1) \cdot C_U + C_F)} \\ ([DigiCodeP_1] - [DigiCodeN_1]) \cdot Bin - ([0000\ 0001] \cdot Bin) \end{array} \right) \cdot V_{REFP} \cdot \frac{C_U}{C_S}
\end{aligned}
\tag{4.30}
$$

Another definition can be made to reduce variable number in the transfer function.

$$[DigiCode2] = ([DigiCode2p] - [DigiCode2n])$$
$$[DigiCode2] = ([DigiCode1p] - [DigiCode1n])$$

(4.31)

And equation (4.30) becomes

$$V_{INP} - V_{INN}$$
$$= \left(\frac{[DigiCode2] \cdot Bin \cdot \frac{C_F}{((2^n-1) \cdot C_U + C_F)}}{[DigiCode1] \cdot Bin - ([0000\ 0001] \cdot Bin)} \right) \cdot V_{REFP} \cdot \frac{C_U}{C_S}$$

(4.32)

The coefficient $\dfrac{C_F}{((2^n - 1) \cdot C_U + C_F)}$ designate the weight of the digital data that is obtained from the second conversion. So that, the actual digital data is the combination of the first and second conversion and the coefficient, K.

$$[DigiCode] = K \times [DigiCode2] + [DigiCode1]$$

(4.33)

In the design the capacitance C_F is equal to $4 \times C_U$ and as a result the coefficient becomes $\dfrac{1}{64.75}$. With the selected C_F value, 14 bits of digital data is obtained from the raw 16 bits data. In addition, the mismatch of the capacitor C_F will cause a nonlinearity since it determines the coefficient K. After the calculation of coefficient K, the input range can be determined by using equation (4.30) where the maximum digital output is produced.

$$[DigiCode1] = [11111111]$$
$$[DigiCode2] = [00111111]$$

(4.34)

If the conversion occurs without any mistakes, the most significant first and second bit of the second 8bit conversion will always be 0.

$$(V_{INP} - V_{INN}) = \left(\begin{array}{c} [11111111] \cdot B_{IN} \\ +[00111111] \cdot B_{IN} \cdot \frac{1}{64.75} - [00000001] \cdot B_{IN} \end{array} \right) \cdot V_{REFP} \cdot \frac{C_U}{C_S}$$

$$(V_{INP} - V_{INN}) = \left((2^8 - 1) + (2^6 - 1) \frac{1}{64.75} - 1 \right) \cdot V_{REFP} \cdot \frac{C_U}{C_S}$$

$$(V_{INP} - V_{INN}) \cong ((2^8 - 1) + 1 - 1) \cdot V_{REFP} \cdot \frac{C_U}{C_S}$$

$$(V_{INP} - V_{INN}) \cong 255 \cdot V_{REFP} \cdot \frac{C_U}{C_S}$$

(4.35)

For a 16V maximum peak voltage and 2V positive reference voltage, value of C_S is

$$C_S = 255 \cdot 2 \cdot \frac{C_U}{16} = 31.875 \cdot C_U$$

(4.36)

Instead of poly-poly capacitance, metal-insulator-metal capacitance is used to realize the sampling capacitance C_S because of high voltage input signal. However, the matching of MIM capacitor is not predict to be reliable as poly-poly capacitance and the exact input range will be calculated experimentally. As a result $32V_{pp}$ input signal range can be achieved with the ADC.

47

5. MODELLING OF THE SAR ADC

Due to the increase of the transistor number and the complexity of the circuits, the usage of behavioral model is inevitable to analyze the performance of an electronic system. The behavioral models that are generated by using MATLAB and SIMULINK offer a convenient method to start design of complex circuit such as analog to digital converters.

There are 4 main ADC architecture: Sigma-Delta ($\Sigma\Delta$) ADC, Successive Approximation Register (SAR) ADC, Pipeline ADC, Flash ADC and each of them use different type of conversion method that has various analog and digital block. Amongst to other types, 2step SAR-ADC is widely used architecture that offers solution for 12 to 16 bit resolution at medium conversion rates up to 5 MSps (Haenzsche et al., 2010). The performance limit of the ADC is dependent to analog components such as the capacitor array, comparator etc. (Lin et al., 2005). Therefore it is important to examine design trade-off between multiple factors including capacitor array mismatch, interstage gain variation, compartor offset, clock signal and effect of parasitic capacitance.

In this section, the measured ADC structure will be modeled by using transfer functions that are derived in chapter 4. Before delving into complex charge model, a simple voltage model will be presented to understand behaviour of the two-step ADC architecture. Note that, simple voltage model does not include any speed or voltage limitations. After that, the charge model which is main subject of the thesis will be introduced including all non-idealities of the system. Each blocks in the system are examined individually and non-idealities of the block are observed one by one during the analysis. Amongs to other blocks, capacitive DAC array is the most dominant block where the matching of the elements is vital to reach aimed resolution. Thus, analysis of the CDAC occupy important part in this chapter. Also, comparator is another important block in the system that has to examined and modeled carefully. After modeling of the block, simulation results will be given for static and dynamic parameters of the ADC. The model is established and verified in Matlab Simulink environment. The simulation results give important observation about SAR-ADC which could be used further design approaches. Finally, the correlation between model and the measurement results will investigated together to verify model at the end of the chapter.

5.1 Voltage Model

Figure 5.1 presents the basic voltage model that is established to understand working principle of the two-step SAR-ADC.
The input signal V_{in} is an ideal sine wave that satisfies:

$$V_{in} = Amp * sin(Freq * t + Phase) + Bias \qquad (5.1)$$

where *Bias* is the input common-mode-voltage, *Amp* is the amplitude, *Freq* is the frequency of the signal. The conversion starts with sampling of the analog input signal
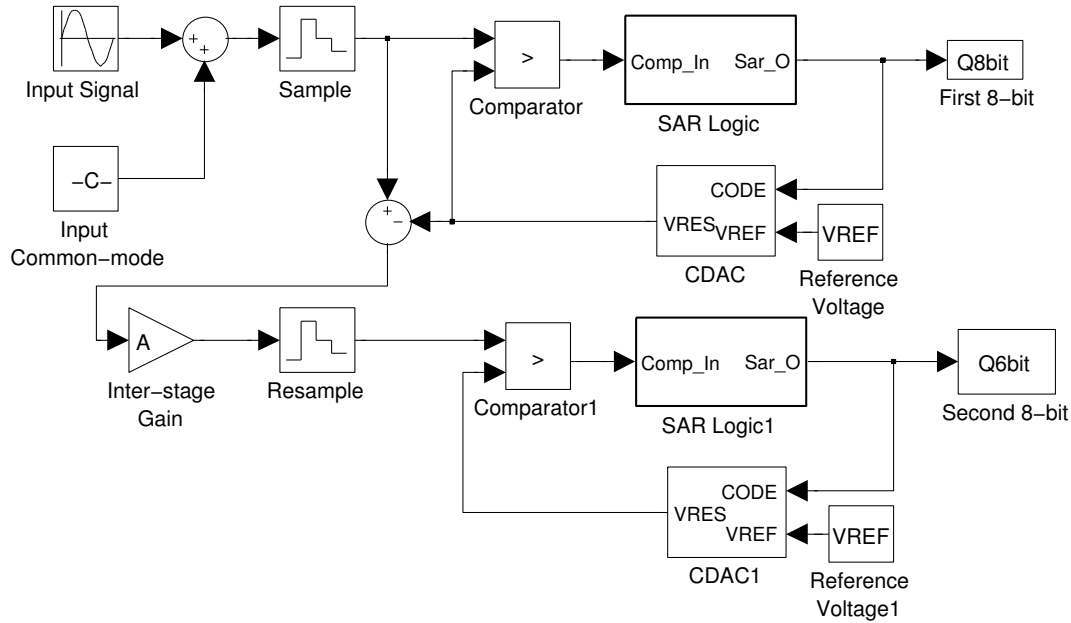
Figure 5.1 : The basic voltage model of two-step SAR-ADC

at sampling frequency. The zero-order hold block from Simulink library is used to achieve sampling function. After that, SAR conversion begins with comparison of the sampled signal and the DAC voltage which is obtain with respect to mid-code and reference voltage. At the end of the first SAR conversion, residue voltage which is subtraction of the sampled signal and final DAC voltage. Then residue voltage is amplified by inter-stage gain block to perform another SAR conversion. Next, the amplified signal is re-sampled to repeat another SAR conversion. At the end of the second conversion, first and second 8bit are combined with respect to inter-stage gain of 64 which is accomplished by gain block. As a result 14bit digital data is obtained.

5.2 Charge Model

A common SAR-ADC consists of a capacitor DAC array, a comparator, a digital control logic and number of switches (Lin et al., 2005). The conversion is performed by following order: sampling, first conversion, amplification, resampling and second conversion. First, analog input signal is sampled and hold on sampling capacitor as a charge. Then first conversion start with redistribution of the sampled charged between sampling capacitor and DAC array in order to determine 8 bit digital output. The resolution of the ADC is primarily depend on the matching of the capacitor array since charge redistribution is accomplished with respect to capacitor ratio. Variations of capacitor values results errors in voltage comparisons that is performed by the comparator.

Figure 5.2 shows the complete behavioral model of the SAR ADC that is established at the end of the study. Each block in the figure represents sampling, first conversion, amplification, resampling and second conversion phase of the converter. A state machine at the top left corner of the figure is build to control phase of the ADC. It uses clock signal as an input and generate 5 signal for 5 phase of the ADC. Each block accepts an enable signal and the clock signal. When the state machine sends the enable signal to the related block, the block is activated. Then, it makes calculation in

every rising edge of the clock signal. Therefore every block that is used for each state is triggered enabled subsystems. Apart from them, there are two static blocks in the figure that are named as Inputs and References. Inputs block generates analog input signal with respect to user definition as a sine wave or a ramp signal. References block generates necessary reference voltages as positive and negative reference voltage for DAC, output common mode voltage for amplifier and reference voltage for internal node of the ADC.

Before delving into the behavioral model simulation results, fundamental blocks of the SAR architecture are investigated and modeled including non-ideal effects. Each block has different type of non-idealities that are examined carefully. At the end of the section measurement results and the model results will be investigated.

5.2.1 Blocks

5.2.1.1 State machine

The state machine is modeled by using the flow diagram of the 2-step SAR ADC that was drawn in Figure 5.3. It requires a clock signal as an input and counts number of rising edges to generate output signals: SAMPLE, SAR1, AMPLIFY, RESAMPLE and SAR2. The modeled state machine is shown in Figure 5.3.

Clock frequency, clock amplitude, rise and fall times are editable parameters for the system clock. The state machine generates the output signals with respect to these parameters. Also, each output has a number of clock cycle parameter and the state machine decides length of the each state by using clock cycle parameter which are named as NSAMPLE, NSAR1, NAMPLIFY, NRESAMPLE, NSAR2.

At the beginning, the counter start to count from 0 to NSAMPLE value. When the NSAMPLE value is reached, reset signal rise and the counter starts to count from 0 to NSAR1 value. Then, the counter counts from 0 to NAMPLIFY, NRESAMPLE and NSAR2 respectively. At the end of the SAR2 all cycle is repeated again.

Figure 5.4 shows the modeled state machine and a clock signal. The clock signal has 1MHz frequency, 1V amplitude and the clock cycle parameters are NSAMPLE= 3, NSAR1= 8, NAMPLIFY= 3, NRESAMPLE= 3 and NSAR2= 8. According to these parameters, the output signals are plotted in figure 5.5.

However these signals are generated from a ideal clock signal. In order to add rise and fall time parameter to the signals, rate limiter block is placed after the CLK block. Another non-idealities about clock signal is sampling jitter that results variations on the sampling time.

$$y(t + \Delta t) - y(t) = \Delta t \times \frac{d}{dt} y(y) \tag{5.2}$$

where Δt is uncertainty in the clock signal $y(t)$. Figure 5.6 shows the block diagram of the clock jitter model that is generated by using equation (5.2).

5.2.1.2 Comparator

The basic comparator circuit is an opamp used in the open-loop configuration as shown in Figure 5.7. The output voltage is given by
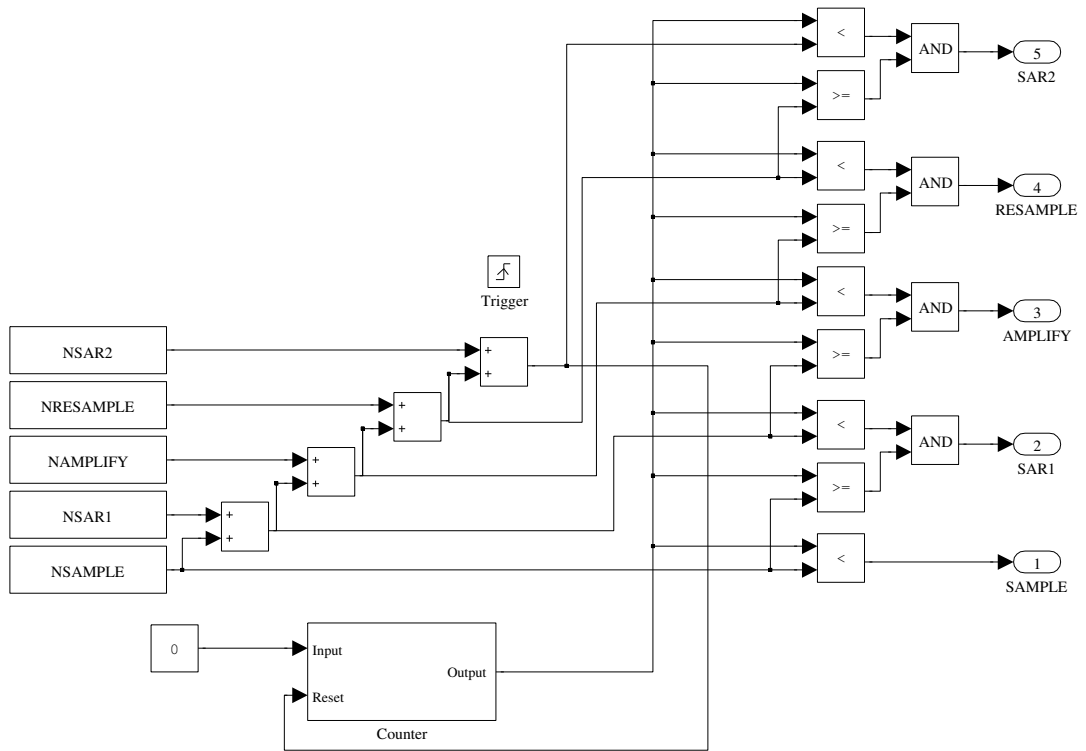
$$V_o = A(V_{inp} - V_{inn}) \tag{5.3}$$

51

Figure 5.2 : Complete behavioral model of the SAR ADC

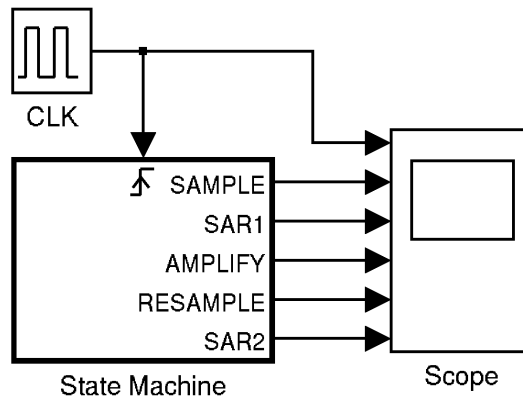**Figure 5.3** : Modeled state machine block diagram

Figure 5.4 : Modeled state machine with clock signal

where *A* is open-loop gain, V_{inp} and V_{inn} are voltages at the non-inverting and inverting terminals respectively. According to (5.3), voltage transfer characteristic of the comparator is draw in Figure 5.8.

For $V_{inp} - V_{inn} > V_t$, V_{out} reaches positive supply voltage and for $V_{inp} - V_{inn} < -V_t$, V_{out} reaches negative supply voltage. For $-V_t < V_{inp} - V_{inn} < +V_t$, the comparator in linear region and the output voltage does not reach rail voltages.

The comparator is fundamental block of the SAR ADC as it creates a link between the analog and digital domain (Salah Hanfoug & Barra, 2014). In every conversion step, the comparator decides whether the input signal is larger than the reference voltage or not. So, an error in the comparison causes an incorrect comparator decision that shows itself as INL and DNL error in the ADC characteristic.

Figure 5.9 presents the comparator model that is used in behavioral SAR model. First, the two input signals are subtracted at the input as it stated in equation (5.3). After
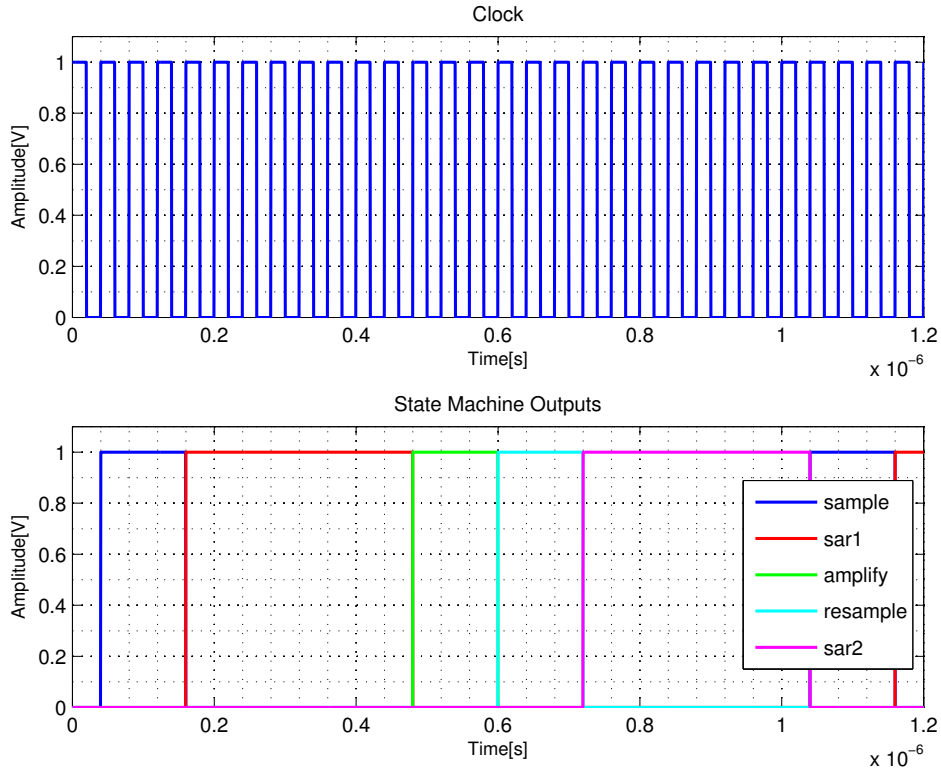
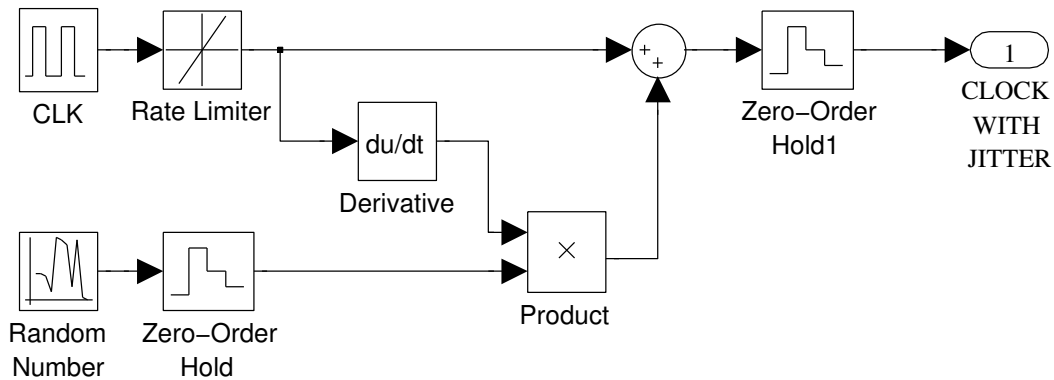**Figure 5.5** : Output signals of state machine

Figure 5.6 : Clock signal with jitter

that, a constant value is added to subtraction to model comparator offset. Then, a gain block is used as open-loop gain with respect to (5.3). Then, the amplified signal is saturated to positive or negative supply voltage. Finally, it is converted to logic levels by compare to zero block that decides the output is logic 1 or logic 0.

5.2.1.3 Capacitive divider DAC

First, examine two capacitors connected in series in 5.10. The output voltage, V_{out}, becomes;

$$V_{out} = V_{ref} \frac{C_1}{C_1 + C_2} \tag{5.4}$$

If C_1 and C_2 is made by k and $2^n - k$ unity element respectively, according to equation
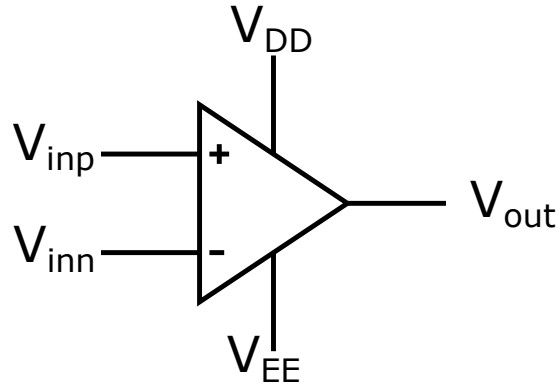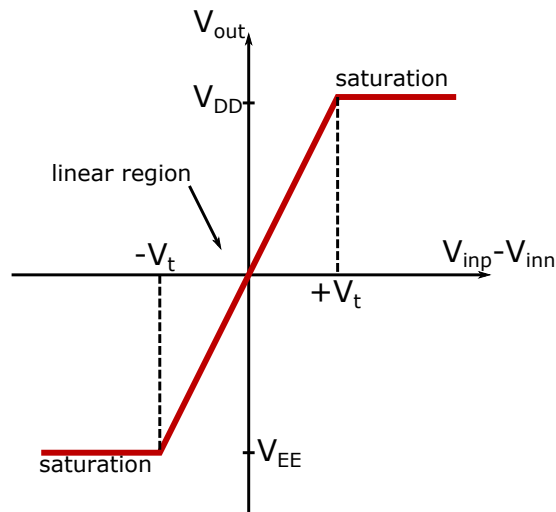
54

Figure 5.7 : Basic comparator symbol

Figure 5.8 : Voltage transfer characteristic of comparator
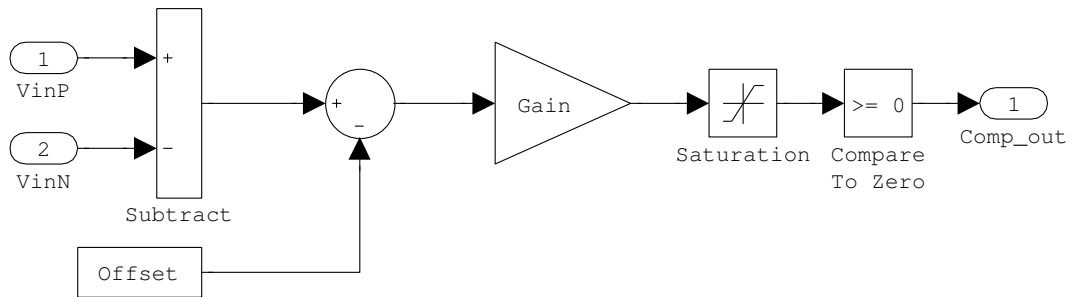
Figure 5.9 : Comparator model block diagram

(5.4), output voltage become $V_{out} = V_{ref}\dfrac{k}{2^n}$ as needed by a DAC (Maloberti, 2007). Figure 5.10 illustrates unity element (C_u) selection for a simple capacitor divider that use 2^n unity capacitor in total. Generally, instead of using two capacitors, binary weighted capacitor array architecture is used that is illustrated in Figure 5.11.

The bottom terminals of the capacitors are connected to ground or reference voltage, V_{REF}, through a switch. Depending on the switches positions, the two capacitor

Figure 5.10 : Simple capacitor divider

Figure 5.11 : Binary weighted capacitor array

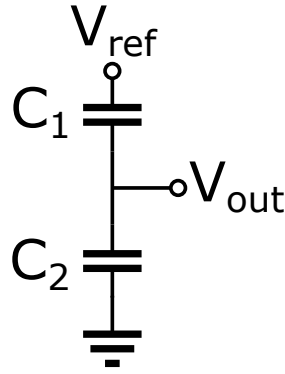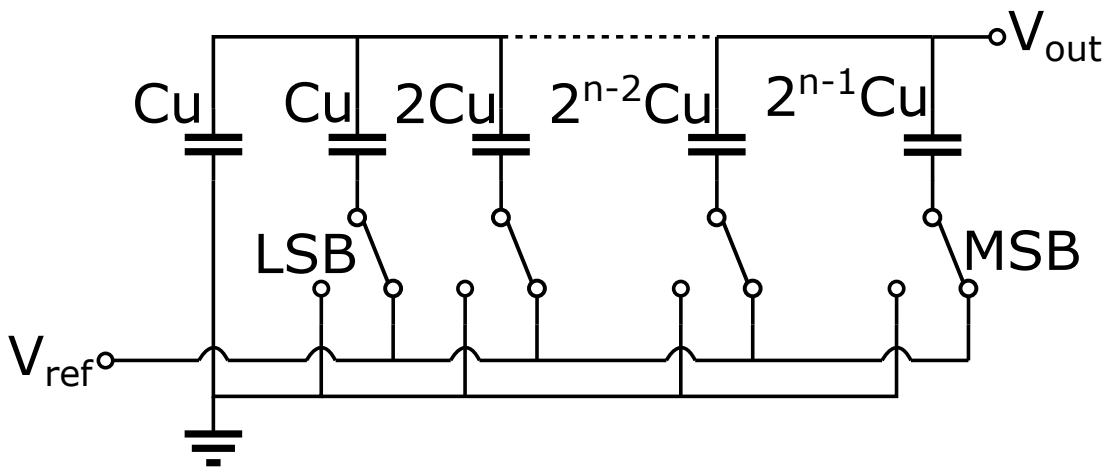structure at the beginning is formed for 2^n different output voltage level. Also, the full scale voltage is $(2^n - 1)V_{ref}/2^n$ because one unity capacitance is always placed between ground and the output in order to obtain proper binary transition.

The parasitic capacitances connected to the bottom terminals do not alter the transfer function of capacitor divider as it is located on low impedance nodes. However the parasitic capacitances connected to the output node disturb the capacitive division ratio and the output voltage for a binary weighted capacitor array becomes

$$V_{out} = V_{ref} \frac{\sum_{1}^{n} b_i C_i}{\sum_{0}^{n} C_i + \sum_{0}^{n} C_{p,i}} \tag{5.5}$$

where b_i are the digital control bits, C_i is binary weighted capacitor matrix and $C_{p,i}$ is parasitic capacitors on the output node. If the value of parasitic capacitances is constant, the effect of it results as a gain error on the equation (5.5). Nevertheless, it is dependent on the output voltage and shows itself as a non-linear dependence that cause harmonic distortion.

The relation between the total number of element in the capacitor divider and number

56

of the bits is exponential and it is not efficient to use the binary weighted architecture directly because higher number of the bits increase the silicon area and capacitive load. Using the minimum size unity capacitance can solve the problem but it is not preferable due to matching requirement.
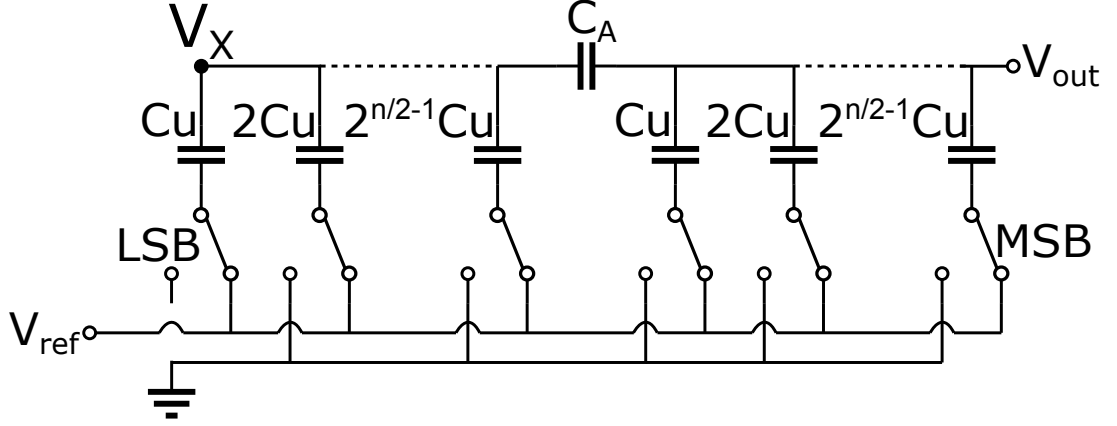

Figure 5.12 : The capacitor array with an attenuator capacitor in the middle

The binary weighted capacitor array with an attenuator capacitance in figure 5.12 is an alternative approach that has smaller capacitive load and lower power consumption as result of reduced the total number of element (Wong et al., 2009). However, the linearity of the DAC array is more dependent on parasitic capacitance in the circuit and must be examined carefully.

The attenuation capacitance and the entire left array creates an attenuation factor K.

$$K = \frac{C_A \cdot (2^{n/2} - 1)C_u}{C_A + (2^{n/2} - 1)C_u} \tag{5.6}$$

which is the series combination of them. If the value of C_A is equal to C_u, the attenuation factor K yields;

$$K = \frac{C_u(2^{n/2} - 1)C_u}{C_u + (2^{n/2} - 1)C_u} = \frac{(2^{n/2} - 1)}{2^{n/2}}C_u \tag{5.7}$$

and In order to find effect of split capacitor on the output voltage, two different condition is examined;

- If the corresponded bit, m, is at the LSB side of the array goes to logic $1(m < n/2)$, charge equation at X node

$$
\begin{aligned}
&V_X\left(2^{n/2} - 1 - 2^{m-1}\right) + \left(V_X - V_{ref}\right)C_u + (V_X - V_{out})C_u = 0 \\
&V_X\left(2^{n/2} - 1\right)C_u - V_{ref}2^{m-1}C_u + (V_X - V_{out})C_u = 0 \\
&V_X2^{n/2}C_u - V_{ref}2^{m-1}C_u - V_{out}C_u = 0 \\
&V_X2^{n/2} - V_{ref}2^{m-1} - V_{out} = 0
\end{aligned}
\tag{5.8}
$$

and charge equation at the output node

$$V_{out}\left(2^{n/2}-1\right)C_u+(V_{out}-V_X)C_u=0$$

$$V_{out}2^{n/2}C_u-V_XC_u=0 \tag{5.9}$$

$$V_X=V_{out}2^{n/2}$$

If the equation (5.9) is written in (5.8)

$$V_{out}2^{n/2}2^{n/2}-V_{ref}2^{(m-1)}-V_{out}=0$$

$$V_{out}2^n-V_{ref}2^{(m-1)}-V_{out}=0$$

$$V_{out}\left(2^m-1\right)-V_{ref}2^{(m-1)} \tag{5.10}$$

$$V_{out}=V_{ref}\frac{2^{(m-1)}}{2^n-1}$$

- If the corresponded bit, m, is at the MSB side of the array goes to logic $1(m>n/2)$, charge equation at X node

$$V_X\left(2^{n/2}-1\right)C_u+(V_X-V_{out})C_u=0$$

$$V_X=V_{out}2^{-m/2} \tag{5.11}$$

and charge equation at the output node

$$V_{out}\left(2^{n/2}-1-2^{(m-n/2-1)}\right)C_u+\left(V_{out}-V_{ref}\right)2^{(m-n/2-1)}C_u+(V_{out}-V_X)C_u=0$$

$$V_{out}2^{n/2}C_u-V_{ref}2^{(m-n/2-1)}C_u-V_XC_u=0$$

$$V_{out}2^{n/2}-V_{ref}2^{(m-n/2-1)}-V_X=0$$

$$\tag{5.12}$$

If the equation (5.11) is written in (5.12)

$$V_{out}2^{n/2}-V_{ref}2^{(m-n/2-1)}-V_{out}2^{-m/2}$$

$$V_{out}=\frac{2^{(m-n/2-1)}}{2^{n/2}-2^{-n/2}}V_{ref} \tag{5.13}$$

$$V_{out}=\frac{2^{m-1}}{2^n-1}$$

Equations (5.10) and (5.13) prove that if C_A is equal to C_u, LSB and MSB side of the array has the same effect on the output voltage. Therefore, equation (5.5) is still valid but it needs to re-arrange according to the previous analysis.

$$V_{out}=V_{ref}\frac{\sum_1^{n/2}\frac{b_iC_i}{K}+\sum_{n/2+1}^n b_iC_i}{\sum_1^{n/2}\frac{C_i}{K}+\sum_{n/2+1}^n C_i+\sum_1^{n/2}\frac{C_{p,i}}{K}+\sum_{n/2+1}^n C_{p,i}} \tag{5.14}$$

where K is attenuation factor in equation (5.6). However, equation (5.6) should be re-examined by including the effect of parasitic capacitance.

$$K = \frac{C_A \cdot ((2^{n/2} - 1)C_u + \sum_{1}^{n/2} C_{p,i})}{C_A + (2^{n/2} - 1)C_u + \sum_{1}^{n/2} C_{p,i}} \tag{5.15}$$

By using equation (5.14), a capactive DAC model is presented in figure 5.13. Assume
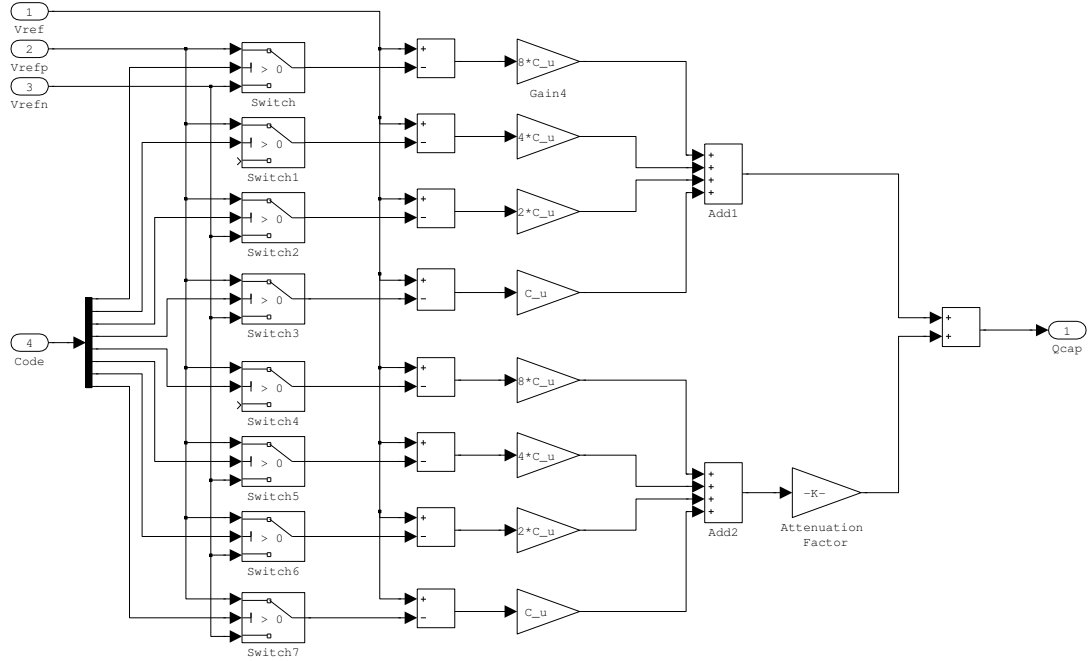

Figure 5.13 : The capacitor array block diagram

that digital code is mid-code 10000000. All switches, except the most significant capacitor's switch, connected to V_{REFN} voltage to the bottom of capacitor array and the the most significant capacitor's switch is connected to V_{REFP}. Since top terminal of the capacitors are connected to V_{REF}, capacitors can be modeled as gain block. In addition, there is an extra gain block which is located at the left side of the array name as attenuation factor. This block is used to model split capacitor effect as it stated in equation (5.6).

Figure 5.14 shows ideal transfer curve of 8 bit split DAC where attenuation factor is $1/16$. It has the same transfer function of ideal binary weighted capacitor array. There are 255 quantization step at the transfer function and each step length is 1LSB. Thus, DNL and INL error is zero. Also, output spectrum of the ideal DAC is shown in figure 5.15. Input signal is a sine wave at 17.69kHz and calculated SNR is 86.04dB. It is clear that there is no any spur at the output spectrum. Only noise source in the system is unavoidable quantization noise.

In order to examine effect of parasitic capacitance, the least and most significant capacitor value is scattered respectively while the attenuation factor is still $1/16$. Figure 5.16 shows the transfer function, INL and DNL plots when the weight of the
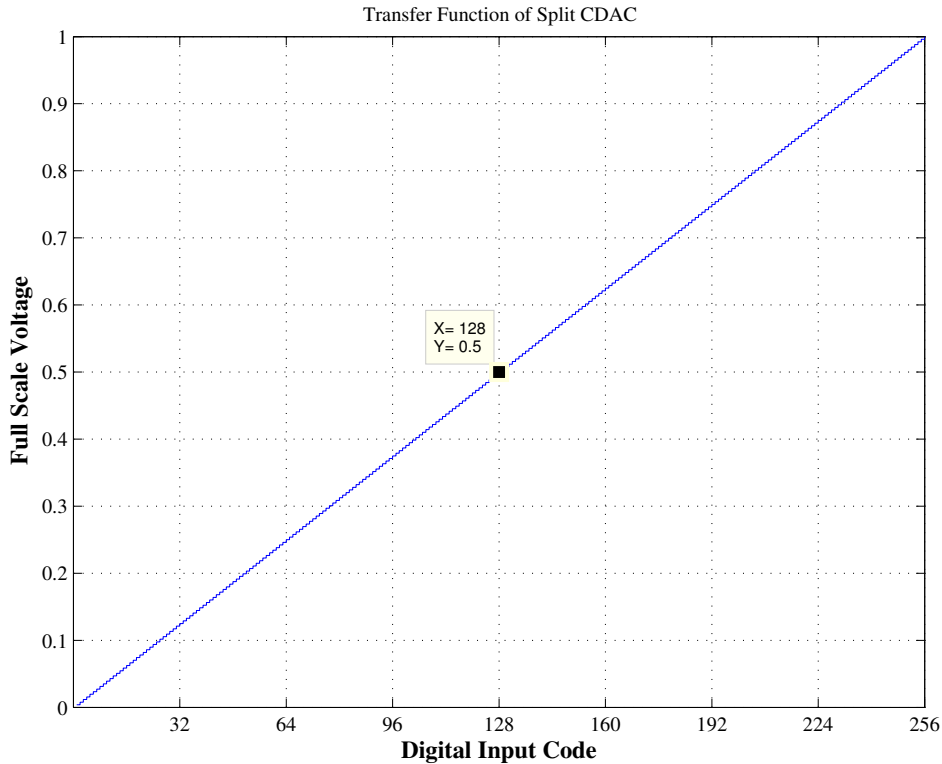
Figure 5.14 : Ideal transfer curve of DAC model

least significant capacitor is scattered form 1 to 1.5.

As it seen from figure 5.16, the mid-code does not give half of full scale voltage(0.5V) anymore. Since the least significant capacitor is scattered, DNL error become 0.497 and -0.5LSB respectively because in every step of DAC, the capacitor is charged to V_{ref} or 0 to provide 1LSB quantization step. INL error is laying between 0.497 and -0.497 LSB and it is monotone decreasing behaviour by the reason of DNL error.

Figure 5.17 shows the transfer function, INL and DNL plots when the weight of the most significant capacitor is scattered form 8 to 7.5.

When the most significant bit becomes 1, the most significant capacitor is charged to V_{ref} and the transfer function has a bump at the mid-code which is '10000000'. This time, DNL error stay constant at 0.032LSB except mid-code where DNL error reaches -8.277LSB. Similarly, INL error raise with the same slope until mid-code. When it reaches the maximum value, 4.113LSB, the slope direction is changed and the minimum value, -3.984LSB is obtained.

Figure 5.18 demonstrates the transfer function, DNL and INL plots when the attenuation factor is scattered from $1/16$ to $1/15$.

As the figure 5.18 16 distinguished segments at every 16th code, resulting from attenuation capacitor variation. All capacitors at the left side of the CDAC are effected by same ratio($1/15$) and thus they have the same DNL error. As it seen from DNL graph, first 15 code has the same DNL error which is 0.0625LSB. Then 16th code reaches -1LSB peak value because the least significant capacitor at the right side of the array is activated. Since INL is integration of DNL, INL error has the same slope for first 15 code, then 16th code has a similar bump in DNL eror. Maximum INL error
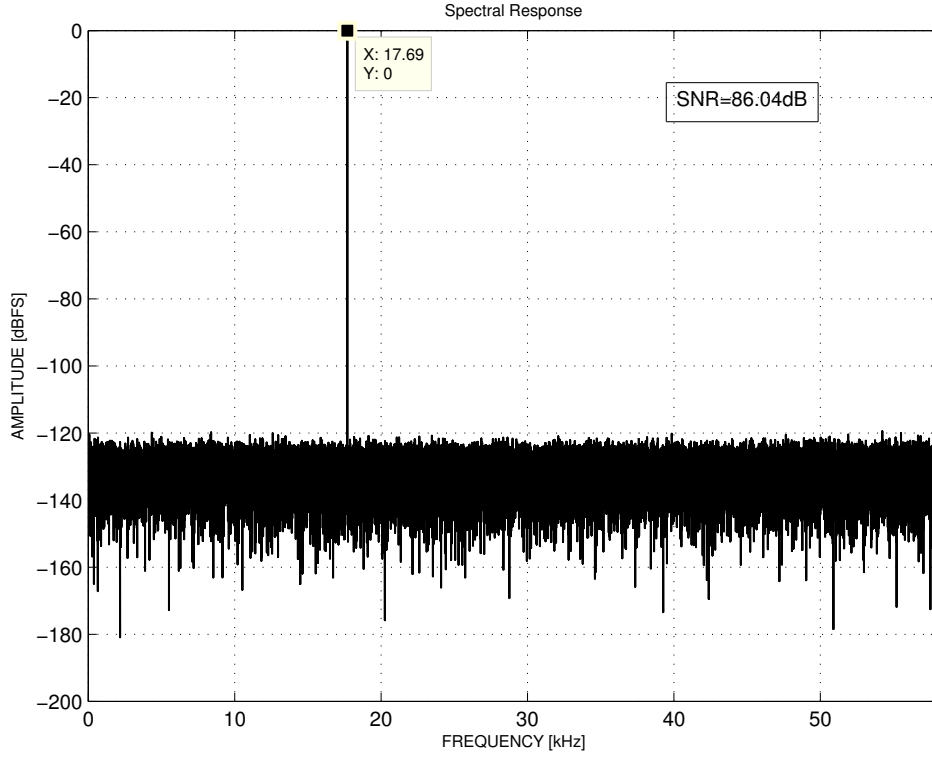
Figure 5.15 : Spectrum response of ideal DAC model

is determined as 0.9375LSB.

It is known fact that the maximum obtainable resolution of SAR ADC is primarily limited by the matching of the capacitors in the circuit (Lin et al., 2005). Independent from the layout placement or routing, capacitance mismatch is proportional to the square root of the occupied area and is given by

$$\sigma(\frac{\Delta C}{C}) = \frac{A_C}{\sqrt{WL}}$$ **(5.16)**

where A_C is a parameter depends on technology and capacitor type, the parameter W and L are the width and the length of the capacitor, the term $(\frac{\Delta C}{C})$ is standard deviation of the nominal value of the capacitor (Haenzsche et al., 2010). The parameter A_C is 0.45% for AMS 0.35μ technology. The relation between achievable resolution and deviation of the capacitor is given in (Lin et al., 2005)

$$N_{max} = log_2(\frac{1 + \sigma + \sqrt{1 + 2\sigma - 3\sigma^2}}{2\sigma})$$ **(5.17)**

where N_{max} is maximum obtainable resolution and σ is standard deviation of the unit capacitor in the circuit. However as it is stated in (Haenzsche et al., 2010), a worst case analysis is used to determine relation between unit capacitance and the attainable resolution N. Figure 5.19 shows the distribution based approach that leads more realistic approximation than (Lin et al., 2005).

To obtain 14bit resolution, σ should be smaller than 3×10^{-4}. The unity capacitor
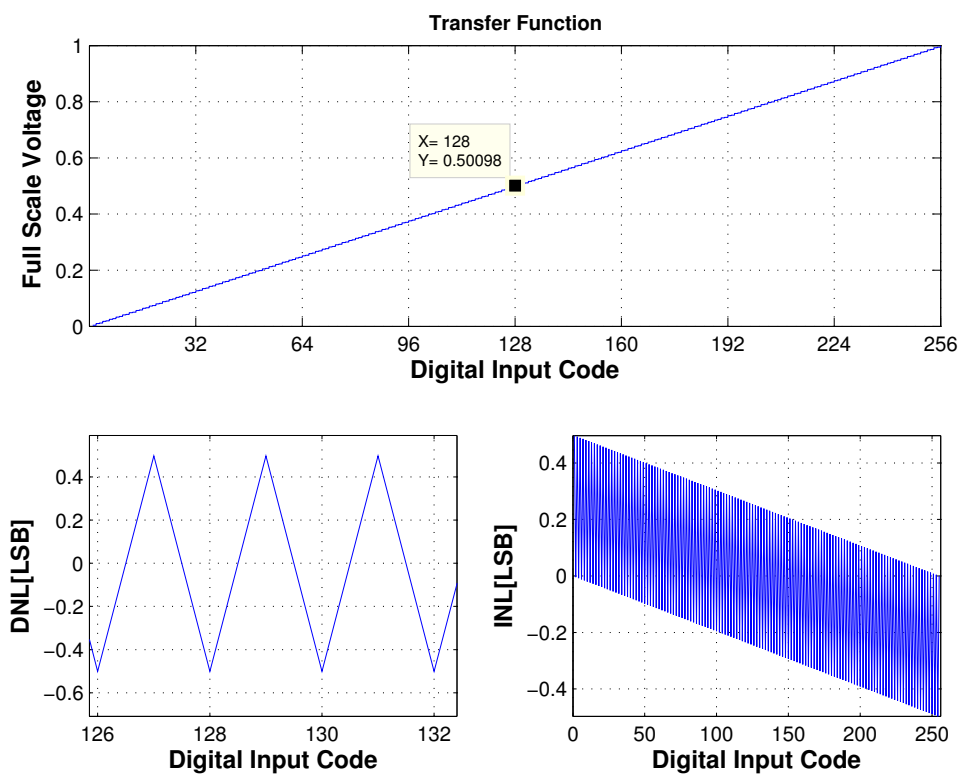
Figure 5.16 : The transfer funtion, DNL and INL plot
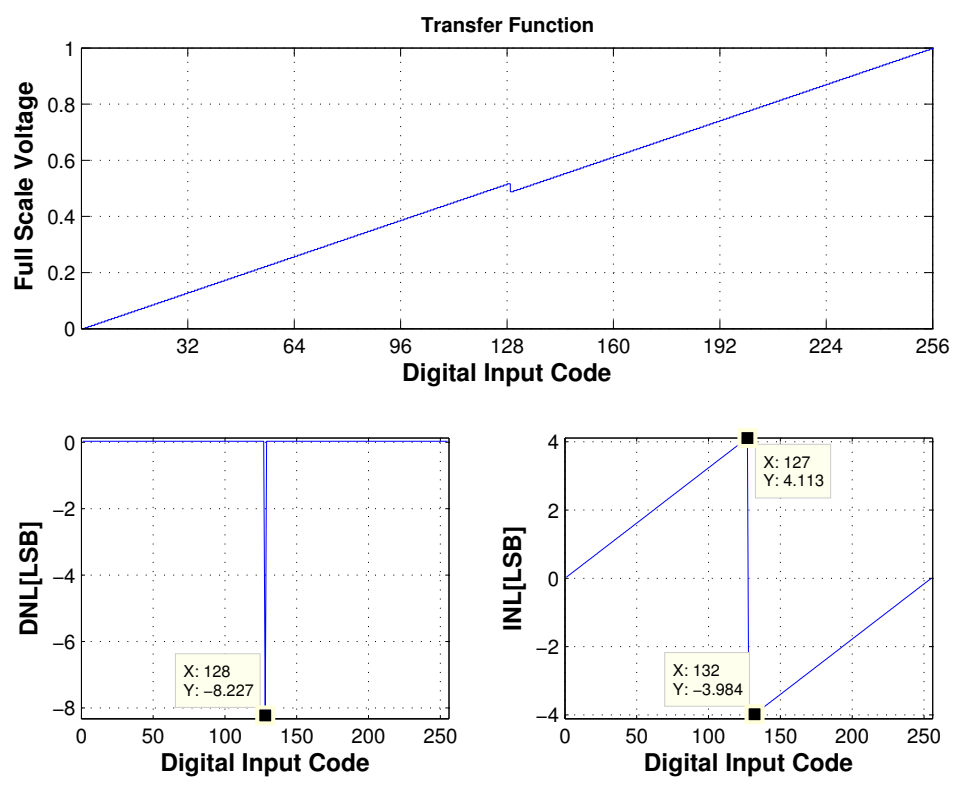
Figure 5.17 : The transfer funtion, DNL and INL plot
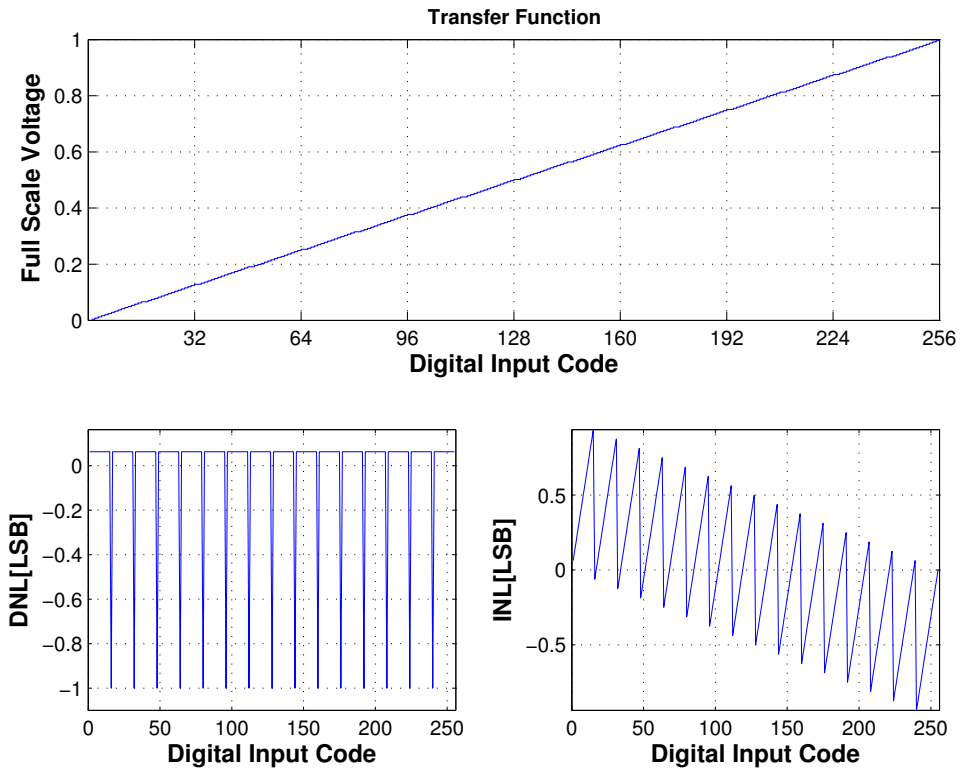
Transfer Function

Figure 5.18 : The transfer funtion, DNL and INL plot
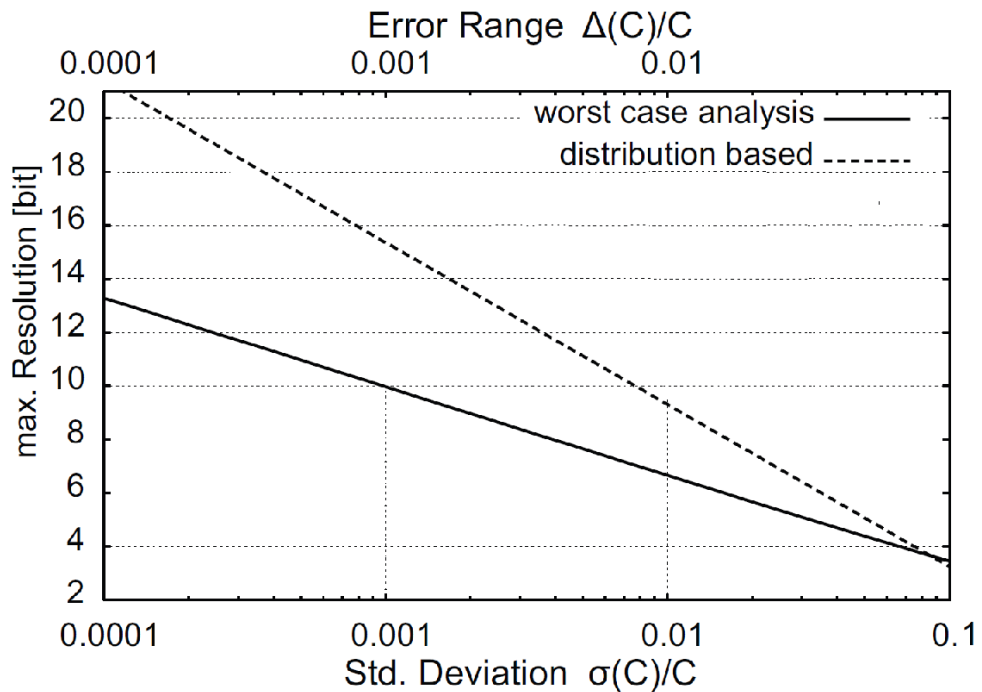
Figure 5.19 : Maximum attainable resolution depending on the matching of an unity capacitor

in the designed ADC has $W = 20\mu$ and $L = 20\mu$ and it leads to $\sigma = 2.25 \times 10^{-4}$ that is enough to reach $N = 14$bit resolution. In addition, a new unit cell design is accomplished in AMS 0.18μ process technology to begin new version of the

SAR-ADC. The layout of the capacity cell is presented in Figure 5.20.
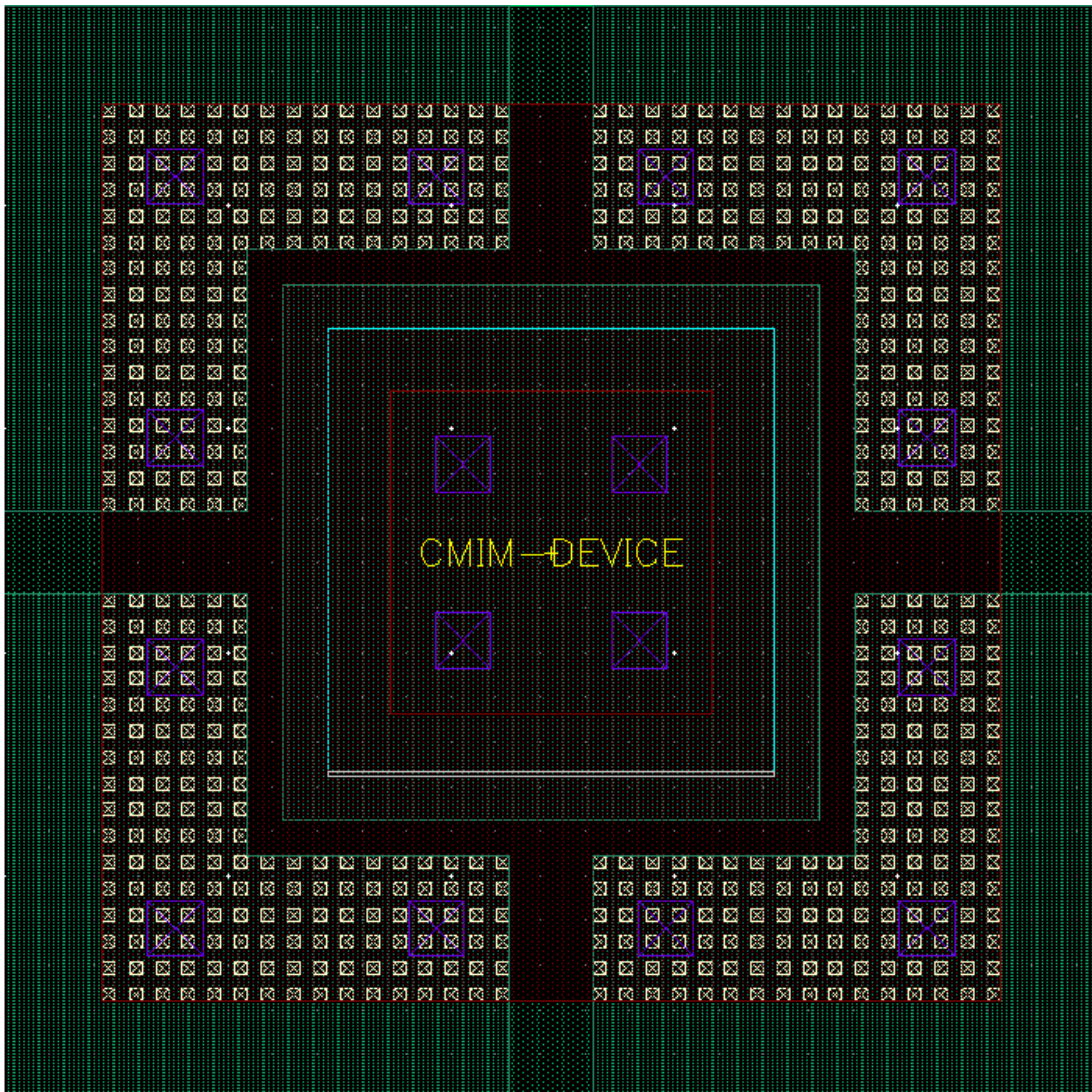
Figure 5.20 : Layout of the unity capacitor

Unity capacitor is comprised of metal-insulator-metal(MIM) that is 30.94fF, $W = 10\mu$, $L = 10\mu$. Top plate is covered by layer AM which is top-metal in the process and bottom plate is covered by layer MT4. Every side of the cell has an entry point that allows to connect top terminals of the capacitor array.

5.2.1.4 SAR logic block

SAR logic block determines the bit values sequantially by using binary search algorithm and then final digital data is stored in the registers and send to the DAC array at the end of the conversion phase. A generic way to implement SAR is using linear feedback shift register that is shown in figure 5.21 (Anderson, 1972). At least $2N$ flip flops are occupied in this architecture.

The architecture based on two rows of flip-flops that are used to produce digital binary data for DAC array. The first row is a shift register that shift logic 1 in each clock rising edge to set sequentially one of the flip flops in the second row. Depending on comparator decision related flip flop in the second row stores the result of comparator. When the last flip flop in the second row is activated, RST signal goes to logic 1 and

Figure 5.21 : A generic SAR block diagram

all flip flops in the first row are reset to start a new conversion.

Table 5.1 shows a detailed description of SAR algorithm. Each conversion takes 9 clock cycle to finalize SAR algorithm. In the cycle 0, all registers that are used to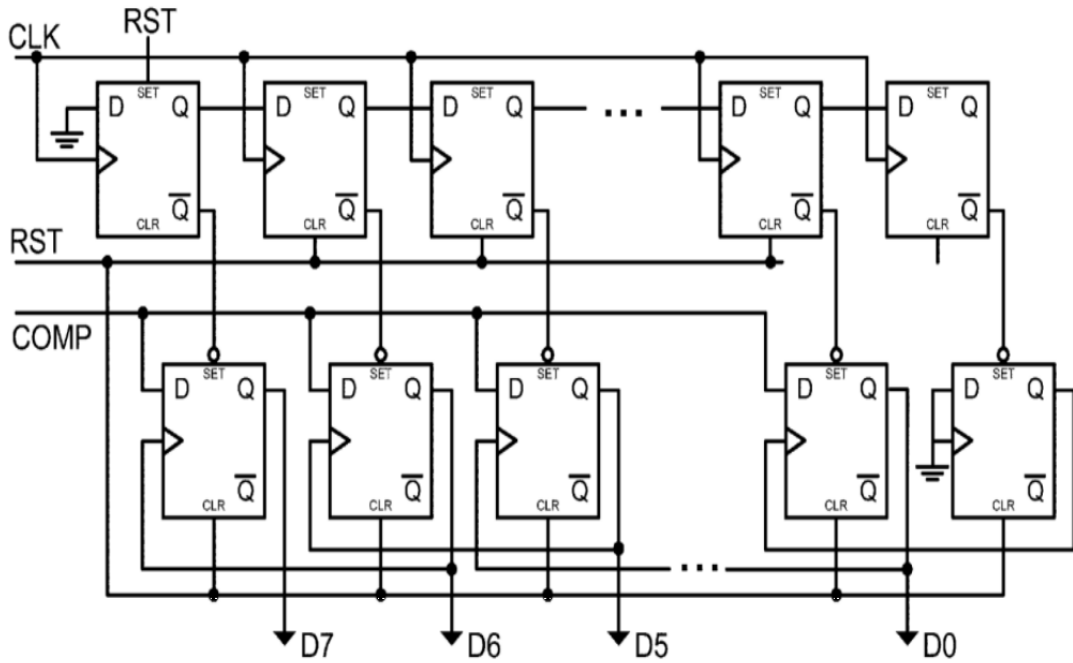 send output data are reset to logic 0. In the next cycle, the flip flop that correspond most significant bit is set to logic 1 to make an initial guess. Then, SAR loads the result of MSB data and again makes a guess logic 1 for next bit. When 8 approximation is made for 8bits, final digital output is send to DAC array. SAR conversion requires 10 clock cycles to generate digital output in every conversion phase.

Table 5.1 : SAR Logic truth table

Cycle	Reset	D7	D6	D5	D4	D3	D2	D1	D0	Comp
0	1	0	0	0	0	0	0	0	0	-
1	0	1	0	0	0	0	0	0	0	a_7
2	0	a_7	1	0	0	0	0	0	0	a_6
3	0	a_7	a_6	1	0	0	0	0	0	a_5
4	0	a_7	a_6	a_5	1	0	0	0	0	a_4
5	0	a_7	a_6	a_5	a_4	1	0	0	0	a_3
6	0	a_7	a_6	a_5	a_4	a_3	1	0	0	a_2
7	0	a_7	a_6	a_5	a_4	a_3	a_2	1	0	a_1
8	0	a_7	a_6	a_5	a_4	a_3	a_2	a_1	1	a_0
9	0	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	−

According to figure 5.21, SAR block requires a counter and a logic that decides the output whether is logic 0 or 1. Figure 5.22 shows the counter that is realized in Simulink.

It starts to count from 0 to 8 that is necessary for SAR logic. A delay function z^-1 is used to wait each clock rising edge.

Figure 5.23 presents the sar model that use 8 subsystem for 8 bit output. Each
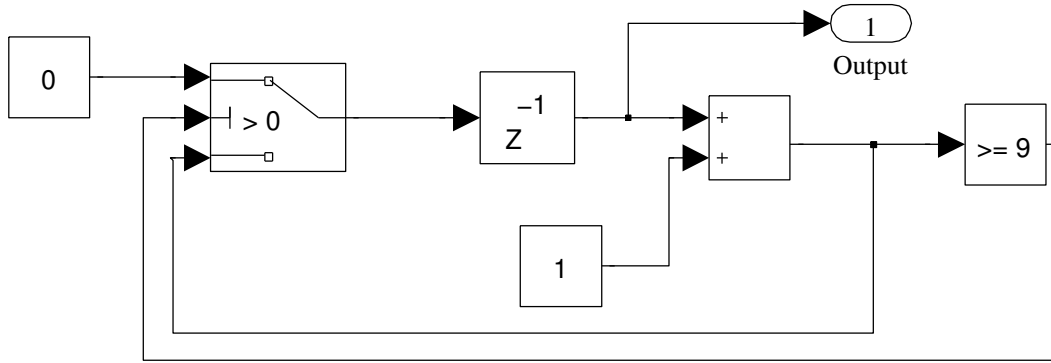
Figure 5.22 : Block diagram of counter
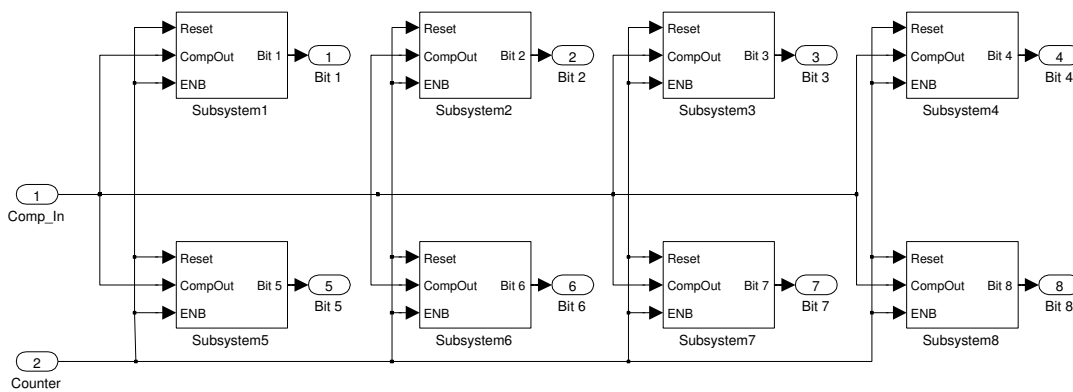
Figure 5.23 : Block diagram of SAR Logic

subsystem has 3 input terminals as Reset, CompOut, ENB and 1 output terminal as BitN. According to input signals, output bit is determined by using logic that is shown in Figure 5.24. Block diagram in figure 5.24 is use the counter output as ENB signal.
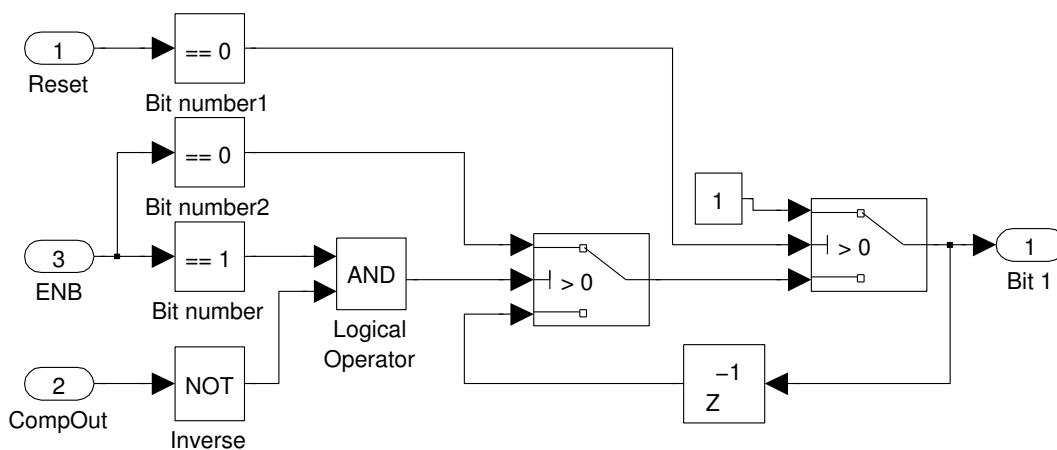
Figure 5.24 : Block diagram of SAR subsystem

Note that Reset and ENB are the same signal since both of them are generated by the counter. However Reset signal is used in the last switch that decides switch condition.

Also there is a delay function which is used to store determined bit value until Reset signal goes logic 1 or ENB signal goes logic 1.

5.2.2 Model Results
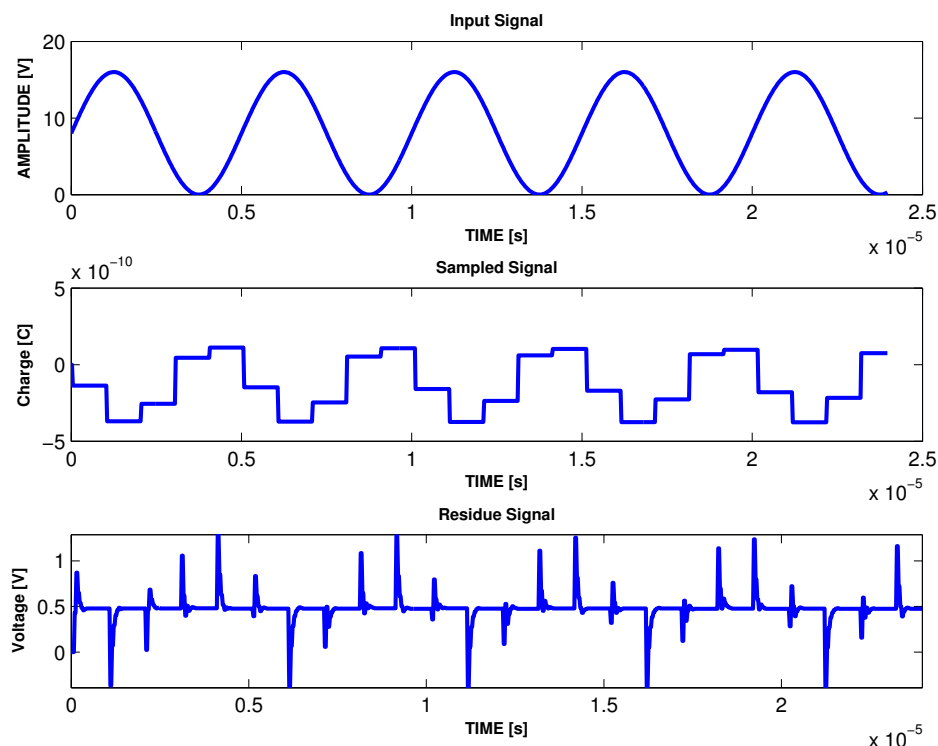
Figure 5.25 : Time domain response of behavioral model

First, the model is simulated without using any non-idealties. Figure 5.25 and 5.26 shows time domain signal response of the modeled ADC for sine wave input signal, sampled charge and first residue charge.

In order to show movement of the signals, simulation time is kept as 25μs for this case. First, 16V input signal is sampled and sampled charge is constant until the next sampling phase. Because of SAR algorithm, residue voltages are changing during the conversion phase. However if the conversion is performed without any mistakes, residue voltage will be smaller than 1LSB voltage which is 976νV

Figure 5.27 shows spectral response of 14bit ideal SAR-ADC model. SNR is 86.02dB, SFDR -117dB and noise floor is at -130dB. There is no any significant spur that will effect linearity of the ADC.

Assume that comparator and amplifier have offset V_{offc} and V_{offa} respectively. At the end of the first 8bit conversion phase, relation between input and output can be stated as:

$$V_{in} = D1 \times V_{LSB} + V_{res1} \tag{5.18}$$

where V_{in} is the analog input voltage, $D1$ is 8bit digital output code, V_{LSB} is LSB voltages of the CDAC and V_{res1} is residue voltage at the end of the first 8 bit conversion state.

Figure 5.26 : Time domain response of behavioral model

Figure 5.27 : Spectral response of behavioral model

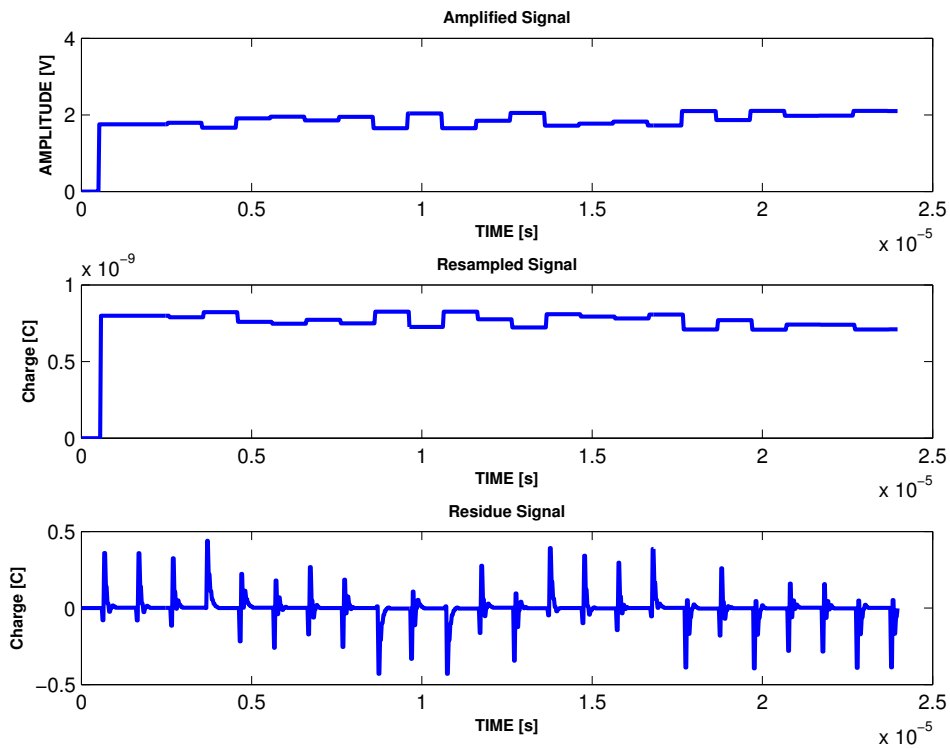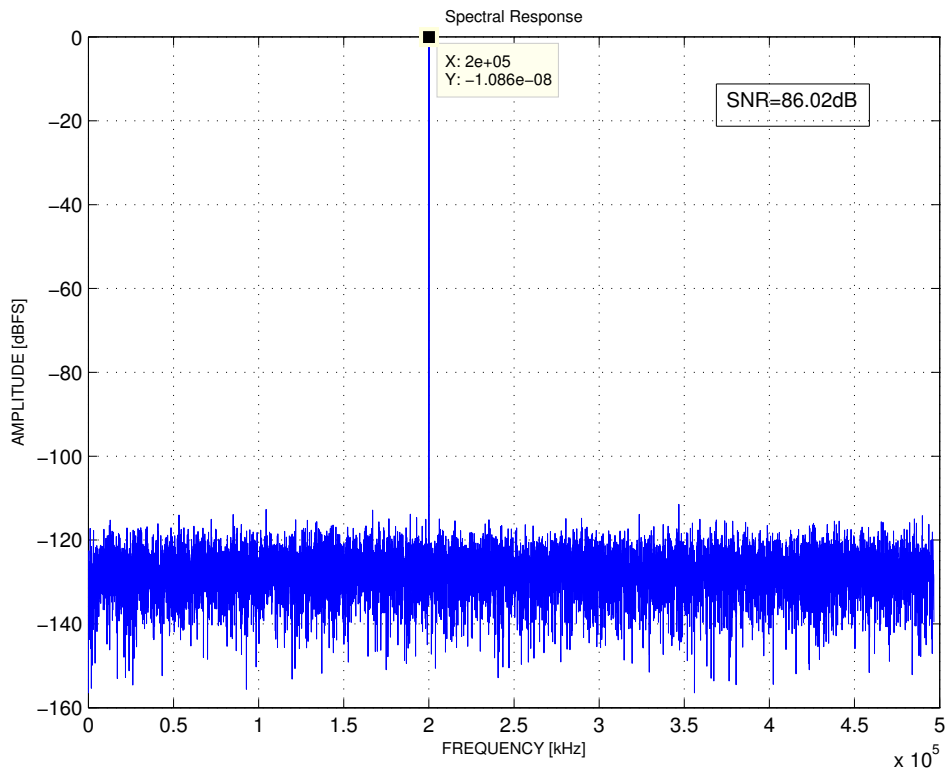Comparator offset affects comparator decision when the difference between CDAC output and input voltage is smaller than comparator offset. It affects digital output code that is obtained at the end of the conversion state.

In the amplification phase, the residue signal is amplified by inter-stage gain of K

$$V_{amp} = K \cdot (V_{res1} - V_{offa})$$ (5.19)

where V_{amp} represent amplified signal. After that, the amplified signal is resampled on the CDAC to perform another 8bit conversion.

$$V_{amp} = D2 \cdot V_{LSB} + V_{res2}$$ (5.20)

where $D2$ is 8bit digital output code and V_{res1} is residue voltage at the end of the second 8bit conversion. To obtain actual digital output, D_{tot} can determined as:

$$D_{tot} = D1 + \frac{D2}{K}$$ (5.21)

If the $D1$ and $D2$ is replaced by using equation (5.18), (5.19), (5.20):

$$D_{tot} \cdot V_{LSB} = (V_{in} - V_{res1}) + \frac{(V_{res1} - V_{offa}) \cdot K - V_{res2}}{K}$$ (5.22)

After making proper simplifications:

$$D_{tot} \cdot V_{LSB} = V_{in} - V_{offa} - \frac{V_{res2}}{K}$$ (5.23)

It is clear that, the relation between the input signal voltage and digital output is linear. The amplifier offset just shift the transfer function. Also, comparator offset cause a shift on residue voltage as amplifier offset.

Figure 5.28 shows variations of the inter-stage gain from 60 to 68. As it expected maximum SNR value is reached when inter-stage gain is 64. The effect of variation is almost symmetrical according to middle point.

A different approach is accomplish by adding measured INL error to capactive DAC array. Figure 5.29 shows spectral response of the ADC with inter-stage gain, K, of 64. SNR is 79.75dB and SFDR is 89.57 which are close to measurement results of the SAR ADC.

Finally, figure **??** shows together measurement and model results of the SAR ADC. Blue and red curves show calculated SNR values versus full scale voltage for model and measured results. In order to effect of inter-stage gain and variaton of capacitive DAC array, related variables are changed and green curve shows SNR values which is close to measurement results until $10V_{pp}$ full scale voltage. Beyond that point, HVBS dominates the performance of the ADC and results SNR degradation. Yellow curve shows measurement results when the capacitive array mismatch and inter-stage gain are changed dramatically.
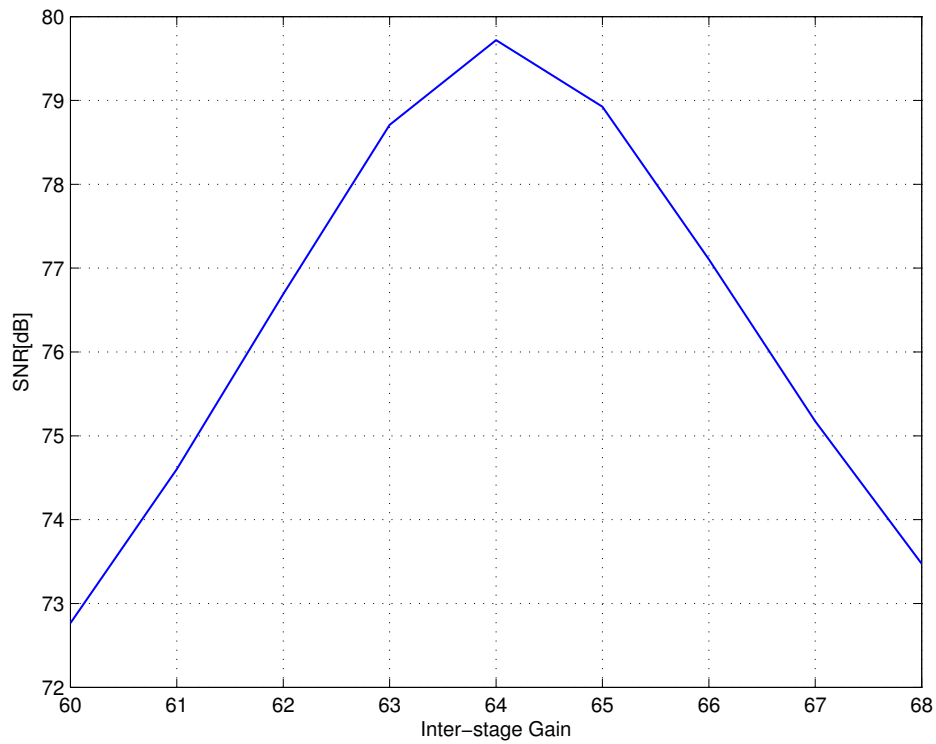
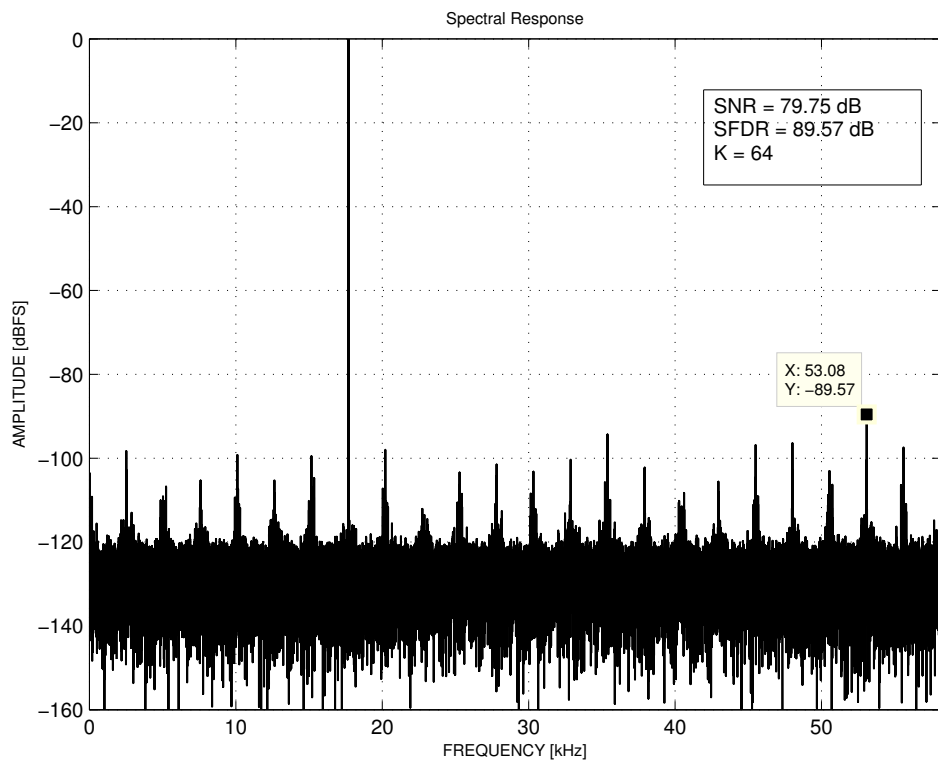**Figure 5.28** : Variaton of inter-stage gain versus SNR

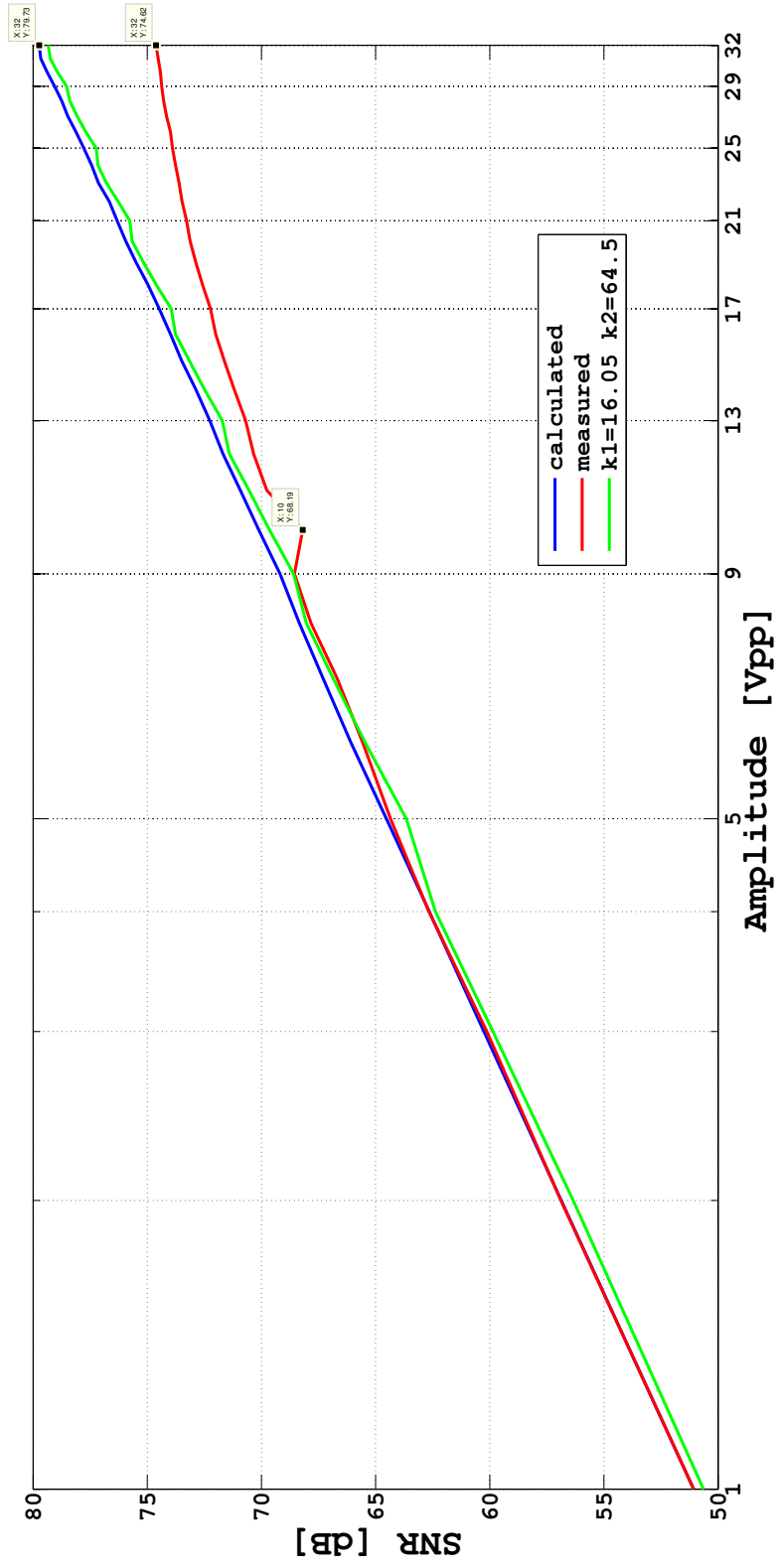**Figure 5.29** : Spectral response of the ADC with measured INL

70

Figure 5.30 : Measurment versus model plot

6. CONCLUSION

In this study, complete behavioral modeling of the two-step SAR ADC architecture with its ideal and non-ideal effects is implemented based on derived transfer function of the converter by using Matlab Simulink blocks.

The model contains main blocks of the SAR ADC which are state machine, comparator, capacitive DAC and SAR logic block. Also, every parameter in the model including input signal, clock signal and reference voltages are stated as variable that can be changed by user at the beginning of the simulation. Therefore, it can be used to implement any N-bit SAR architecture in the literature or market to analyze the effects of non-idealities on the ADC performance.

During the thesis, great efforts are spent to understand transfer functions of the 2step SAR architecture that are derived as charge equations for every state. While realizing transfer functions in Simulink, measurement results are also included to system to verify established model. According to results that are obtained from measurement and model, capacitive DAC array is the most critical part that dominates performance of the ADC. After that comparator is next important block that determines performance of the converter. Since it is a 2step architecture, inter-stage gain is another vital parameter that defines characteristic of the ADC. Variation on the inter-stage gain cause SNR degradation and DNL-INL error at the output signal. The model

Simulation time is incredibly short in the established model. Thus, it offers a fast solution for full system simulation that take long simulation time while saving time. In addition, specification for sub-blocks of the ADC can be found by using the model. That gives opinion to the designer about demanded ADC is achievable or not for given technology.

The realized behavioral model will be a good start point for a designer who is dealing with SAR ADC. It will help to understand the basic principles and investigate complex behaviour of the ADC by using top-down methodology. Since all parameters are left as an input variable, it can be easily used for any SAR structure. Most of all, based on the presented work, transistor-level simulation time that is spend during design of a SAR ADC will be shorten.

References

Aksin, D., & Ozkaya, I. (2009, Sept). 25v sampling switch for power management data converters in 0.35u cmos with dnmos. In *Esscirc, 2009. esscirc '09. proceedings of* (p. 136-139).

Anderson, T. O. (1972, November). Optimum Control Logic for Successive Approximation Analog-to-Digital Converters. *Deep Space Network Progress Report*, *13*, 168-176.

Baker, B. (2006). *A glossary of analog-to-digital specifications and performance characteristics* (Application Report No. SBAA147B). United States: Texas Instruments.

Bennett, W. R. (1948, July). Spectra of quantized signals. *Bell System Technical Journal*, *27*, 446-471.

Fredenburg, J., & Flynn, M. (2012, July). Statistical analysis of enob and yield in binary weighted adcs and dacs with random element mismatch. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, *59*(7), 1396-1408. doi: 10 .1109/TCSI.2011.2177006

Haenzsche, S., Henker, S., & Schuffny, R. (2010, June). Modelling of capacitor mismatch and non-linearity effects ini charge redistribution sar adcs. In *Mixed design of integrated circuits and systems (mixdes), 2010 proceedings of the 17th international conference* (p. 300-305).

Ilter, O. (2010). *Wide input signal range 14 bits 1msps sar adc in 0.35um high voltage cmos process* (Master Thesis). Istanbul Technical University.

Jarman, D. (1995). *A brief introduction to sigma delta conversion* (AN No. 9504). United States: Interstil Corporation.

Kester, W. (2007). *Analog-digital conversion*. Dordrecht, the USA: Analog Devices.

King, I. (2011). *Understanding high-speed signals, clocks, and data capture* (AN No. 103). Dallas, United States: Texas Instruments.

Lin, Z., Yang, H., Zhong, L., Sun, J., & Xia, S. (2005, Oct). Modeling of capacitor array mismatch effect in embedded cmos cr sar adc. In *Asic, 2005. asicon 2005. 6th international conference on* (Vol. 2, p. 982-986). doi: 10.1109/ICASIC.2005 .1611492

Lundberg, K. H. (2002). Analog-to-digital converter testing.

Malcovati, P., Brigati, S., Francesconi, F., Maloberti, F., Cusinato, P., & Baschirotto, A. (2003, Mar). Behavioral modeling of switched-capacitor sigma-delta modulators. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, *50*(3), 352-364. doi: 10.1109/TCSI.2003 .808892

Maloberti, F. (2007). *Data converters*. Dordrecht, The Netherlands: Springer.

Maxim Integrated. (2001a). *Histogram testing determines dnl and inl errors* (AN No. 2085). United States.

Maxim Integrated. (2001b). *Inl/dnl measurements for high-speed analog-to-digital converters (adcs)* (AN No. 283). United States.

Maxim Integrated. (2001c). *Understanding sar adcs: Their architecture and comparison with other adcs* (AN No. 1080). United States.

Maxim Integrated. (2002a). *Adc and dac glossary* (AN No. 641). United States: Maxim Integrated.

Maxim Integrated. (2002b). *Coherent sampling vs. window sampling* (AN No. 1040). United States.

Osipov, D., & Bocharov, Y. (2012, June). Behavioral model of split capacitor array dac for use in sar adc design. In *Ph.d. research in microelectronics and electronics (prime), 2012 8th conference on* (p. 1-4).

Ozkaya, I., Gurleyuk, C., Ergul, A., Akkaya, A., & Aksin, D. (2014, Sept). A 50v input range 14bit 250ks/s adc with 97.8db sfdr and 80.2db snr. , 71-74. doi: 10.1109/ESSCIRC.2014.6942024

P. Carbone, F. X., S. Kiaei. (2014). *Design, modeling and testing of data converters*. Dordrecht, The Netherlands: Springer.

Razavi, B. (2001). *Design of analog cmos integrated circuits* (1st ed.). New York, NY, USA: McGraw-Hill, Inc.

Sadkowski, R. (2013). *Process high-voltage input levels into a low-voltage adc without losing much snr* (APP No. 5282). United States: Maxim Integrated.

Salah Hanfoug, N.-E. B., & Barra, S. (2014). Behavioral non-ideal model of 8-bit current-mode successive approximation registers adc by using simulink. *International Journal of u- and e- Service, Science and Technology*.

Wong, S.-S., Zhu, Y., Chan, C.-H., Chio, U.-F., Sin, S.-W., U, S.-P., & Martins, R. (2009, Nov). Parasitic calibration by two-step ratio approaching technique for split capacitor array sar adcs. In *Soc design conference (isocc), 2009 international* (p. 333-336). doi: 10.1109/SOCDC.2009.5423780

Zhang, D. (2009). *Design and evaluation of an ultra-low power successive approximation adc* (Master Thesis). Linköping Institute of Technology.

APPENDICES

APPENDIX A.1

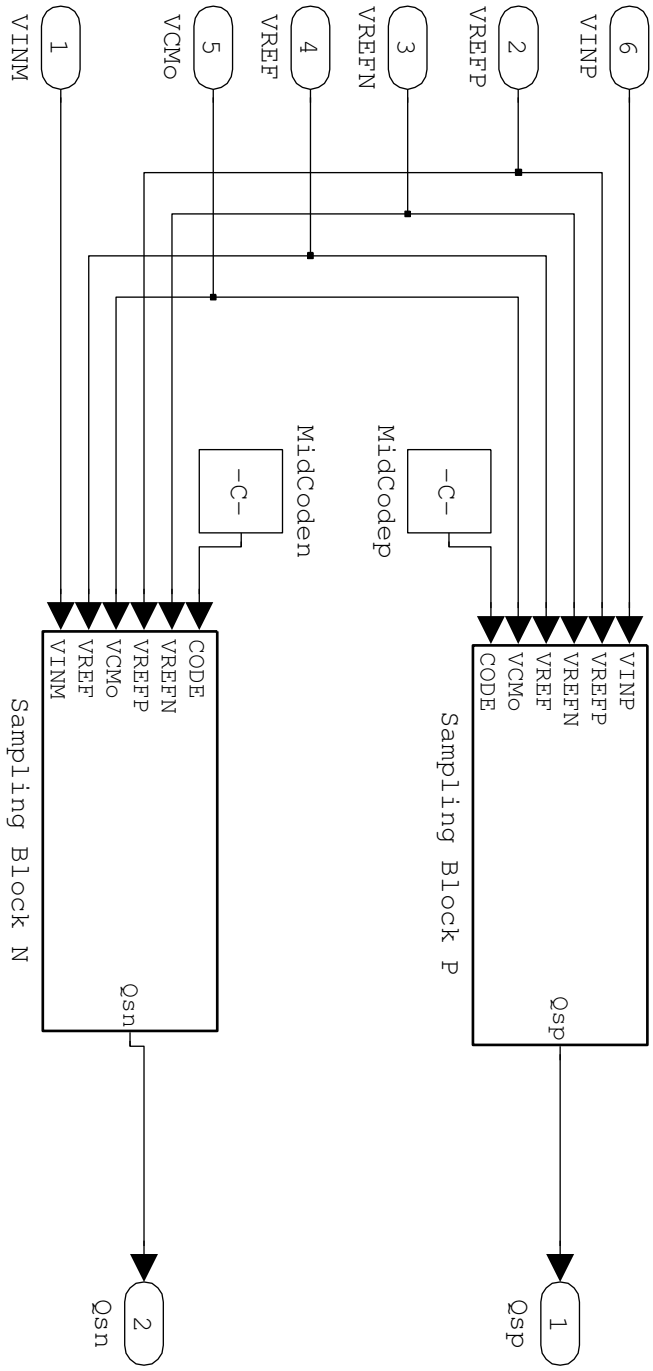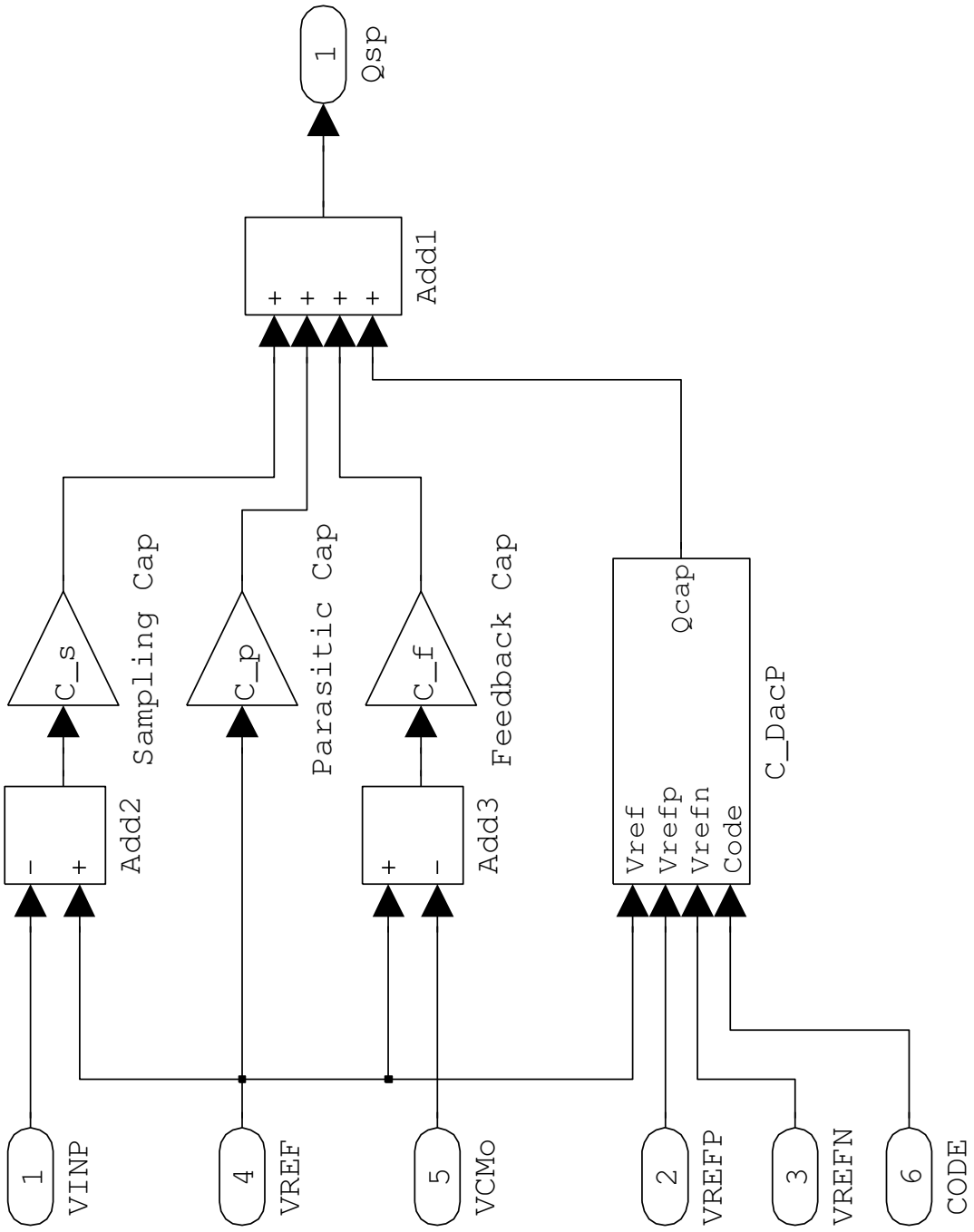**Figure A.1** : Sampling block of the ADC

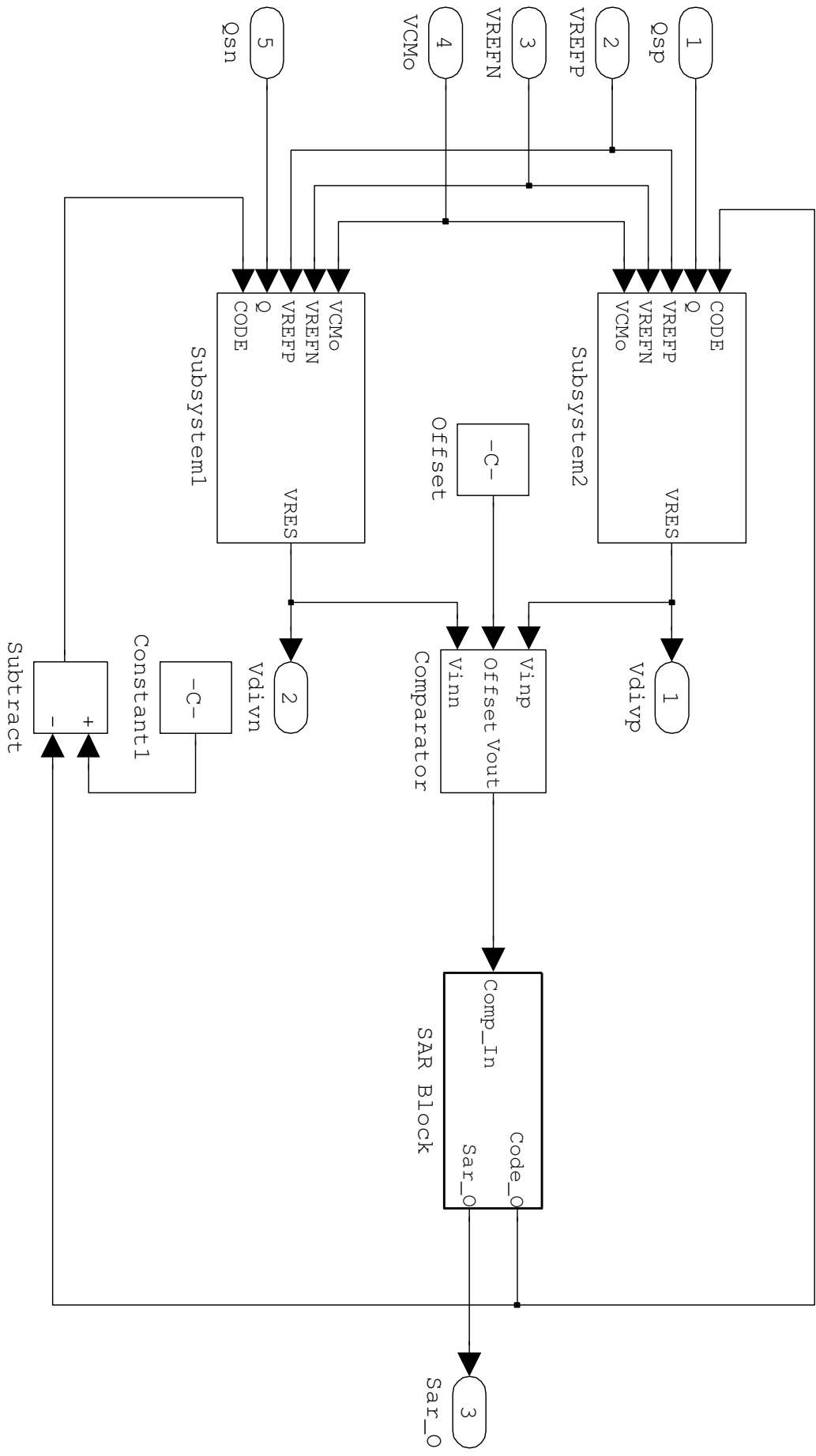**Figure A.2** : Sampling subblock of the ADC

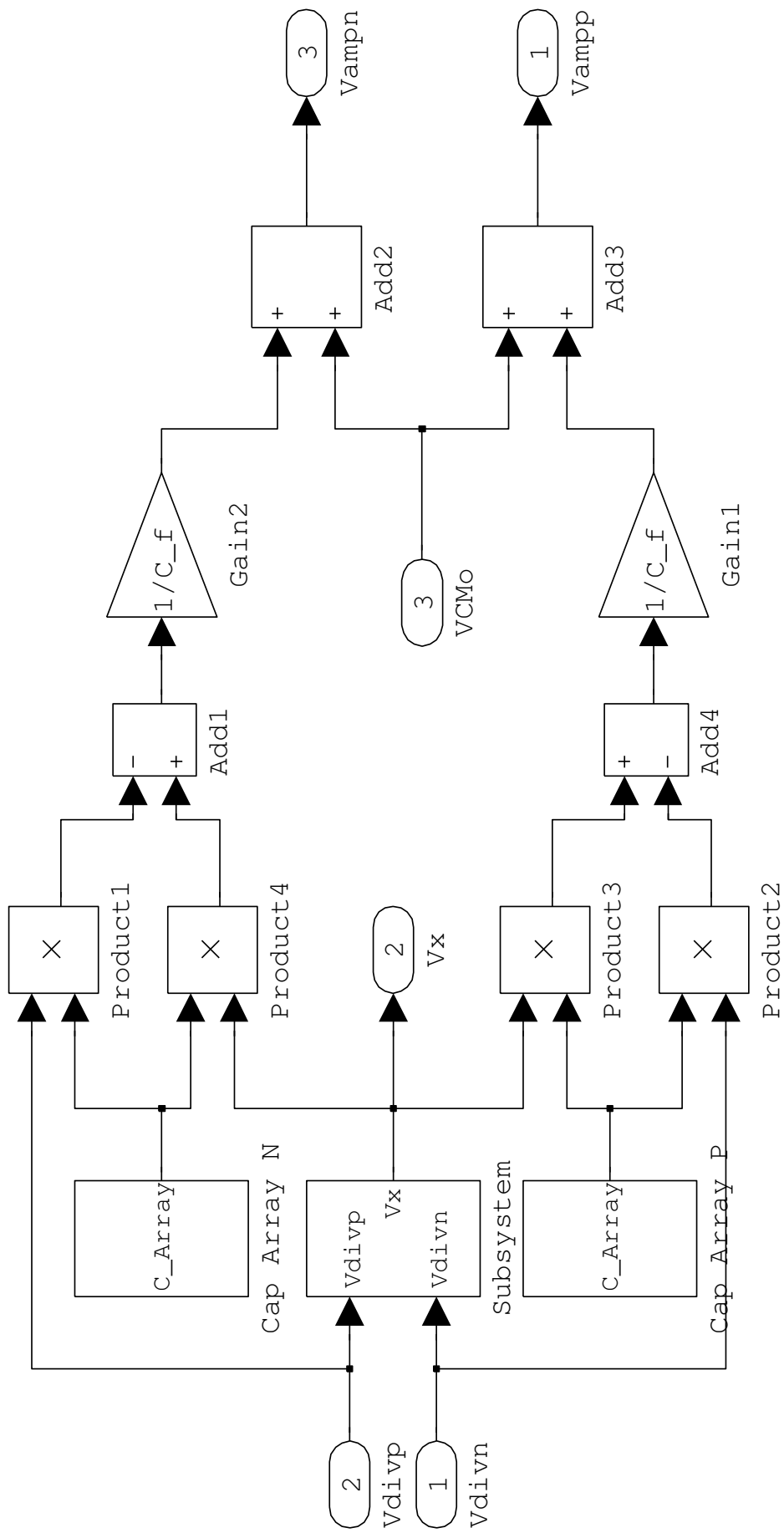**Figure A.3** : SAR Block of the ADC

Figure A.4 : Amplification block of the ADC

83
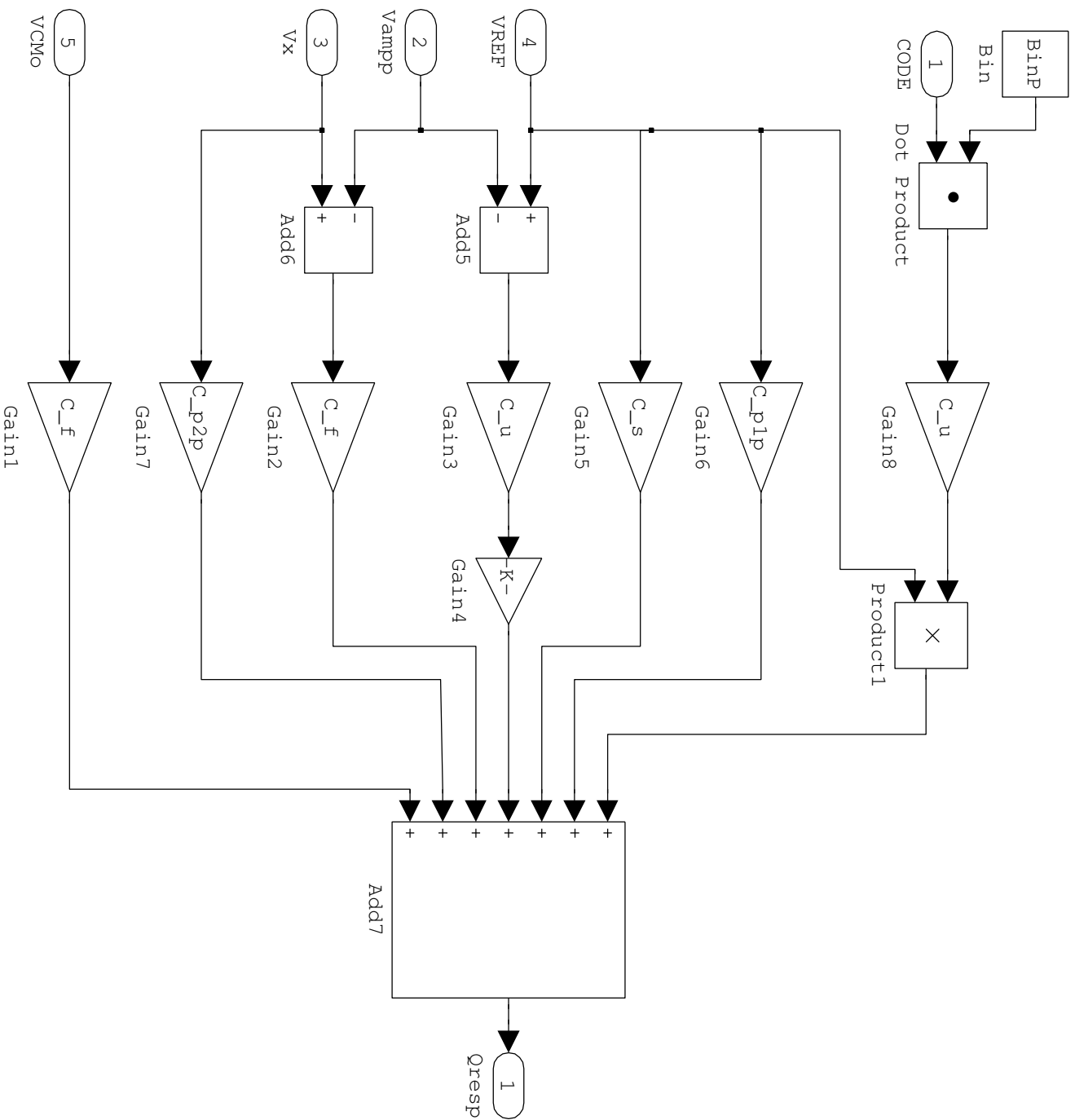
Figure A.5 : Resampling block of the ADC

84

APPENDIX A.2

```matlab
function [snrdB,ptotdB,psigdB,pnoisedB] = calcSNR(vout,f,fB,w,N,Vref)
% SNR calculation in the time domain (P. Malcovati, S. Brigati)
% vout: Sigma-Delta bit-stream taken at the modulator output
% f:  Normalized signal frequency (fs -> 1)
% fB: Base-band frequency bins
% w: windowing vector
% N:  samples number
% Vref: feedback reference voltage
%
% snrdB:  SNR in dB
% ptotdB:  Bit-stream power spectral density (vector)
% psigdB: Extracted signal power spectral density (vector)
% pnoisedB: Noise power spectral density (vector)
%
fB=ceil(fB);
signal=(N/sum(w))*sinusx(vout(1:N).*w,f,N); % Extracts sinusoidal signal
signal2=(N/sum(w))*sinusx(vout(1:N).*w,2*f,N);
signal3=(N/sum(w))*sinusx(vout(1:N).*w,3*f,N);
signal4=(N/sum(w))*sinusx(vout(1:N).*w,4*f,N);
signal5=(N/sum(w))*sinusx(vout(1:N).*w,5*f,N);


noise=vout(1:N)-signal-signal2-signal3-signal4-signal5 % Extracts noise components
signal=signal+signal2+signal3+signal4+signal5

stot=((abs(fft((vout(1:N).*w)'))).^2);  % Bit-stream PSD
ssignal=(abs(fft((signal(1:N).*w)'))).^2; % Signal PSD
snoise=(abs(fft((noise(1:N).*w)'))).^2;  % Noise PSD
pwsignal=sum(ssignal(1:fB));    % Signal power
pwnoise=sum(snoise(1:fB));      % Noise power
snr=pwsignal/pwnoise;
snrdB=dbp(snr);
norm=sum(stot)/Vref^2;         % PSD normalization

if nargout > 1
  ptot=stot/norm;
   ptot=ptot+eps;
  ptotdB=dbp(ptot);
end

if nargout > 2
  psig=ssignal/norm;
  psigdB=dbp(psig);
end

if nargout > 3
  pnoise=snoise/norm;
  pnoisedB=dbp(pnoise);
end


function y=dbp(x)
% dbp(x) = 10*log10(x): the dB equivalent of the power x
y = -Inf*ones(size(x));
nonzero = x~=0;
y(nonzero) = 10*log10(abs(x(nonzero)));

function y=dbv(x)
% dbv(x) = 20*log10(abs(x)); the dB equivalent of the voltage x
y = -Inf*ones(size(x));
nonzero = x~=0;
y(nonzero) = 20*log10(abs(x(nonzero)));

function outx = sinusx(in,f,n)
%
% Extraction of a sinusoidal signal
%
sinx=sin(2*pi*f*[1:n]);
cosx=cos(2*pi*f*[1:n]);
in=in(1:n);
a1=2*sinx.*in;
a=sum(a1)/n;
b1=2*cosx.*in;
b=sum(b1)/n;
outx=a.*sinx + b.*cosx;
```

APPENDIX A.2

```matlab
clc;
clear;
close all;
%% Variables
InputSig = 1;              % Input Signal 1 for Sine Wave -1 for Ramp
CapModel = 0;              % Cap Array 1 for with attenutor -1 for binary cap array

FIN     = 200e3;           % Input Signal Frequency
AIN     = 16;              % Input Signal Amplitude
VINCM   = 8;               % Input Signal Common Mode

FCONV   = 1e6;             % Sampling Frequency
NFFT    = 2^14;            % FFT point

FFTCycle = FIN*NFFT/FCONV; % Number of Cycle

while ~isprime(round(FFTCycle)) % Coherent Sampling Check
    FFTCycle = round(FFTCycle)+1;
end

FCONV   = FIN*NFFT/FFTCycle; % Sampling Frequency

ACLK     = 1;              % Clock Amplitude
NSAMPLE = 1;               % Sampling Time
NSAR1   = 9;               % First SAR Conv Time
NAMPLIFY = 1;              % Amplification Time
NRESAMPLE = 1;             % Resampling Time
NSAR2   = 9;               % Second SAR Conv Time
NTOT    = NSAMPLE+NSAR1+NAMPLIFY+NRESAMPLE+NSAR2; % Total Conv Time
FCLK    = FCONV*NTOT;      % Clock Frequency

VREFP = 2;                 % Positive Reference Voltage
VREFN = 0;                 % Negative Reference Voltage
VREF  = 3;                 % Reference Voltage
VCMo  = 1.65;              % Output Common Mode Voltage
K     = 64;                % Interstage Gain

C_u   = 1e-12;                 % unit capacitance
C_ap  = C_u*1;                 % attenuator capacitance on P Node
C_an  = C_u*1;                 % attenuator capacitance on N Node

Bin = [2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0]; % Cap Array
C_misP = 0;
C_misN = 0;

if(CapModel == 1)
    Bin = [2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0]; % Cap Array with Attenuator
    BinP = Bin+C_misP*0;                 % Cap Array on P Node
    BinN = Bin+C_misN*0;                 % Cap Array on N Node

    BinP(5:8) = sum(BinP(5:8))*C_u*C_ap/(sum(BinP(5:8))*C_u+C_ap)/(sum(BinP(5:8))*C_u)*BinP(5:8); % Attenuator Cap Effect on Node P
    BinN(5:8) = sum(BinN(5:8))*C_u*C_an/(sum(BinN(5:8))*C_u+C_an)/(sum(BinN(5:8))*C_u)*BinN(5:8); % Attenuator Cap Effect on Node N
else
    BinP = Bin;
    BinN = Bin;
end

C_s   = sum(Bin)*VREFP/AIN*C_u; % sampling capacitance
C_f   = sum(Bin)*C_u/(K-1); % feedback capacitance

C_p1p = 0.0*C_u;           % parasitics capacitance
C_p2p = 0.0*C_u;           % parasitics capacitance
C_p1n = 0.0*C_u;           % parasitics capacitance
C_p2n = 0.0*C_u;           % parasitics capacitance

C_p   = C_p1p+C_p2p;       % parasitics cap on node p
C_n   = C_p1n+C_p2n;       % parasitics cap on node p

int_gain = (sum(Bin)*C_u+C_f)/C_f; % interstage gain

%% Simulation
if(InputSig == 1)
    stoptime = FFTCycle/FIN+10/FCONV; % simulation stop time
    stoptime = 19/FCONV
else
    stoptime = 2^8*32/FCONV; % simulation stop time
    VINCM  = 0;
end
slope  = AIN/stoptime;       % slope of the ramp signal

sim('bit14Sar.mdl');         % 14 bit SAR model

%% FFT
LSB = AIN/(2^8-1);           % LSB value
if(InputSig == 1)

    % 14 bit SAR FFT
    Q8bit = simout.signals.values;
    Q8bit = Q8bit(2:size(Q8bit));

    Q6bit = simout1.signals.values;
    Q6bit = Q6bit(2:size(Q6bit));

    Q = Q8bit(1:NFFT)+Q6bit(1:NFFT);
```

```matlab
    % FFT calc SNR

    Q = Q-mean(Q);
    w = rectwin(NFFT);

    % SNR
    [SNR,ptotdB,psigdB,pnoisedB] = calcSNR(Q', FIN./FCONV, NFFT/2, w', NFFT, sqrt(2));
    f = FCONV/2*linspace(0, 1, NFFT/2);

    plot(f,ptotdB(1:NFFT/2))
    xlabel('FREQUENCY [kHz]','Fontsize',8)
    ylabel('AMPLITUDE [dBFS]','FontSize',8)
    title('Spectral Response','FontSize',8,'FontWeight','normal'))
    grid on
    SNR
end

%% INL DNL
if(InputSig==0)
    Outhist14bit = hist(Q-95,2^14);
    Idealhist = 32;

    dnl = 1-Outhist14bit/mean(Outhist14bit);
    dnl = dnl(2:16383);
    dnl14bit = dnl-mean(dnl);

    inl14bit = cumsum(dnl14bit);

    Outhist8bit = hist(Q8bit,2^8);
    Idealhist = 32*64;

    dnl = 1-Outhist8bit/mean(Outhist8bit);
    dnl = dnl(2:255);
    dnl = dnl-mean(dnl);

    inl = cumsum(dnl);

    figure('name','DNL','numbertitle','off')

    subplot(2,1,1)
    plot(dnl)
    title('DNL TFC, N = 8');
    xlabel('Code');
    ylabel('LSB');

    subplot(2,1,2)
    plot(dnl14bit)
    title('DNL TFC, N = 14');
    xlabel('Code');
    ylabel('LSB');


    figure('name','INL','numbertitle','off')

    subplot(2,1,1)
    plot(inl);
    title('INL TFC, N = 8');
    xlabel('Code');
    ylabel('LSB');

    subplot(2,1,2)
    plot(inl14bit);
    title('INL TFC, N = 14');
    xlabel('Code');
    ylabel('LSB');
end
```

APPENDIX A.3

```matlab
clc;
clear all;
close all;
load dnl_inl.mat;        % Measured INL-DNL
load list;               % Amplitude Array

%% Parameters
N = 14;                  % Input Bit Number
k1 = 1/16;               % First Gain Stage
k2 = 1/64;               % Seconde Gain Stage

% input signals
input  = 0:2^N-1;        % N Bit Input
digcode = de2bi(input,'left-msb');

% input sine signal
F_s = 5e6/43;            % Sampling Frequency
T_s = 1/F_s;             % Sampling Period
N_points = 2^16;         % Number of Points
F_in = 9973*F_s/N_points; % Input Frequency
% F_in = 17e3
noise = wgn(N_points,1,-93.58);
% load noise.mat;
sigma = std(noise);

list = list/64.75;       % Normalization of Amplitude

A     = 0.5;             % Amplitude
Vcm   = 0.5;             % Common mode Voltage
t     = (1:N_points)*T_s; % time
x     = A*sin(2*pi*F_in*t); % sine wave

amp1 = 10^(108.4/20);
amp2 = 10^(51.9/20);
amp3 = 10^(96.33/20);
amp4 = 10^(-18.1/20);

a1 = 1;
a2 = amp2/amp1;
a2 = 0;
a3 = 100*30*amp4/amp3/6.32;
a3 = 0;

y = Vcm+a1*x+a2*x.^2+a3*x.^3+noise';
% y = awgn(y,93.58);

% capacitor array
A1 = [8 4 2 1];
A2 = A1.*k1;
A3 = A1.*k2;
A4 = A1.*k1.*k2;

DWM1 = [A1';A2'];
DWM2 = [A3(3:4)';A4'];
DWM3 = [DWM1;DWM2];

% initialize variables
tfc = zeros(2^N,1);
q   = zeros(N_points,1);

%% transfer function

tfc = digcode*DWM3;
tfc = tfc/max(tfc);   % normalization of
LSB1 = max(tfc)/(2^N-1);
% addition of inl
tfc(2:(2^N-1)) = tfc(2:(2^N-1)) + LSB1.*meas_inl'*0;

%% quantization of sine wave

for i = 1:length(y)
    if(y(i)>0)
        q(i) = find(y(i) >= tfc, 1, 'last');
    else
        q(i) = find(y(i) <= tfc, 1, 'first');
    end
end

q = q - 1;
LSB2 = max(q)/(length(q)-1);

%% snr calculation

y = y-mean(y);
q = q-mean(q);

L = N_points;
w = rectwin(L);
```

```matlab
[SNR,ptotdB,psigdB,pnoisedB] = calcSNR(q', F_in./F_s, L/2, w', N_points, 1);
SNR
% SNRA(d) = SNR
f = F_s/2*linspace(0, 1, N_points/2);

%% fft
figure
L = N_points;
w = rectwin(L);

Q = fft(q(1:N_points),N_points)./N_points;
Q = 20*log10(abs(Q));
Q = Q-max(Q);
plot(Q)
%%
figure

Q = fft(y(1:N_points),N_points)./N_points;
Q = 20*log10(abs(Q));
Q = Q-max(Q);

plot(f/1000,Q(2:N_points/2+1),'Linewidth',1,'Color','k')
xlabel('FREQUENCY [kHz]','Fontsize',8)
ylabel('AMPLITUDE [dBFS]','FontSize',8)
title('Spectral Response','FontSize',8,'FontWeight','normal')
axis ([min(f)/1000 max(f)/1000 -120 0])
grid on
```

APPENDIX A.4

```matlab
clc;
clear;
close all;
%% Variables
InputSig = 0;              % Input Signal 1 for Sine Wave -1 for Ramp
CapModel = 0;              % Cap Array 1 for with attenutor -1 for binary cap array

FIN     = 200e3;          % Input Signal Frequency
AIN     = 16;             % Input Signal Amplitude
VINCM   = 8;              % Input Signal Common Mode

FCONV   = 1e6;            % Sampling Frequency
NFFT    = 2^14;           % FFT point

FFTCycle = FIN*NFFT/FCONV; % Number of Cycle

while ~isprime(round(FFTCycle)) % Coherent Sampling Check
    FFTCycle = round(FFTCycle)+1;
end

FCONV   = FIN*NFFT/FFTCycle; % Sampling Frequency

ACLK    = 1;              % Clock Amplitude
NSAMPLE = 1;              % Sampling Time
NSAR1   = 9;              % First SAR Conv Time
NAMPLIFY = 1;             % Amplification Time
NRESAMPLE = 1;            % Resampling Time
NSAR2   = 9;              % Second SAR Conv Time
NTOT    = NSAMPLE+NSAR1+NAMPLIFY+NRESAMPLE+NSAR2; % Total Conv Time
FCLK    = FCONV*NTOT;     % Clock Frequency

VREFP = 2;               % Positive Reference Voltage
VREFN = 0;               % Negative Reference Voltage
VREF  = 3;               % Reference Voltage
VCMo  = 1.65;            % Output Common Mode Voltage
K     = 64;              % Interstage Gain
offset = 0
C_u    = 1e-12;              % unit capacitance
C_ap  = C_u*1;               % attenuator capacitance on P Node
C_an  = C_u*1;               % attenuator capacitance on N Node

Bin = [2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0]; % Cap Array
C_misP = 0;
C_misN = 0;

if(CapModel == 1)
    Bin = [2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0]; % Cap Array with Attenuator
    BinP = Bin+C_misP*0;                  % Cap Array on P Node
    BinN = Bin+C_misN*0;                  % Cap Array on N Node

    BinP(5:8) = sum(BinP(5:8))*C_u*C_ap/(sum(BinP(5:8))*C_u+C_ap)/(sum(BinP(5:8))*C_u)*BinP(5:8); % Attenuator Cap Effect on Node P
    BinN(5:8) = sum(BinN(5:8))*C_u*C_an/(sum(BinN(5:8))*C_u+C_an)/(sum(BinN(5:8))*C_u)*BinN(5:8); % Attenuator Cap Effect on Node N
else
    BinP = Bin;
    BinN = Bin;
end

C_s    = sum(Bin)*VREFP/AIN*C_u; % sampling capacitance
C_f    = sum(Bin)*C_u/(K-1); % feedback capacitance

C_p1p  = 0.0*C_u;           % parasitics capacitance
C_p2p  = 0.0*C_u;           % parasitics capacitance
C_p1n  = 0.0*C_u;           % parasitics capacitance
C_p2n  = 0.0*C_u;           % parasitics capacitance

C_p    = C_p1p+C_p2p;       % parasitics cap on node p
C_n    = C_p1n+C_p2n;       % parasitics cap on node p

int_gain = (sum(Bin)*C_u+C_f)/C_f; % interstage gain

%% Simulation
if(InputSig == 1)
    stoptime = FFTCycle/FIN+10/FCONV; % simulation stop time
%    stoptime = 500/FCLK
else
    stoptime = 2^8*1/FCONV; % simulation stop time
    VINCM  = 0;
end
slope  = AIN/stoptime;        % slope of the ramp signal

sim('bit14Sarv5.mdl');        % 14 bit SAR model

%% FFT
LSB = AIN/(2^8-1);            % LSB value
if(InputSig == 1)
    % 8 bit SAR FFTsynchronous
    % Q8bit = simout.signals.values;
    % Q8bit = Q8bit(2:size(Q8bit));
```

91

```matlab
    % Q = Q8bit(1:NFFT);

    % 14 bit SAR FFT
%   Q8bit = simout.signals.values;
%   Q8bit = Q8bit(2:size(Q8bit));

%   Q6bit = simout1.signals.values;
%   Q6bit = Q6bit(2:size(Q6bit));

    Q = Q8bit(2:NFFT+1)+Q6bit(2:NFFT+1)/K;

    % FFT calc SNR

    Q = Q-mean(Q);
    w = rectwin(NFFT);

    % [SNR, Y_sig, Y_noi, Y_tot] = calculateSNR(Q(1:NFFT), FIN/FCONV, NFFT/2, NFFT, 1);
    % SNR
    [SNR,ptotdB,psigdB,pnoisedB] = calcSNR(Q', FIN./FCONV, NFFT/2, w', NFFT, sqrt(2));
    f = FCONV/2*linspace(0, 1, NFFT/2);

    plot(f,ptotdB(1:NFFT/2))
    xlabel('FREQUENCY [kHz]','Fontsize',8)
    ylabel('AMPLITUDE [dBFS]','FontSize',8)
    title('Spectral Response','FontSize',8,'FontWeight','normal')
%   axis ([min(f)/1000 max(f)/1000 -120 0])
    grid on
    SNR
end

%% INL DNL
LSB = AIN/2^8
if(InputSig==0)
    Q = Q8bit+Q6bit/K
    Outhist14bit = hist(Q,2^14);
    Idealhist = 32;

    dnl = 1-Outhist14bit/mean(Outhist14bit);
    dnl = dnl(2:16383);
    dnl14bit = dnl-mean(dnl);

    inl14bit = cumsum(dnl14bit);

    Outhist8bit = hist(Q8bit,2^8);
    Idealhist = 32*64;

    dnl = 1-Outhist8bit/mean(Outhist8bit);
    dnl = dnl(2:255);
    dnl = dnl-mean(dnl);

    inl = cumsum(dnl);

    figure('name','DNL','numbertitle','off')

    subplot(2,1,1)
    plot(dnl)
    title('DNL TFC, N = 8');
    xlabel('Code');
    ylabel('LSB');

    subplot(2,1,2)
    plot(dnl14bit)
    title('DNL TFC, N = 14');
    xlabel('Code');
    ylabel('LSB');


    figure('name','INL','numbertitle','off')

    subplot(2,1,1)
    plot(inl);
    title('INL TFC, N = 8');
    xlabel('Code');
    ylabel('LSB');

    subplot(2,1,2)
    plot(inl14bit);
    title('INL TFC, N = 14');
    xlabel('Code');
    ylabel('LSB');
end
```

CURRICULUM VITAE

Name Surname: Mehmet Arda AKKAYA

Place and Date of Birth: Istanbul 18.10.1989

Adress: Istanbul-TURKEY

E-Mail: ardaakkaya@gmail.com

B.Sc.: Istanbul Technical University

List of Publications and Patents:

▪ Ozkaya, I. and Gurleyuk, C. and Ergul, A. and **Akkaya**, A. and Aksin, D.Y., 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,*