İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

WIDE INPUT SIGNAL RANGE 14 BITS 1MSPS SAR ADC in 0.35µm HIGH VOLTAGE CMOS PROCESS

M.Sc. Thesis by
İlter ÖZKAYA

Department : **Electronics and Communication Engineering**
Programme : **Electronics Engineering**

September 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

WIDE INPUT SIGNAL RANGE 14 BITS 1MSPS SAR ADC in 0.35μm HIGH VOLTAGE CMOS PROCESS

M.Sc. Thesis by
İlter ÖZKAYA
504071213

Date of submission :	**13 September 2010**
Date of defence examination:	**20 September 2010**

Supervisor (Chairman) :	**Assis. Prof. Dr Devrim Yılmaz AKSIN (ITU)**
Members of the Examining Committee :	**Prof. Dr. Uğur ÇİLİNGİROĞLU (Yeditepe U.)** **Assis. Prof. Dr. Türker KÜYEL (ITU)**

September 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

0.35 µm YÜKSEK GERİLİM CMOS PROSESİNDE GİRİŞ İŞARET ARALIĞI GENİŞLETİLMİŞ 14 BIT 1MSPS SAR ADC

YÜKSEK LİSANS TEZİ
İlter ÖZKAYA
504071213

Tezin Enstitüye Verildiği Tarih : 13 Eylül 2010
Tezin Savunulduğu Tarih : 20 Eylül 2010

Tez Danışmanı : Yrd. Doç. Dr. Devrim Yılmaz AKSIN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Uğur ÇİLİNGİROĞLU (Yeditepe Ü.)
Yrd. Doç. Dr. Türker KÜYEL (İTÜ)

Eylül 2010

FOREWORD

I would like to express my deep appreciation and thanks for my advisor Assist. Prof. Dr. Devrim Yılmaz AKSIN. He not only helped me choosing the thesis subject but also guided me throughout the entire process of developing and realizing the project that constitutes my thesis.

I would like to thank my co-workers from Mikroelektronik Ar-Ge Ltd. Özgür ATEŞ, Ali E. KILIÇ, Emre APAYDIN and my director Barbaros ŞEKERKIRAN for their discussions and support. I would like to thank another colleague of mine Ercan ALTUNTAŞ separately for his invaluable help in digital part of the project.

I also would like to thank to Uğur UYANIK and Başak BAŞYURT for their help in realizing the project.

Finally, I would like to thank my family and my lovely wife for their unconditional support.

September 2010

İlter ÖZKAYA
Electronics Engineering

TABLE OF CONTENTS

Page

TABLE OF CONTENTS .... **vii**
ABBREVIATIONS .... **ix**
LIST OF TABLES .... **xi**
LIST OF FIGURES .... **xiii**
SUMMARY .... **xv**
ÖZET .... **xvii**
1. INTRODUCTION **1**
  1.1 Organization of Thesis 4
2. LITERATURE AND HISTORICAL REVIEW **5**
  2.1 SARADC 5
  2.2 Steps to Sampling Signals above the Supply Voltage 7
    2.2.1 Transmission gate 8
    2.2.2 High voltage controlled switch 9
    2.2.3 Bootstrapped switch 11
    2.2.4 High voltage sampling bootstrapped switch for input signals beyond supply voltage 12
  2.3 Similar Products Comparison 13
  2.4 Specifications of the Proposed SARADC 15
  2.5 Applications 16
    2.5.1 Motor control 16
    2.5.2 Power and voltage monitoring 16
    2.5.3 Pressure measurement 17
    2.5.4 Automotive industry 17
    2.5.5 Telecom industry 17
    2.5.6 Hall sensors 17
3. DESIGNED SUBBLOCKS **19**
  3.1 High Voltage Sampling Bootstrapped Switch 19
    3.1.1 Simulation results 23
  3.2 Fully Differential Operational Amplifier 26
    3.2.1 Comparator feature 27
    3.2.2 Adaptive compensation 28
    3.2.3 Simulation results 31
  3.3 Capacitive Array 34
  3.4 Digital Part 38
  3.5 Bandgap Reference and Power on Reset (POR) 39
4. ALGORITHM **41**
  4.1 Sample State 45
  4.2 SAR State 45
  4.3 Amplification State 46
  4.4 Resample State 47
  4.5 Full Conversion Cycle 48
5. OBTAINING TRANSFER FUNCTION OF SARADC **49**

5.1 Sampling Phase 50
5.2 1st Conversion Phase 51
5.3 Amplification Phase 52
5.4 Resampling Phase 54
5.5 2nd Conversion Phase 55
6. EFFECTS OF NONIDEALITIES 61
6.1 Comparator and Differential Amplifier Offset Voltage 61
6.2 Mismatch at the Capacity Array 62
6.3 Noise 67
6.4 Amplification Factor Error 70
7. MEASUREMENTS 73
7.1 SCATTENUATOR 73
7.2 SARADC Simulation Results 79
7.3 SARADC Tests 80
7.3.1 Spurious free dynamic range and total harmonic distortion (SFDR and THD) 83
7.3.2 Differential and integral non-linearity (DNL and INL) 84
7.3.3 Signal to noise ratio (SNR) 84
7.3.4 Matching of the capacitive DAC 84
7.3.5 Power consumption 85
7.3.6 Bandgap reference voltage and reference current 85
7.3.7 Power on reset operation 86
7.3.8 Integrity of the reference signals inside the chip 86
7.3.9 Stand alone high voltage switch 86
7.3.10 Observation of state control signals 86
8. CONCLUSIONS 89
REFERENCES 93
APPENDICES 95
CURRICULUM VITAE 111

ABBREVIATIONS

ADC **:** Analog to Digital Converter
DAC **:** Digital to Analog Converter
SARADC **:** Successive Approximation Register Analog to Digital Converter
HVSBS **:** High Voltage Sampling Bootstrapped Switch
MSB **:** Most Significant Bit
LSB **:** Least significant Bit
SCA **:** Switched Capacitor Attenuator
Vpp **:** Volts peek-to-peek
DC **:** Direct Current
AC **:** Alternating Current
MSps **:** Mega samples per second
OPAMP **:** Operational Amplifier
POR **:** Power on Reset

LIST OF TABLES

Page

Table 1: High voltage SARADC products comparison. 14
Table 2: Standard input range SARADC products comparison................................ 14
Table 3: Ideal node voltages of each node of the HVSBS. 22
Table 4: Device sizes.. 23
Table 5: Pin explanation of the test chip. 81
Table 6: I2C controlled test signals. 83
Table 7: I2C_TST control configuration.. 87

LIST OF FIGURES

Page

Figure 1.1: Analog to digital converter types and specifications............................... 1
Figure 1.2: Block diagram of SAR ADC.. 2
Figure 1.3: Binary search algorithm. 2
Figure 1.4: Illustration of AC power measurement by sampling............................... 3
Figure 2.1: 1954 "DATRAC" 11-Bit, 50 KSps SAR ADC by B. Gordon at EPSCO.6
Figure 2.2: ADC-12U 12-Bit, 10-µs SAR ADC from Analog Devices, 1969. 6
Figure 2.3: Transmission gate and its equivalent model.. 8
Figure 2.4: Transmission gate transconductance for different supply voltages.......... 9
Figure 2.5: High voltage generating circuit and switch... 10
Figure 2.6: Transient signal flow for high voltage controlled switch...................... 10
Figure 2.7: Bootstrapped switch. 11
Figure 2.8: Switch for sampling input voltages above supply voltage. 13
Figure 3.1: The transistor level schematic of the high voltage sampling switch..... 20
Figure 3.2: HVSS: OFF-ON transition. 24
Figure 3.3: HVSS complete cycle sampling. 24
Figure 3.4: V_{GS} voltage of M13 and M14. 25
Figure 3.5: DC voltage of pass transistors M13 and M14 for vdd=2.2V. 25
Figure 3.6: Fully differential operational amplifier schematics............................... 26
Figure 3.7: Comparator. 28
Figure 3.8: Amplification and resample configurations. 29
Figure 3.9: Adaptive compensation technique... 30
Figure 3.10: Open loop AC response for both unity gain and amplification modes. 32
Figure 3.11: Step response of adaptive compensation and standard compensation. 33
Figure 3.12: Transient simulation results: Opamp in comparator mode.................. 33
Figure 3.13: Classical binary weighted capacitive DAC. 34
Figure 3.14: The use of an attenuator in the middle of the array............................. 34
Figure 3.15: Layout of the capacity cell. 37
Figure 3.16: Top view layout of the capacitive array with bottom plate driving circuitry around. 38
Figure 4.1: State flow diagram... 41
Figure 4.2: SAR algorithm.. 43
Figure 4.3: Top level schematic of the designed SARADC. 44
Figure 4.4: Sample state... 45
Figure 4.5: SAR state... 46
Figure 4.6: Amplification state. 46
Figure 4.7: Resample state... 47
Figure 4.8: Full cycle conversion... 48
Figure 5.1: SARADC... 49

Figure 6.1: Ideal and non-ideal transfer functions. 64
Figure 6.2: Monte Carlo analyses for capacitor array mismatch. 66
Figure 6.3: DNL plots for 10 runs of SARADC matlab model. 66
Figure 6.4: INL Plots for 10 runs of SARADC matlab model................................. 67
Figure 6.5: Amplification factor error affect on transfer function. 71
Figure 7.1: Switched capacitor attenuator schematic... 73
Figure 7.2: FFT results of SC attenuator simulation.. 76
Figure 7.3: Die micrograph of SCA. 77
Figure 7.4: Measurement of high-voltage sampling switch (S3) in track and hold configuration. Input 20 Vpp + 15 V_{DC}, clock 1 MHz. 78
Figure 7.5: Measurement result of SC attenuator showing V_{INP} and V_{OP} terminal voltages, test condition 20Vpp differential input on $10V_{DC.}$.................... 78
Figure 7.6: FFT analysis of SARADC top level simulation Fin = 15.6KHz. 79
Figure 7.7: FFT analysis of SARADC top level simulation Fin = 475 KHz. 79
Figure 7.8: Top level symbol for test chip. 80
Figure 7.9: Top level test control signal.. 82
Figure A.1: High voltage switch schematics.. 95
Figure A.2: Layout of high voltage sampling switch... 95
Figure A.3: Schematics of fully differential operational amplifier........................... 96
Figure A.4: Layout of the differential amplifier. 97
Figure A.5: Schematics of the bandgap reference. 97
Figure A.6: Layout of the bandgap reference. 98
Figure A.7: Schematics of power on reset. 98
Figure A.8: Layout of the power on reset. 99
Figure A.9: Layout of the top level chip. 99
Figure A.10: Bonding diagram. 100

WIDE INPUT SIGNAL RANGE 14 BITS 1MSPS SAR ADC in 0.35μm HIGH VOLTAGE CMOS PROCESS

SUMMARY

In the last two decades, integrated circuits have been increasingly used in process control systems. Moreover, digital control loops have become more and more popular with the rapid increase in computation speed and integration density.

For instance, in the past, various techniques were used for motor control such as using resistors as sensors, isolating amplifiers, and magnetic sensors. However, each technique had drawbacks like increased power consumption, complicated circuitry, and especially performance degradation. Thus, digital control has become the best solution for high performance with low cost. As a result, analog to digital converters are required for converting the analog voltage and current signals into the digital data the controller can handle.

An important difficulty in implementing such systems is the voltage difference between the analog signal to be sampled and supply voltage of the data converter. Obviously, in most motor control applications high voltages are required for high power. On the other hand, the performance of the integrated circuits has increased due to the reduction in transistor size, consequently reducing breakdown voltages.

To date, the most common way of solving the mentioned problem was to attenuate the analog signal to fit into the supply voltage range, at the expense of reduced signal integrity. Some companies developed industrial processes and used extra supplies in their products to sample high voltages, which complicates the system with the need for extra supply voltages and increases power consumption.

In this study, a novel high voltage sampling technique is proposed for sampling high voltages with standard supply voltages for integrated circuits. The sampling technique is used to implement a wide input signal range successive approximation register analog to digital converter. Besides the one implemented, the proposed sampling technique can be used to realize other ADCs with different specifications.

The proposed SARADC is implemented with AMS's 0.35 μm CMOS process with drain extended NMOS (DNMOS) transistors, whose drain terminal breakdown voltage is 50V. The system consumes 10 mW power under typical conditions, and the total area of the chip is 1.45mm x 1.85mm.

Some of the possible application areas of the proposed SARADC may be the automotive and telecom industries where 12V and 48V are standards, respectively. Furthermore, pressure sensors used in the industry have an analog output range of 0-10V.

0.35 µm YÜKSEK GERİLİM CMOS PROSESİNDE GİRİŞ İŞARET ARALIĞI GENİŞLETİLMİŞ 14 BIT 1MSPS SAR ADC

ÖZET

Son yirmi yılda süreç control sistemlerinde, tümdevre kullanımı giderek artmıştır. Buna ek olarak, hesaplamadaki hız artışı ve yüksek entegrasyonla beraber sayısal control döngüleri daha da yaygınlaşmıştır. Örneğin, eskiden, motor kontrolü için dirençlerin sensor olarak kullanımı, yalıtıcı kuvvetlendirici ve manyetik sensör kullanımı gibi teknikler mevcuttu . Ancak, her tekniğin güç tüketimi artışı, devre karmaşıklığı ve özellikle başarım düşmesi gibi dezavantajları vardı. Bundan dolayı sayısal kontrol, düşük maliyetli yüksek perfomans için en iyi çözüm oldu. Sonuç olarak analog gerilim ve akım işaretlerini işlemcinin kullanabileceği sayısal veriye dönüştürmek için analog sayısal çeviricilere gereksinim duyulmaktadır.

Bu tür sistemlerin kurulmasına zorluk teşkil eden en büyük unsur örneklenecek analog sinyal ile kaynak arasındaki gerilim farkıdır. Bilindiği gibi çoğu motor kontrol uygulamalarında yüksek güç için yüksek gerilim kullanılmaktadır. Diğer taraftan entrgre devrelerdeki performans artışının temel nedeni tranzistor boyutlarının küçülmesidir. Bunun sonucu olarak da kırılma gerilimleri düşmüştür.

Günümüze kadar bahsedilen problemin en yaygın çözümü işaret bozulması pahasına giriş işaretinin kaynak gerilimi düzeyine inmesini sağlayacak şekilde zayıflatılmasıydı. Bazı şirketler endüstriyel prosesler geliştirdi ve ürünlerinde yüksek gerilimi doğrudan örneklemek için fazladan gerilim kaynağı kullanmaktadırlar. Bu da tasarlanan sistemi karmaşık hale getirmekte ve güç tüketimi arttırmaktadır.

Bu çalışmada standart kaynak gerilimiyle yüksek gerilim örneklemesi yapmayı sağlayacak bir yöntem önerilmiştir. Önerilen yöntem geniş giriş aralıklı ardışıl yaklaşımlı bir analog sayısal çevirici (SARADC) gerçekleştirmek için kullanılmıştır. Önerilen yöntem bu çalışmada tasarlanandan farklı özelliklerde analog sayısal çeviricilerin yapımında da kullanılabilir.

Önerilen SARADC genişletilmiş savaklı NMOS (DNMOS) tranzistor sunan AMS 0.35 µm CMOS teknolojisi kullanılarak üretilmiştir. Bahsedilen DNMOS tranzistorlar için savak kırılma gerilimi 50 V’tur. Sistem tipik çalışma koşullarında 10 mW güç harcamaktadır. Tasarlanan çipin yüzey alanı 1.45mm x 1.85mm’dir.

Önerilen SARADC için bazı kullanım alanları standart gerilimlerin 12V ve 48V olduğu otomotiv ve telekom sanayileridir. Bundan başka endüstride yaygın olarak kullanılan basınç sensörlerinin çalışma aralığı da 0-10V olduğundan bu sensörlerle birlikte de kullanılabilir.

1. INTRODUCTION

Analog to Digital Converters (ADCs) are widely used in a variety of applications in data acquisition, communications, instrumentation, and interfacing for signal processing. Most of the time, the application area of the ADCs determine the specifications such as resolution, conversion speed, and power consumption. In order to cover the broad range of specifications different architectures are presented.

The basic ADC architectures are Sigma-Delta (Σ-Δ) ADC, Successive Approximation Register (SAR) ADC, Pipeline ADC, and Flash ADC. The sampling rate and resolution specifications of each ADC architecture are given in Figure 1.1 together with some application areas.

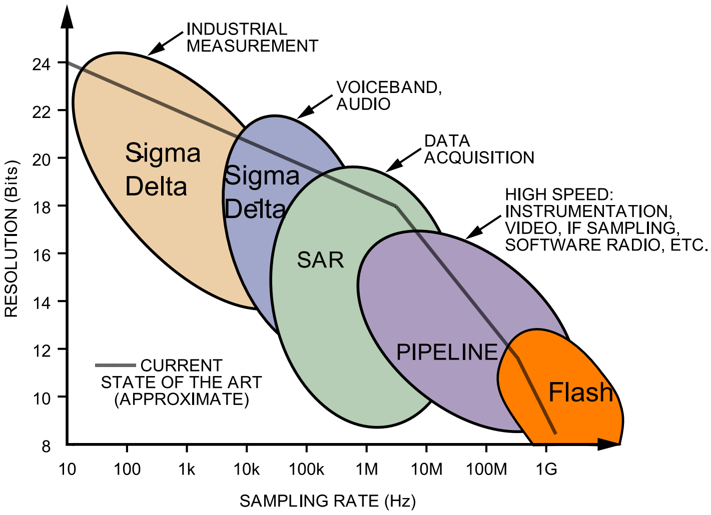


Figure 1.1: Analog to digital converter types and specifications.

Amongst other types, Successive Approximation Register ADCs are frequently the architecture of choice for medium to high-resolution (8 to 18 bits) applications with sampling rates up to several megasamples-per-second (5 MSps) with low power consumption.

As the name implies, the SARADC implements a binary search algorithm. Therefore, sample rate is a fraction of the frequency of the clock signal applied to it.

The block diagram of a basic SAR ADC is given in Figure 1.2.

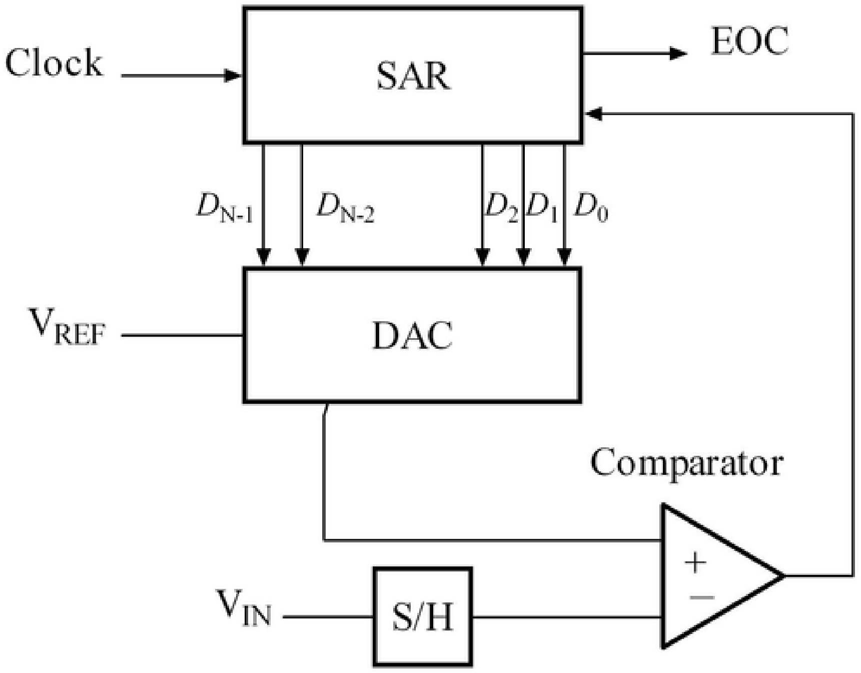


Figure 1.2: Block diagram of SAR ADC.

In the figure analog input signal V_{IN} is sampled through the sample and hold S/H circuitry. Successive Approximation Register (SAR) is where digital approximation is carried out. A digital to analog converter (DAC) converts the digital signal coming from SAR into an analog signal. Then, comparator compares this signal with sampled input signal and SAR generates another bit according to the comparator output. This way, DAC output gradually approaches the sampled input voltage. This binary search algorithm is illustrated in Figure 1.3.

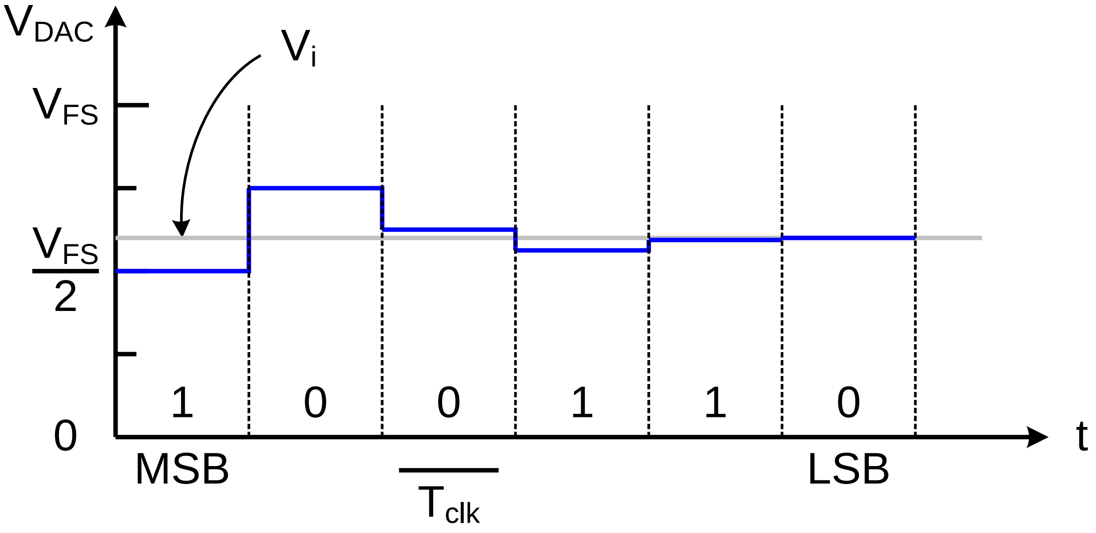


Figure 1.3: Binary search algorithm.

Maximum allowed analog input range is another important specification for applications such as industrial process control and power-line monitoring systems, where employed voltages are much greater than maximum ratings for today's standard integrated circuit technologies. Figure 1.4 demonstrates the fundamentals of the AC power measurement by sampling calculations for instantaneous and average power usage.

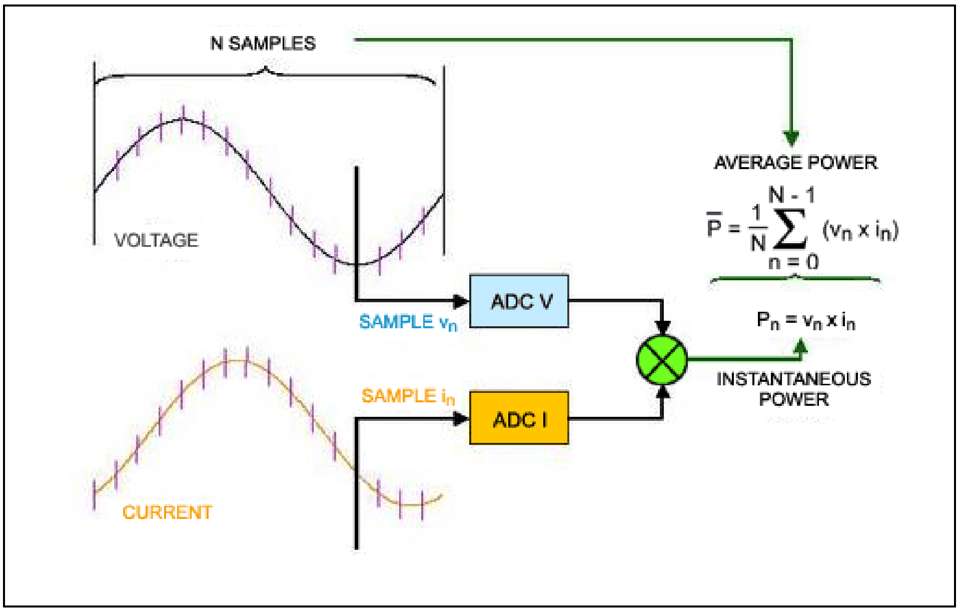


Figure 1.4: Illustration of AC power measurement by sampling.

Many attempts were made to achieve high input voltage sampling, most of which applied the technique of either increasing supply voltage (employing a high voltage process) or dividing the input signal outside the chip to fit in the standard ADC input range. The former technique increases power consumption while the latter one requires extra off-chip components and degrades input signal accuracy. In order not to suffer from both of the drawbacks, the high voltage signal must be sampled using standard supply voltage. Hitherto, to the author's knowledge, input sampling range of twice as high as supply voltage is achieved.

In this study a wide input range (32 Vpp) SARADC with a single supply voltage of 3.3 V is presented. The input signal is directly handled within the die without any external signal division operation. The high voltage sampling is achieved through a novel high voltage sampling switch. The switch utilizes a bootstrapping technique to sample signals much higher than the supply voltage. SARADC is implemented with standard cmos 0.35 μm technology with high voltage option.

1.1 Organization of Thesis

The organization of the thesis can be summarized as follows:

In chapter 2, evolution of SARADCs and high voltage sampling techniques are presented. Moreover, similar products and application areas are presented.

In chapter 3, the design and operation details of the subblocks used to realize the SARADC are presented. The simulation results of the subblocks are provided to show performance and operation characteristics.

In chapter 4, the algorithm used for analog to digital conversion is explained. The configuration of the system at each state is illustrated with the figures.

In chapter 5, the mathematical verification of the algorithm is provided and transfer function of the proposed SARADC is found.

In chapter 6, the effects of non-idealities such as mismatch, amplifier offset voltage and noise are introduced, and their possible effects on the performance are discussed.

In chapter 7, the measurement results of the previously realized switched capacitor attenuator are given. Moreover, the tests procedure of the SARADC is presented and the tests to be conducted are explained.

Finally, chapter 8 summarizes the study and discusses possible future applications of the proposed sampling technique.

2. LITERATURE AND HISTORICAL REVIEW

In this section, firstly, the evolution of the SARADCs used today will be presented in a chronological fashion. After the steps of this evolution are presented, some of the wide input range SARADC products will be compared. Then, the specifications of the proposed SARADC will be given. Finally, some application areas will be mentioned.

2.1 SARADC

The basic algorithm used in the successive approximation (initially called feedback subtraction) ADC conversion process can be traced back to the 1500s relating to the solution of a certain mathematical puzzle regarding the determination of an unknown weight by a minimal sequence of weighing operations [1]. In this problem, as stated, the objective is to determine the least number of weights, which would serve to weigh an integral number of pounds from 1 lb to 40 lb using a balance scale. One solution put forth by the mathematician Tartaglia in 1556, was to use the series of weights 1 lb, 2 lb, 4 lb, 8 lb, 16 lb, and 32 lb. The proposed weighing algorithm is the same as used in modern successive approximation ADCs.

Even though the algorithm has been used widely for centuries, the first mention of successive approximation ADC architecture was by J. C. Schelleng of Bell Telephone Laboratories in a patent filled in 1946 [2]. This was as interesting description of a rather cumbersome successive approximation ADC based on vacuum tube technology. Other more elegant implementations were presented from that day on, mostly by the Bell Telephone Laboratories. However, the great leap was the work of Bernard M. Gordon at EPSCO (now Analogic, Incorporated). Gordon's 1955 patent application [3] describes an all-vacuum tube 11-bit, 50-KSps successive approximation ADC, representing the first commercial offering of a complete converter (Figure 2.1). The DATRAC was offered in a 19" × 26" × 15" housing, dissipated several hundred watts, and sold for approximately $8500.00.

After the fast development of integrated circuitry, new companies were founded offering new products of better usage. ADC-12U of Analog Devices became one of the most common products after its release in 1969, with 12 –bit resolution and 100 KSps (**Figure 2.2**).

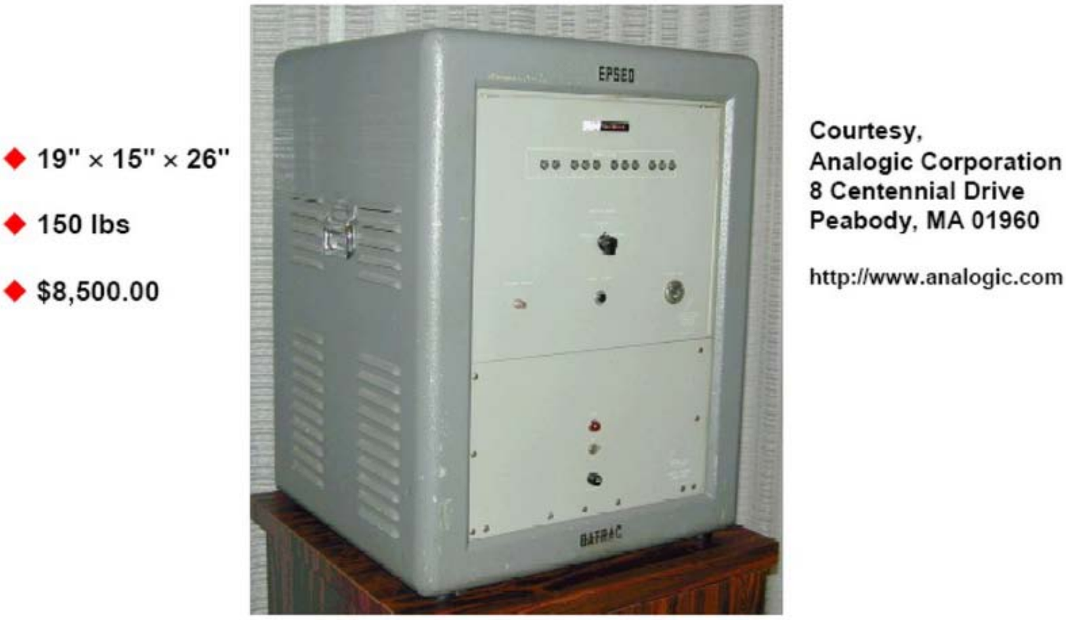


Figure 2.1: 1954 "DATRAC" 11-Bit, 50 KSps SAR ADC by B. Gordon at EPSCO.

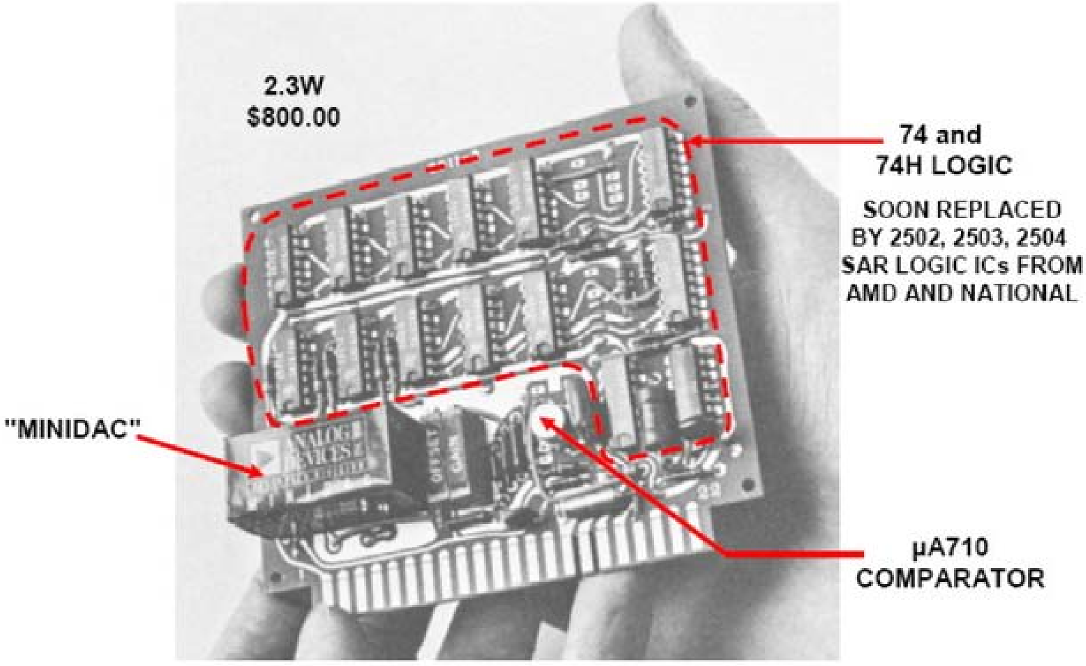


Figure 2.2: ADC-12U 12-Bit, 10-µs SAR ADC from Analog Devices, 1969.

Then came along monolithic data converters. One of the most significant SAR ADC ever introduced was the 12-bit, 35 μs AD574 in 1978. The product used DACs with laser-trimmed thin-film resistors to achieve the desired accuracy and linearity. However, the process of depositing and trimming thin-film resistors adds cost, and the thin-film resistor values may be affected after the device is subjected to the mechanical stresses of packaging.

For these reasons, switched-capacitor (or charge-redistribution) DACs have become popular in newer CMOS-based SAR ADCs. The basic advantage in using capacitors instead of resistors is that the accuracy and linearity is primarily determined by the high-accuracy photolithography. In addition, autocalibration techniques can be made use of by adding small capacitors in parallel with the main capacitors. Another advantage of using capacitors is their good temperature tracking feature. An example of modern charge-redistribution SARADC is Analog Devices AD7641; 18-bit, 2 MSps, fully differential ADC operating at a single supply of 2.5 V.

Over the last decades, with the technology scaling, the supply voltages have been decreasing, limiting the allowed input range to less than or equal to the supply voltage. In industrial applications, the limited input voltage range obsoletes standard ADCs, unless the input signal is divided outside the chip. However, external circuitry is undesired in most of the applications.

Another solution to this problem is implementing the SARADC with special industrial processes that can withstand high voltages. Analog Devices use its own industrial process iCMOS in its high voltage SARADCs. Yet, these SARADCs require high voltage supply (±15 V) apart from the main supply (5 V). This requirement not only complicates the supply distribution on the board, but also increases the power consumption.

2.2 Steps to Sampling Signals above the Supply Voltage

In order not to suffer from both external circuitry and multiple supply requirements, high voltages need to be sampled using sampling switches operating with standard supply voltages.

The techniques developed to achieve such a sampling are described in this section.

2.2.1 Transmission gate

Transmission gates are the most commonly used configuration of the MOS transistors as sampling switches. They simply consist of an NMOS and a PMOS transistor with the control signal and its inverted signal is applied to the gates of the transistors, respectively. The schematic of the transmission gate is given in Figure 2.3.

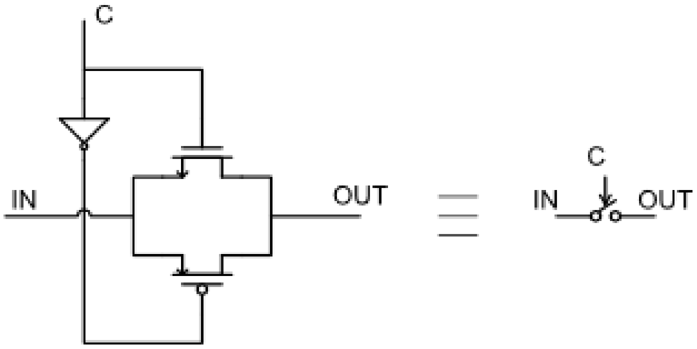


Figure 2.3: Transmission gate and its equivalent model.

NMOS of the T-gate is on conduction when the input signal is low and PMOS of the T-gate is on conduction when the input signal is high. In addition, they are both on conduction for input signal in the middle of the supply voltage.

Rapid decrease in transistor gate size led to decrease in supply voltages used. As the supply voltage decreases, the possible overdrive voltage of the MOS transistors decreases as well. At some point, widely used transmission gates become obsolete. In Figure 2.4 this phenomenon is shown.

As can be seen from the figure, as the supply voltage (denoted as “Vdd”) decreases the transconductance of the T-gate decreases for input signal half the supply voltage. Even at some point T-gate is no more on conduction.

New circuits are introduced to overcome this drawback.

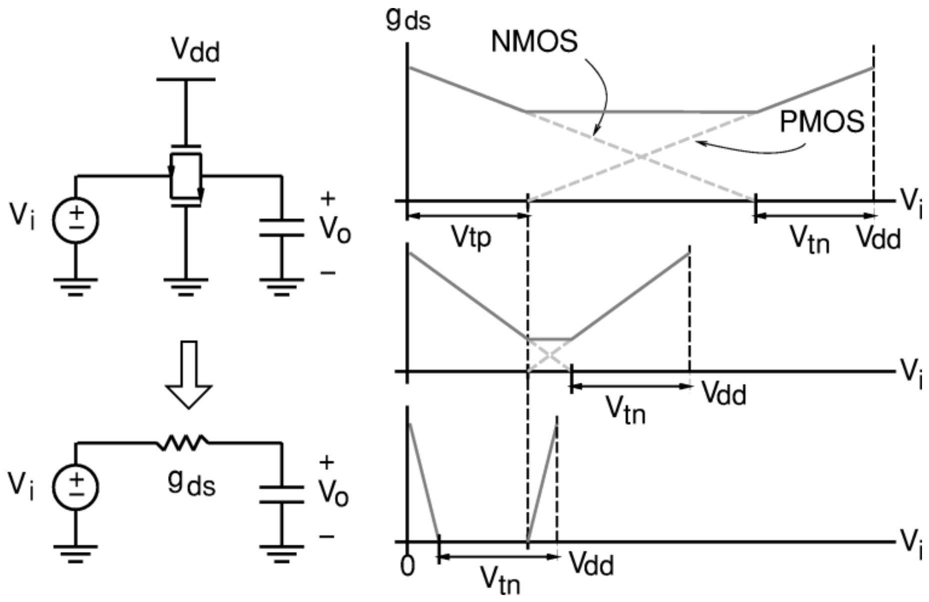


Figure 2.4: Transmission gate transconductance for different supply voltages.

2.2.2 High voltage controlled switch

Thomas B. Cho and Paul R. Gray introduced a high voltage generating circuit to control the operation mode of the switch [4]. The high voltage generating circuit and switch is given in Figure 2.5.

The operation of the circuit is explained through the transient signal flow of control signal CTR and nodes n1, n2 and n3 in Figure 2.6. Initially the capacitors are assumed to be charged to 0 V, and signal CTR_B is the complement of the control signal CTR.

Although this circuit is well designed to overcome the drawback of the previously mentioned T-gate, it suffers from reliability issues. The gate oxide of the transistor MNSW is exposed to high voltage stress when the input voltage is close to 0. In addition, when control signal CTR is 0, the bottom plate of the capacitor C_2 is at Vdd, and top plate is close to 2Vdd. Therefore, independent of input signal voltage, the transistor MP1 suffers from gate oxide voltage stress when control signal is 0. In order to save these two transistors from breaking down, thick oxide transistors are chosen.

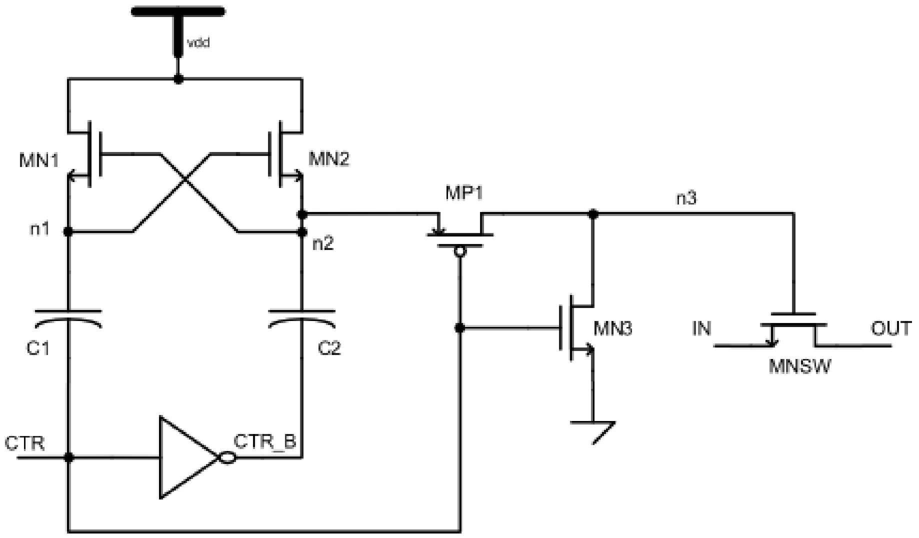


Figure 2.5: High voltage generating circuit and switch.

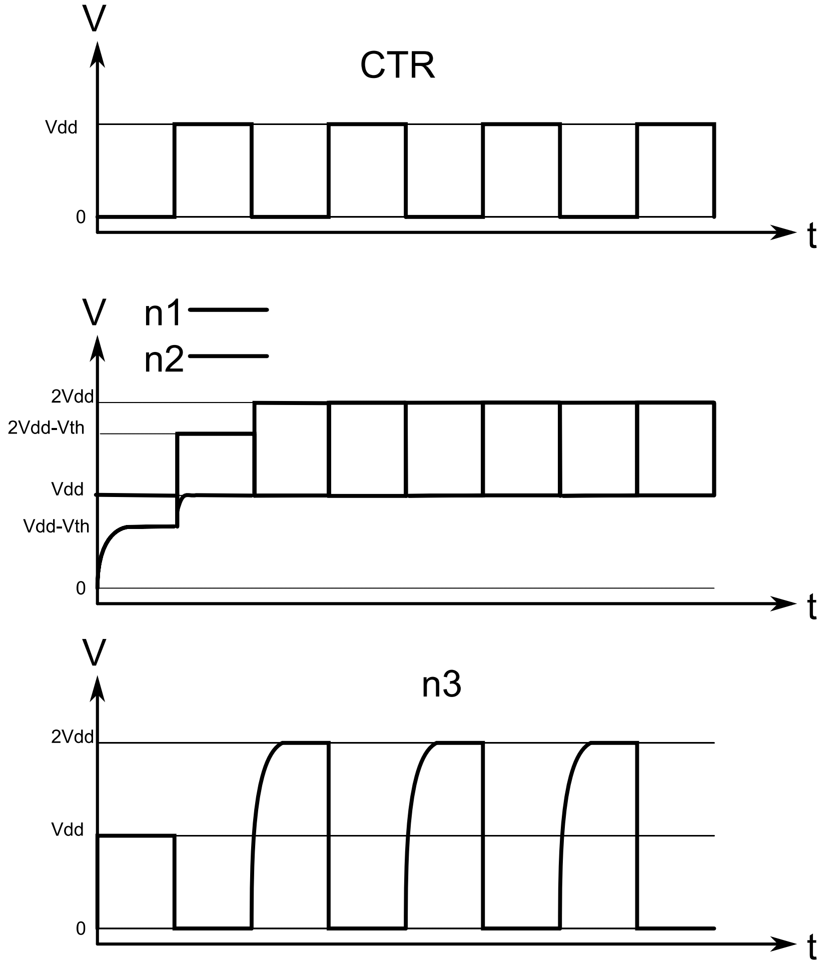


Figure 2.6: Transient signal flow for high voltage controlled switch.

2.2.3 Bootstrapped switch

Another attempt to solve the sampling problem with low supply voltage was presented by Andrew M. Abo and Paul R. Gray [5]. The presented switch is called a “bootstrapped switch”. Bootstrapping refers to a group of metaphors that share a common meaning: a self-sustaining process that proceeds without external help. The term is often attributed to Rudolf Erich Raspe's story “The Surprising Adventures of Baron Munchausen”, where the main character pulls himself out of a swamp, though it's disputed whether it was done by his hair or by his bootstraps [6].

The clever idea behind the switch is to set the overdrive voltage of an NMOS transistor to supply voltage by connecting a pre-charged capacitor between the input and gate of the switching transistor. The circuit implemented to realize this behavior is given in Figure 2.7.

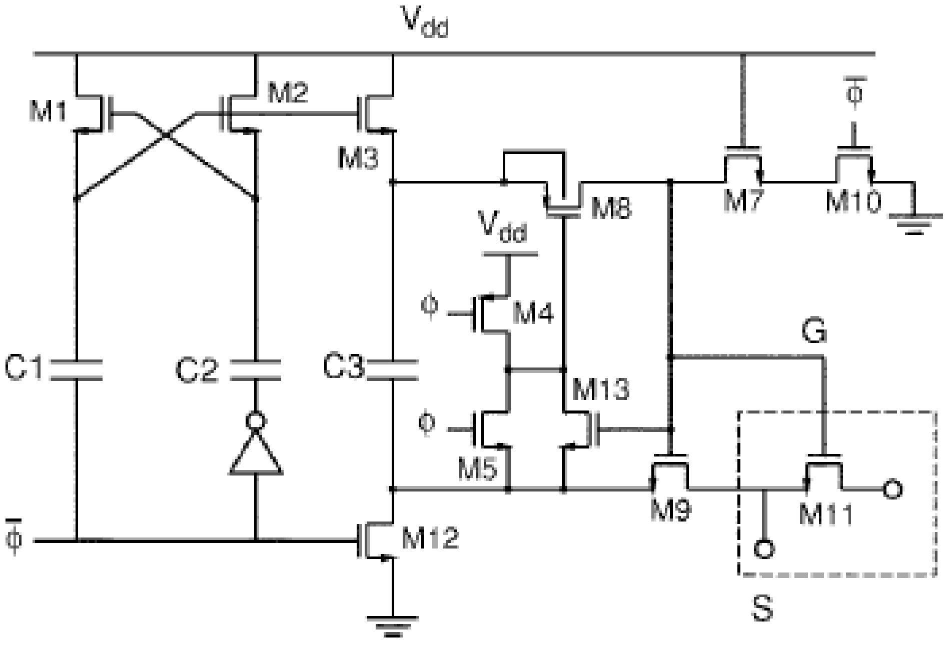


Figure 2.7: Bootstrapped switch.

The capacitor “C3” is charged to positive supply voltage “Vdd” when phase Φ is 0. As the phase Φ goes to 1, transistors M10 and M12 turn off, and M5 turns on, connecting the bottom plate of C3 to the gate of the PMOS transistor M8. As a result, M8 is on to connect the top plate of C3 to the node G, turning on transistors M11, M9, and M13 on. Therefore, C3 is connected between the source and gate of the switching transistor M11 through transistors M9 and M8. As the input signal S approaches to the positive supply voltage Vdd, the transistor M5 goes to cut-off region. That is why M13 is included within the circuit. M5 kick starts the bootstrapping process, and then M13 takes control.

The presented switch has superior performance over the previously mentioned switches; since the overdrive voltage of the switching transistor is equal to “Vdd” independent of the input signal S (parasitic effects will be investigated in detail later). Thus, second and higher order harmonics are suppressed quite well. Moreover, apart from the previously explained “High Voltage Controlled Switch”, bootstrapped switch has no reliability problems, and can be implemented with standard transistors only. That is why it became widely used in low voltage sampling applications. However, as the input voltage exceeds the supply voltage, the bulk-drain diode of the transistor M4 starts to become forward biased and the input voltage is limited.

2.2.4 High voltage sampling bootstrapped switch for input signals beyond supply voltage

Another bootstrapped switch to sample signals as high as 2 times supply voltage is presented in [7]. The circuit used in this implementation is given in Figure 2.8.

Although main idea the same as the bootstrapped switch presented by ABO, this circuit is designed to sample input voltages beyond supply voltage. The sampling starts at transition of signal Φ from 0 to 1. At that instant, MN8 turns on to connect the bottom plate of the capacitor C3 to the input node IN, so that the voltage at node N3 is (Vdd + Vin). If the input voltage is high enough, the transistor MP4 is directly turned on, otherwise, MP4 is turned on by the transistor MN13. Thus, MN1 switches on since its gate voltage is Vdd higher than its source voltage.

This switch is implemented using transistors that can withstand gate oxide voltages of 5.5 V to assure none of the transistors undergo to gate oxide voltage stress. The applied supply voltage is 2.75V.

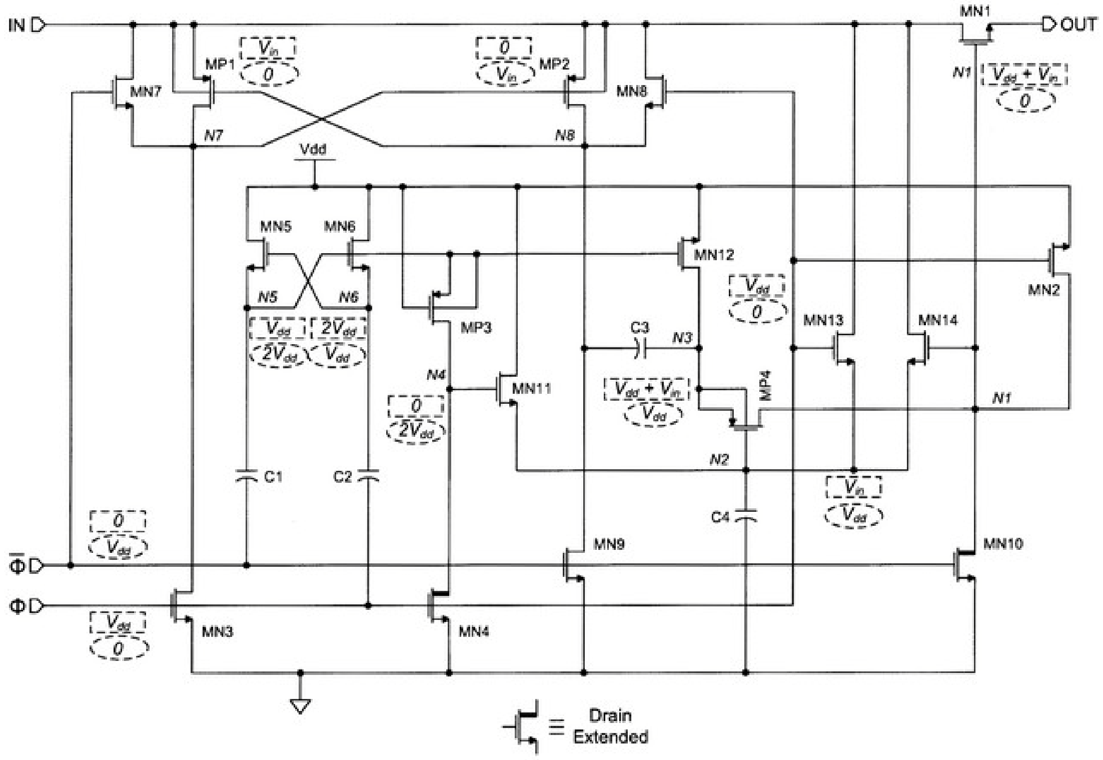


Figure 2.8: Switch for sampling input voltages above supply voltage.

2.3 Similar Products Comparison

SARADCs with wide input range of some of the technology leading companies, such as Analog Devices, Texas Instruments, Maxim, and Linear Tech will be introduced in this section.

Of these companies, Analog Devices uses its own *industrial CMOS* (*iCMOS*) process with extra supply voltage of up to ±15V, while the others use internal resistor dividers to fit into the standard range.

Table 1 summarizes the specifications for some of the commercial products of different companies.

The products investigated within the table have a varity of resolution, sampling rate and power consumption. A tight correlation between sampling rate and power consumption is obvious: The higher the sampling rate is, the higher the power consumption is.

In order to give a general idea about standard input range products with similar specifications another comparison is provided in

Table **2**.

Table 1: High voltage SARADC products comparison.

Company	Model No	Input Range[1](V)	Resolution (Bit)	Sample Rate (KSps)	Power (mW)	Power Supply (V)	SFDR (dB)
Analog Dev.	AD7612	5, 10, ±5, ±10	16	750	190	5, ±15	103
Analog Dev.	AD7952	5, 10, ±5, ±10	14	1000	235	5, ±15	105
Texas Instr.	ADS8504	±10	12	250	70	5	95
Texas Instr.	ADS8505	±10	16	250	70	5	105
Texas Instr.	ADS8515	±10	16	250	100	5	98
Maxim IC	MAX1132	12, ±12	16	200	45	5	105
Maxim IC	MAX1142	12, ±12	14	200	45	5	102
Maxim IC	MAX1272	5, 10, ±5, ±10	12	87	7.5	5	88
Linear Tech.	LTC1609fa	5, 10, ±5, ±10	16	200	65	5	100

Table 2: Standard input range SARADC products comparison.

Company	Model No	Input Range(V)	Resolution (Bit)	Sample Rate (MSps)	Power (mW)	Power Supply (V)	SFDR (dB)
Analog Dev.	AD7980	2.5-5.5[2]	16	1	7	2.5	105
Analog Dev.	AD7622	2.5	16	2	70	2.5	103
Texas Instr.	ADS8329	5	16	1	15	2.7-5	100
Texas Instr.	ADS8471	5	16	1	220	5	110
Maxim IC	MAX1162	4.096	16	0.2	13.75	5	90
Maxim IC	MAX1062	4.096	14	0.2	13.75	5	92
Linear Tech.	LTC2393	4.096	16	1	140	5	108

---

[1] Input range is given for maximum and minimum values allowed at the input terminals. For example if ±10V is given as input range, then this devices’ input range is 2x20V= 40Vpp.

[2] This devices’ input range is determined by the external reference signal and is between 2.5V and 5V.

2.4 Specifications of the Proposed SARADC

The main goal of this study is to design a single supply high voltage sampling SARADC without input scaling. In order to hold both requirements a high voltage sampling with standard supply voltage technique is proposed and implemented.

The design is realized using 0.35 μm twin-well CMOS process with drain extended MOS capability. The available drain extended MOS devices have a drain breakdown voltage of 50V, while their oxide can withstand to only 3.3V. Thus, the supply voltage is limited by 3.3V from above. However, the circuit is designed so that the supply voltage can be reduced down to 2.2V.

Power consumption is directly connected to the supply voltage used. Since no high voltage supply is needed, the power consumption can be kept at minimum. The other factor that is directly affecting the power consumption is the sampling rate. The relation is obvious from the comparison tables provided; as the sampling rate increases power consumption increases as well. The power consumption goal for the proposed SARADC is 10 mW. Compared to the products given in Table 1, this specification is very competitive. On the other hand, most of the products introduced above generate their own reference voltage and/or clock signal increasing the power consumption. A more fair comparison can be made if internal reference were included in the proposed SARADC. The technology used and design allows the maximum input range to be as high as 50V, nevertheless, the goal for this specification is kept conservative to be 16V (making a 32Vpp differential input range). Yet, the maximum value for this specification may improve after measurements.

Another important specification is sampling rate. Since no resistive division is used to attenuate the input signal, the sampling speed is not reduced by the RC time constant. As a result, a high sampling rate can be achieved with the high voltage sampling technique. The goal for this specification is 1 MSps.

An 8-bit charge redistribution DAC is used to realize the design. A subranging algorithm is used to get 16 bits of data after the DAC is used 2 times in one sampling period. Of these 16 bits, 2 bits will be used for digital error connection and the rest 14 bits is the resolution of the proposed DAC.

2.5 Applications

In most of the industrial applications large amount of energy is required. As a result, high voltage systems are implemented for reasonable current levels. Moreover, most of the time, the systems use analog to digital converters to make use of digital controllers for high performance. The proposed SARADC has a wide range of application areas in those where digital controllers are integrated into high voltage operating systems. In this section, some of the possible application areas will be covered briefly.

2.5.1 Motor control

Accurate torque and speed control, low motor ripple, and reliable performance are some of the basic considerations in designing a motor control system. The driving current and voltage of the motor provides information about motor torque, speed, shaft position and direction [8]. In the past, analog techniques, such as magnetic or resistive sensors, used to be implemented for motor control. However, compared to integrated sampling techniques, their high power consumption and low accuracy they became obsolete. Thus, high voltage sampling ADCs became widely used in sensing current and voltage values of motor, for better control. The proposed ADC satisfies the needs of this application area.

2.5.2 Power and voltage monitoring

Three phase voltage and power monitoring is another possible application area of the proposed SARADC. Accurate tracking of the current and voltages are important to ensure line voltage compliance and protect against excessive line voltages. Moreover, power factor can be tightly controlled and corrected via proper power monitoring.

In some applications, portable devices are needed for power monitoring. In that case the power consumption is of crucial importance since the portable devices use batteries as supplies [8]. Thus, low voltage and low power specifications of the proposed SARADC could make it the product of choice for such applications.

2.5.3 Pressure measurement

The most widely used industrial pressure sensors’ output voltages range is 0-10V [9]. This voltage range is impossible to sample with standard ADCs, of course without input attenuation. The proposed SARADC can handle this voltage range without input attenuation.

2.5.4 Automotive industry

Automobile batteries drive all the devices in a car such as GPS, radio, sound systems, and air conditioning. Since many of these systems may be run when battery is not charging, power monitoring becomes a major concern [10]. The standard batteries of cars generate 12 V of output voltage, which fits well into the proposed SARADC input range. Yet, the voltage range in cars can go up to 60V.

Moreover, hybrid and electric cars are becoming more and more popular with the increasing sensitivity on environmental issues. As a result, the application area of high voltage ADCs in the automotive industry is expected to expand.

2.5.5 Telecom industry

The standard voltage used for telecom applications is around 48V, which is quite higher than the input range of the proposed SARADC. However, with just a few parameter changes in the design, up to 50V input sampling can be reached with the proposed sampling technique.

In the past few years the IEEE802.3af Power over Ethernet (PoE) standard is emerging to generate wider applications [10].

Furthermore, communication over power-line is another area of interest in the last years. This technology has many applications such as generating a home networking (LAN) or internet excess over the 220V power-lines. This is a much higher voltage than the limit voltage that can be achieved by the proposed sampling technique for the same technology. Yet, the sampling technique is universal and can be implemented with any technology.

2.5.6 Hall sensors

Hall Sensor is a transducer which generates an output voltage in response to changes in magnetic fields. They are used for proximity, switching speed detection and current sensing applications[11]. Most of the time, the sensors are designed for wide operating range. For example, US4881 (Melexis) has an operating range of 2.2V to 18V, or 49x5 series of Infineon has an operating range of 4V to 24V. When these sensors are used at high power applications, they are driven closer to their high voltage opeation. Thus, they would fit well into the input range of the proposed SARADC.

3. DESIGNED SUBBLOCKS

In this section, the subblocks designed to realize the proposed SARADC are presented. Design details are explained and simulation results for important performance criteria are provided.

The first sub-block is high voltage sampling switch, which utilizes a bootstrapping technique to high input voltages with standard supply. The second sub-block is a differential amplifier used for amplifying the residue signal in sub-ranging algorithm. The amplifier is compensated adaptively for different load conditions. Another feature of this amplifier is it includes a comparator to operate at successive approximation algorithm.

The third sub-block is the capacitive DAC. This sub-block is especially important since the resolution of capacitive DAC is the resolution of SARADC itself. That is why special attention was paid in the design and layout of this part.

Other subblocks included are power on reset that generates an enable signal when power supply is turned on and bandgap reference circuits that generates current bias for differential amplifier.

3.1 High Voltage Sampling Bootstrapped Switch

A new high-voltage bootstrapped sampling switch with input signal range exceeding 11 times its supply voltage is presented [12]. Proposed switch occupies a silicon area of 250μm by 160μm in 0.35μm twin-well CMOS process with drain extended NMOS (DNMOS) capability. The drain terminal breakdown voltage of the available DNMOS devices, shown with thickened drain terminal within the figure, is 50V while their gate oxide withstands only 3.3V [13].

Implemented switch can reliably track and hold 20 V peek-to-peek (Vpp) signal on 15 V_{DC} at 1MSps with supply voltage of as low as 2.2 V without forward biasing any parasitic diode.

The transistor level of the schematic of the proposed switch is given in Figure 3.1. This figure does not include phase generating circuitry. An exact copy of the schematics and layout of this device is given in APPENDIX A.1 (Figure A.1 and Figure A.2 respectively).

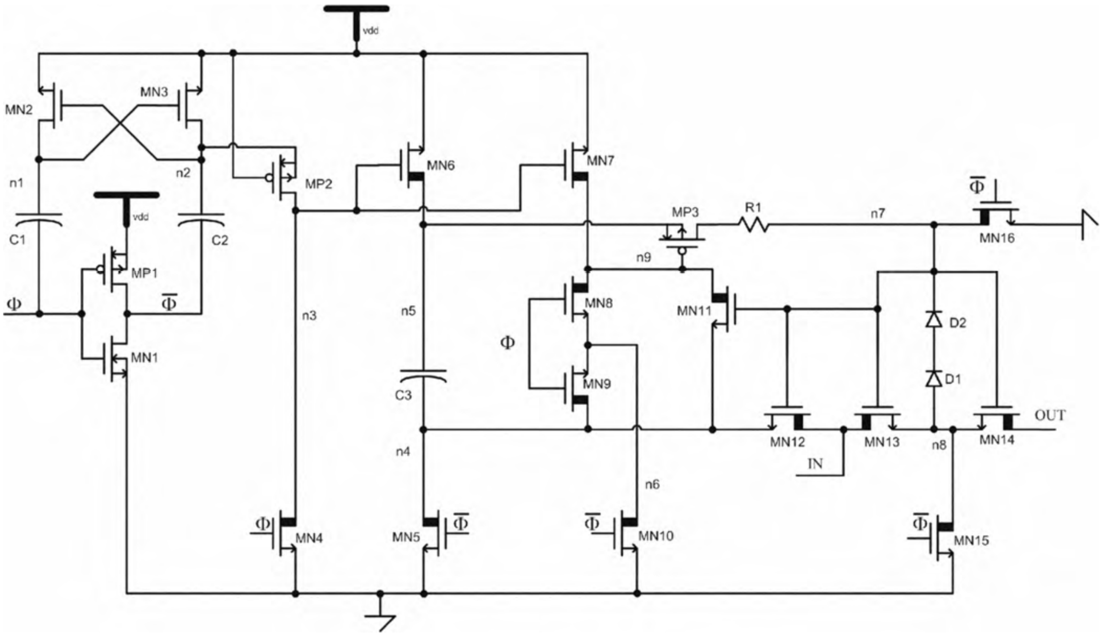


Figure 3.1: The transistor level schematic of the high voltage sampling switch.

When Φ is low, since n3 is equal to 2Vdd, nodes n5 and n9 are forced to Vdd by the transistors MN6 and MN7 to turn of MP3. During this phase all the transistors MN5, MN10, MN15 and MN16 are on to set the nodes n4, n6, n8, and n7 to zero. Since n7 is 0, MN13 and MN14 are off to separate input and output nodes. Also, MN12 is off to separate input signal from the rest of the circuit.

When Φ goes from logic low to logic high, transistors MN8 and MN9 turn on, to discharge n9 to ground. As a result, MP3 turns on and node n7 starts to rise towards n5. Therefore, MN12 starts to conduct and increases the voltage of the bottom plate of the capacitor C3 to further increase the voltage on node n7. At steady state, capacitor C3 is connected between the gate and source terminals of MP3, and transistors MN11-14 are on with an overdrive voltage of Vdd. The nodes n4, n9, IN, n3, and out are all connected to each other via MN11-14.

It should be noted that, the bootstrapping starts with the turning on of MN8-9 to connect the gate of MP3 to the bottom plate of C3. But as the bottom plate voltage of C3 rises towards input voltage IN, MN8 and MN9 go off, and MN11 undertakes their mission.

The resistor R1 and the diodes D1-2 are necessary to protect the transistors MN11–MN14 from over voltage stress during the clock transitions.

When the switch starts to conduct, the output node follows the input node with a time constant consisting RC of the serial ON resistance of the transistors MN13 and MN14, and sampling load capacitance at the output node OUT. Thus, as the output node increases towards input voltage, the Vgs voltage of the transistors is

$$Vgs = V_{IN} + Vdd + V_{DS(MN13)} \quad \textbf{(3.1)}$$

Where $V_{DS(MN13)}$ is the voltage drop on the drain-source terminals of MN13. At steady state, this voltage drop is 0, but at the transition, this voltage will cause gate oxide voltage stress. In order to prevent this, the rise of the node n7 is slowed with the inclusion of the resistor R1. Ideally, both time constants, i.e. sampling RC and switch gate RC, should be designed equal. The selection of the value of R1 is important in terms of reliability and performance. It is necessary to choose a high enough resistor not to jeopardize the gate oxide reliability, and a small enough resistor value not to slow down the operation of the circuit.

It should be noted that the flying capacitor C3 does not go under voltage stress. Although each plate is exposed to high voltages, the voltage difference between the plates is always Vdd.

In addition to the R1, diodes D1 and D2 are necessary for protection during the on-off transition of the switch. In order to close the switch, the signal Φ goes from logic high to logic low. Therefore, the transistors MN15 and MN16 turn on and pull the nodes n8 and n7 down to ground. Let us assume that the capacitance on the node n7 is greater than the capacitance on node n8 (which is correct in terms of parasitic capacitances). In that case, MN15 would start to draw current from MN13 and MN14 before they are turned off by MN16. As a result, two possible outcomes may occur: The output node may be discharged, reducing the sampling performance, and the voltage of n8 may decrease so fast that the gate-source terminals of transistors MN13-14 would experience high voltage stress. In order to avoid these outcomes, the turning on of MN15 is delayed so that MN13 and MN14 are off, when MN15 is on. Still, depending on the process and mismatch variations, node n7 may go down much faster than node n8, again causing gate-source terminal over voltage. To prevent that happen, diodes D1 and D2 are introduced. As the node n7 lows down, if necessary, D1 and D2 turn on to pull node n8 down together with node n7. Thereby, the gate source voltage is limited to the sum of 2 on diode voltages.

As mentioned before, the main advantage of the bootstrapping switch over conventional transmission gate switch is that the bootstrapping switch technique keeps the gate overdrive of the switch transistor almost constant while input signal changes.

This statement is true only if the parasitic capacitance at node n7 is 0, or input signal is 0. Since neither of those is true, the gate overdrive of the proposed switch becomes a function of the input signal. Assuming Cp is the total parasitic capacitance loading n7, V_{GS} of MN13 and MN14 can be expressed as

$$V_{GS(M13)} = \frac{C_3}{C_3 + C_P} V_{DD} - \frac{C_P}{C_3 + C_P} V_{IN} \tag{3.2}$$

The first term in the equation (3.2) is a DC value and decreases the overdrive voltage of the output transistors. On the other hand, second term is an input dependent contributor to the overdrive voltage. Thus, it introduces distortion to sampling function.

The performance of this bootstrapped is switch is directly determined by the sampling speed, input signal range, and capacitor C3. For a given speed and input signal range the capacitor C3 should be carefully valued for necessary performance.

The switch is designed so that the maximum gate to source voltage of each transistor is limited with the supply voltage and only their drain terminal is exposed to the high voltage levels. The reliability of each device can be checked using Table 3. The sizes of devices used are given in Table 4.

Table 3: Ideal node voltages of each node of the HVSBS.

node	Φ	Φ’	node	Φ	Φ’
n1	2Vdd	Vdd	n6	Vdd-Vth	0
n2	Vdd	2Vdd	n7	V_{IN}+Vdd	0
n3	0	2Vdd	n8	V_{IN}	0
n4	V_{IN}	0	n9	V_{IN}	Vdd
n5	V_{IN}+Vdd	Vdd			

Table 4: Device sizes.

Transistor	width(μm)	length(μm)	Capacitor	Value(F)	Resistor	Value(Ω)
MN1	10	0.35	C1	1p	R1	10K
MN2	10	0.35	C2	1p		
MN3	10	0.35	C3	3p		
MN4	10	0.5				
MN5	10	0.5				
MN6	10	0.5				
MN7	10	0.5				
MN8	10	0.5				
MN9	10	0.5				
MN10	10	0.5				
MN11	10	0.5				
MN12	10	0.5				
MN13	10	0.5				
MN14	10	0.5				
MN15	100	0.5				
MN16	20	0.5				
MP1	20	0.35				
MP2	20	0.35				
MP3	20	0.35				

3.1.1 Simulation results

The voltages of the important nodes during the turn on instant are given in Figure 3.2. This simulation is conducted with a power supply of 2.5V. The bootstrapping mechanism can be observed easily on the voltage on node n7.

One period of a 16 Vp sine wave is sampled on a 4 pF load capacitor with HVSS and the transient waveforms are given in Figure 3.3.

The V_{GS} voltage of transistors M_{13} and M_{14} are given in Figure 3.4 for two periods of sine sampling. The mean value of V_{GS} during sampling is around 2 V, whereas the supply voltage V_{DD} is 2.5 V. Moreover, V_{GS} varies with the input signal. These observations approve the results given by the equation (3.2).

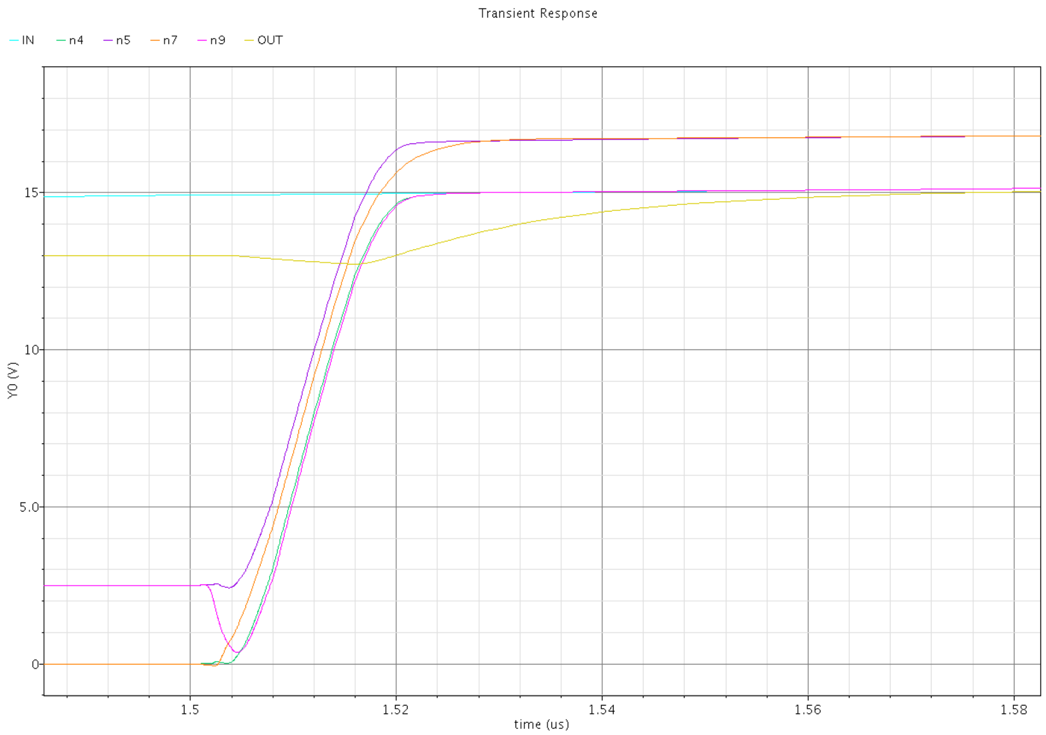


Figure 3.2: HVSS: OFF-ON transition.

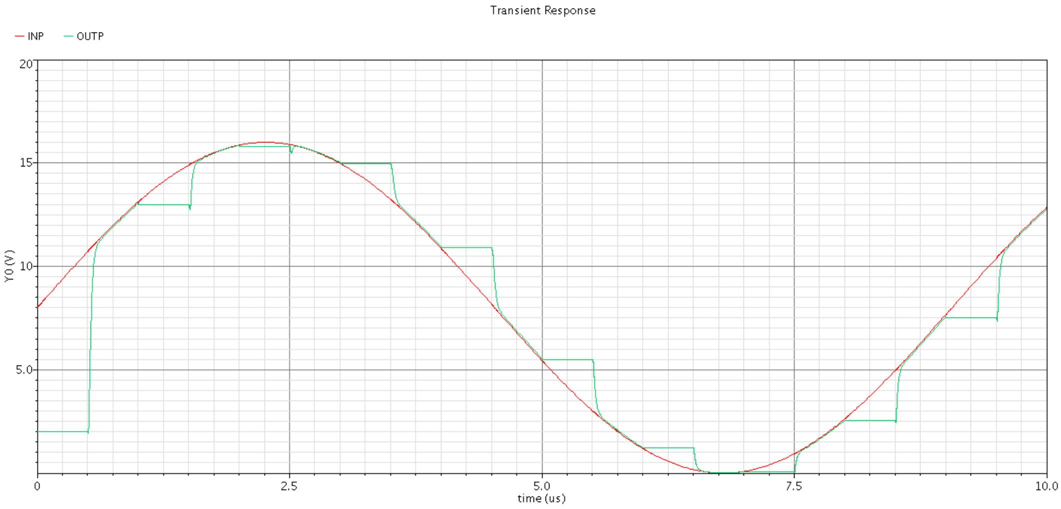


Figure 3.3: HVSS complete cycle sampling.

In addition, it can be observed that the vulnerable VGS terminals are never exposed to high voltage stress, except the first sampling instant. This peek is due to initial conditions of the circuit and is not expected to cause any harm since it is applied for a very short time (a few nanoseconds).

In order to further investigate the behavior of the circuit explained by equation (3.2), DC values of the V_{GS} voltage of transistors M_{13} and M_{14} is sampled and handled in

MATLAB. Throughout this simulation the supply voltage was kept at the nominal minimum value of 2.2V. The results are given in Figure 3.5.

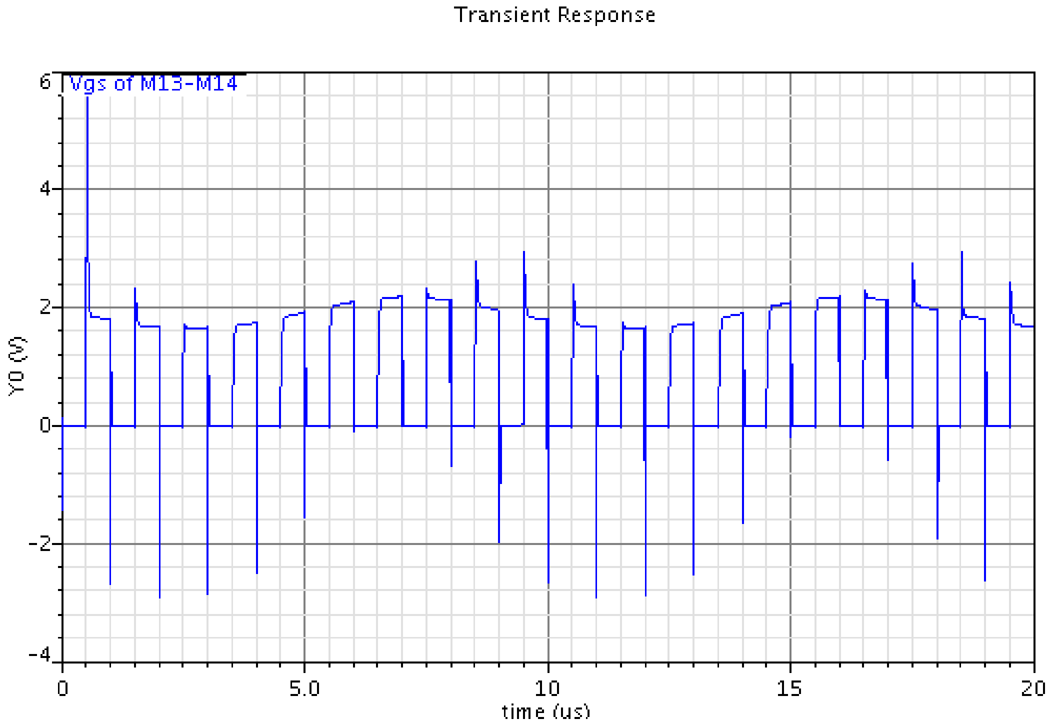


Figure 3.4: V_{GS} voltage of M13 and M14.

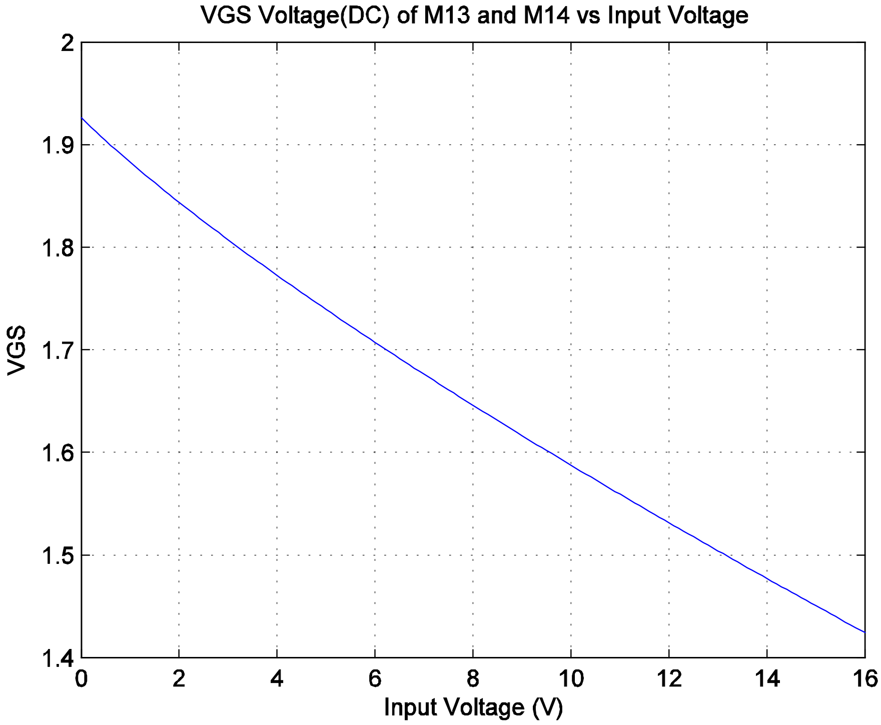


Figure 3.5: DC voltage of pass transistors M13 and M14 for vdd=2.2V.

3.2 Fully Differential Operational Amplifier

The fully differential amplifier architecture, implemented with plain 3.3V CMOS devices, is a folded cascode NMOS input, Miller compensated two-stage amplifier with continuous common mode feedback [14]. The transistor level schematic of the fully differential amplifier is given in Figure 3.6. The implementation details of the bias current circuitry are removed from the figure for the sake of simplicity. The exact copy of the schematics and layout can of the fully differential amplifier are given in APPENDIX A.1 (Figure A.3 and Figure A.4 respectively).

The transistors MN1 and MN2 constitute the differential input pair, whose differential current is steered on nodes n3 and n4 through the folded cascode stage composed of transistors MP1-4. The nodes n3 and n4 drive the gates of output transistors MN3 and MN4. The capacitors C_{C1} and C_{C2} are Miller capacitors to compensate the amplifier with nulling resistors R_{C1} and R_{C2}. The operational amplifier is properly compensated to drive capacitances of values up to 15 pF. The open loop DC gain of the amplifier is around 125 dB. The phase margin is around 56º and gain bandwidth product is 56 MHz. The ground current of the amplifier is 950 µA most of whose is used on the output stage (660 µA).

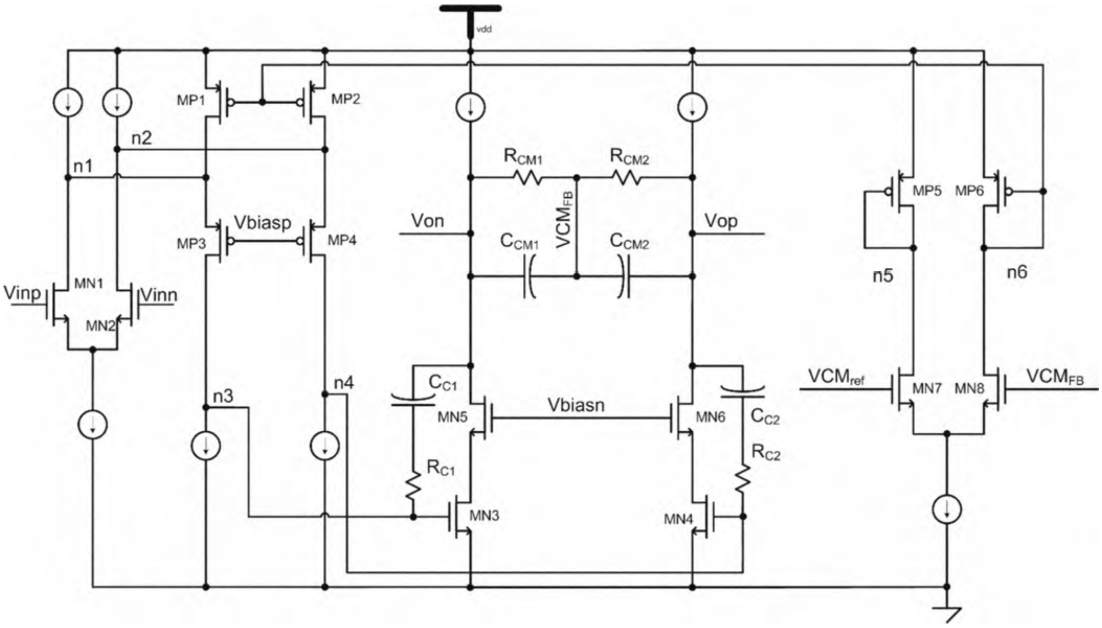


Figure 3.6: Fully differential operational amplifier schematics.

The supply voltage range for the amplifier is 2.3V-3.3V, with an output swing of 4 Vpp in the worst case.

R_{CM1} and R_{CM2} are large valued resistors used to detect the output common voltage VCM_{FB}. The capacitors C_{CM1} and C_{CM2} are used to improve high frequency response of the common mode feedback path. The common mode feedback path input differential pair consists of transistors MN7 and MN8. Reference common mode and output common mode signals are applied to the gates of MN7 and MN8, respectively. The voltage difference is reflected to the nodes n3 and n4 through the transistor MP6, MP1, and MP2. As a result, output common voltage is set to the reference voltage.

3.2.1 Comparator feature

The designed differential amplifier can be used as a comparator in the open loop mode, but the settling of the output nodes would be slow due to compensation capacitances especially for small input signals. Thus, a latched comparator circuit is integrated into the differential amplifier. The integrated comparator circuitry is given in Figure 3.7.

The input differential pair of the comparator consists of the input transistors of the differential OPAMP itself, in order to keep the input offset voltage constant. The load transistors MP9 and MP10 are small transistors with minimum gate length (6µm x 0.35µm) in order to keep the parasitic capacitance at minimum. Moreover, by limiting their size, their transconductance is much smaller than the input differential pair so that they do not contribute to the input offset voltage.

When the comparator operation is needed, a mode control signal generated by the digital control block “EN_COMP” signal is set to 0. So that the transistors MP5 and MP6 are turned off, and the differential current of the input transistors is steered away from the folded cascode stage of the opamp towards the load transistors of comparator MP9-MP10. The load transistor currents are mirrored by the transistors MP12 and MP13 on nodes OP and ON. When signal “LATCH” is at Vdd, the cross-coupled pairs MN3-MN4 and MP13-MP14 are off. When the signal goes to 0, the cross-coupled pairs constitute a positive feedback and the latch’s output is set. Although only positive end is used as the comparator output, both ends are loaded with inverters. In that way, both ends are equally loaded and isolated from the circuit the positive end drives.

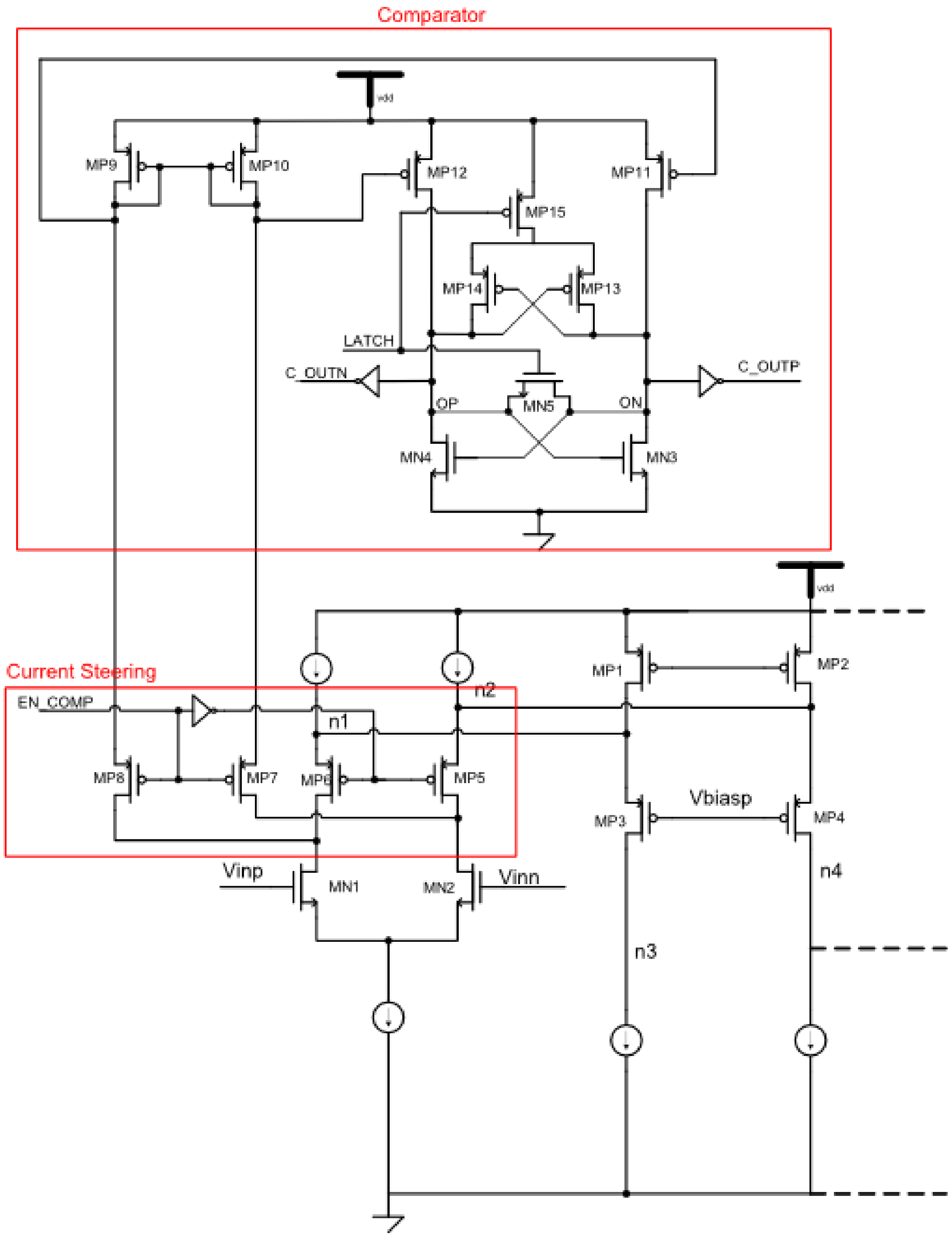


Figure 3.7: Comparator.

3.2.2 Adaptive compensation

In the SARADC process flow, after 8 bits of digital data is acquired, the residue is amplified by a factor of 64, and the amplified signal is resampled on the bottom plate of the capacity matrices. The configuration of the circuit of the amplification and resampling states are given in Figure 3.8.

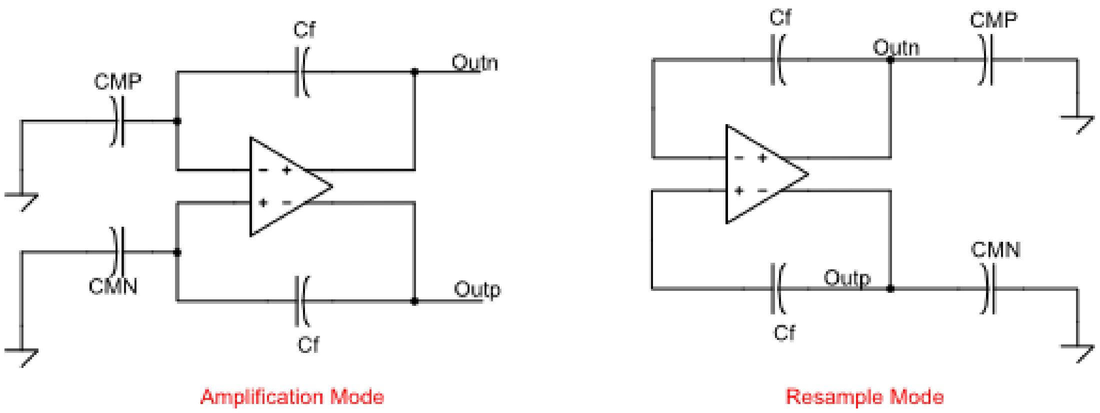


Figure 3.8: Amplification and resample configurations.

In the amplification configuration, the output of the opamp is loaded with only feedback capacitors Cf, whose capacitance are 1/64 of the equivalent capacitance of capacity matrices CMP and CMN. In addition, feedback signal is 1/64 of the output signal.

In the resampling configuration, the opamp is loaded with the entire capacity matrices CMP and CMN in unity gain feedback mode.

It is obvious that the amplifier is loaded with different load capacitances and with different feedback gains. In order to ensure stability the amplifier should be compensated according to the heaviest load and least gain factor, which is the Resample Configuration in this case. In that case, the opamp would be overcompensated for the light load and high gain factor case, amplification mode, and the operation speed would be limited by the unnecessary large compensation capacitances. As a result, the sampling rate would be reduced.

For both safe and fast operation, adaptive compensation technique is introduced. The technique is to switch on and off extra compensation capacity when necessary. This technique is shown in Figure 3.9. In the figure, transistors MN3 and MN5; nodes n3, Vbiasn, and Von, and nulling resistor R_{C1} are the same transistors as in Figure 3.6. The compensation capacitor of the original opamp is separated into 2 capacitors C_{CA} and C_{CR} in the modified version. Only one side of the differential output stage is given in the figure for simplicity. The same modification is applied on the other side as well.

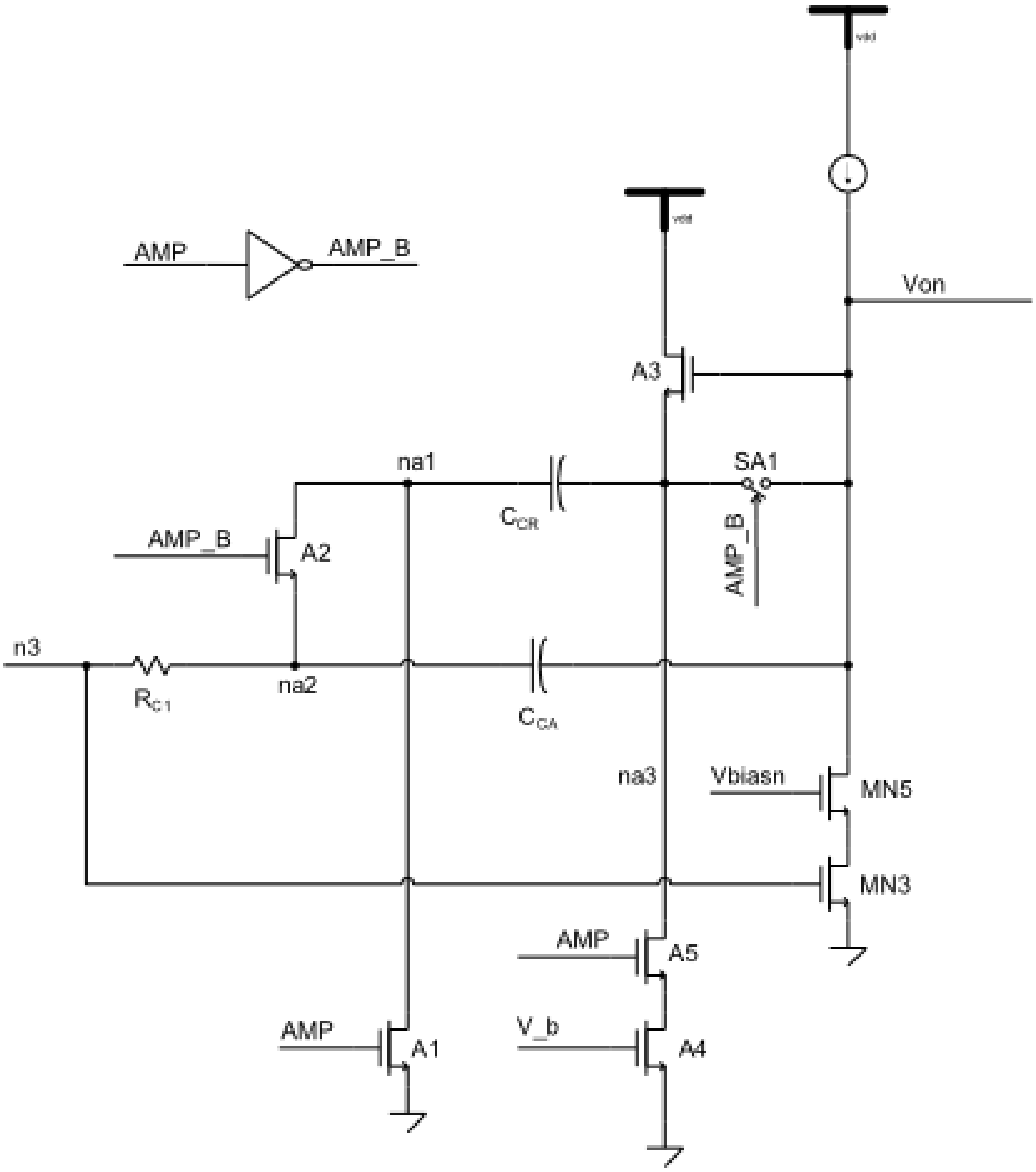


Figure 3.9: Adaptive compensation technique.

The adaptive compensation technique is implemented with transistors A1-5; switch SA1, which is a simple T-gate; and digital control signal AMP. AMP_B is the inverse of the control signal AMP. The signal V_b is the common bias signal used within to bias current sink sources.

The operation is as follows:

When the opamp is in resampling mode, AMP signal is equal to 0, turning transistors A1 and A5 off; and transistor A2 and switch SA1 on. Thus, the capacitor C_{CR} is in parallel to C_{CA}. Since SA1 shorts the source and gate terminals of transistor A3, it is off as well.

When the control signal AMP is set to Vdd, the switch SA1 and transistor A2 are off to separate the compensation capacitor C_{CR} from the rest of the circuit. It should be noted that the separated capacitor is not left floating. While the transistor A1 sets its top plate to 0V, the source follower A3 sets its bottom plate to Von-VgsA3, where VgsA3 is the gate-source voltage of the transistor A3. So that the total voltage difference across the capacitor is equal to Von-VgsA3. This value is especially important in transition from amplification state to resample state. In order to understand why this value is important, the voltage difference of the capacitor CCR must be investigated in the resampling mode: The voltage of the bottom plate is equal to Von and the voltage of the top plate is equal to VgsMN3, where VgsMN3 is the gate-source voltage of the transistor MN3. Subtracting the 2 values, the voltage across this capacitor is found to be Von-VgsMN3. It can be assumed that the gate-source voltages of transistors MN3 and A3 are equal to each other. So, the capacitor is precharged to its final value not to disturb the circuit during the transition from amplification to resampling mode.

The capacitance of C_{CA} is 135 fF, whereas CCR is 3.5 pF, a significant reduction in compensation capacitance!

3.2.3 Simulation results

In order to investigate the stability of the operational amplifier, a differential AC signal is applied to the positive and negative inputs of the amplifier for both unity gain and amplification configurations. Each of the differential outputs is loaded with the proper load, which is 15 pF for unity gain and 1 pF for amplification mode. The open loop AC response of the opamp is given in Figure 3.10. The red curves indicate unity gain configuration AC response, while the blue curves indicate amplification configuration AC response. The open loop gain of the amplification configuration is divided by the gain factor, 64, in order to find the actual feedback signal. The phase margin for both configurations is above 50^o.

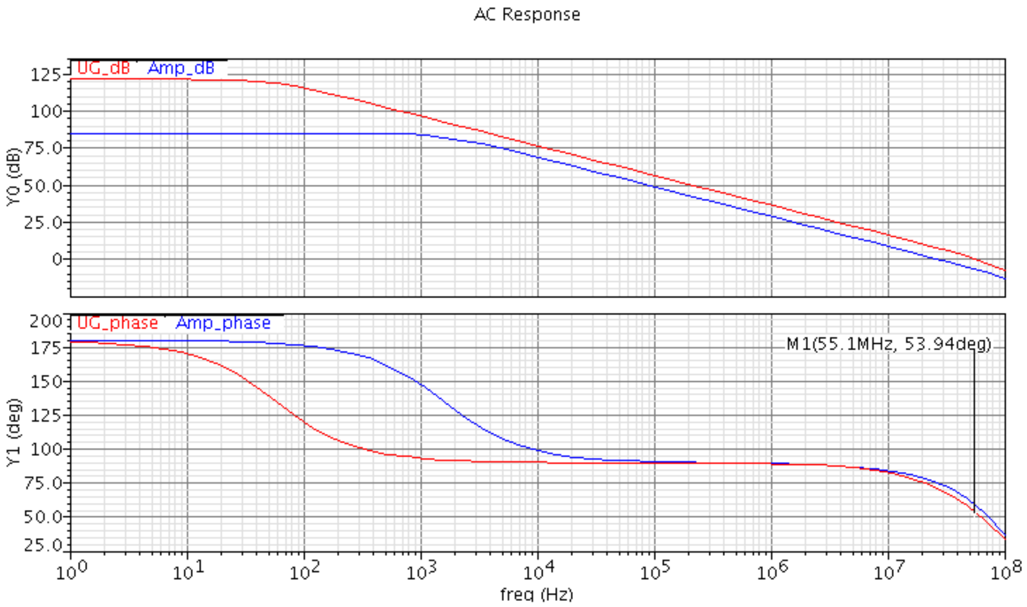


Figure 3.10: Open loop AC response for both unity gain and amplification modes.

In order to investigate the adaptive compensation large signal behavior, a differential step of 20 mV is applied to the amplification configuration to get an output step response of 1.28V for both adaptive and standard compensation. The improvement in the step response of the adaptive compensation technique can be observed in Figure 3.11. The blue curve is the step response of non-adaptive (standard) compensation mode, and the red curve is the step response of adaptive compensation mode. The settling time is reduced to less than 100 ns from 1μs, an improvement of more than 10 times!

The comparator feature of the operational amplifier is simulated by applying an input signal of 2mVpp at $2V_{DC}$ common mode voltage sine wave. The clock frequency was 100 MHz, twice the actual clock frequency. The positive output of the comparator is observed. The simulation results are provided in Figure 3.12. The output is available at the negative levels of the clock. Comparing the output logic value with the input signal, it can be observed that no mistakes were made by the comparator. As a result it can be conclude that the resolution and operating speed of the designed comparator are better than needed.

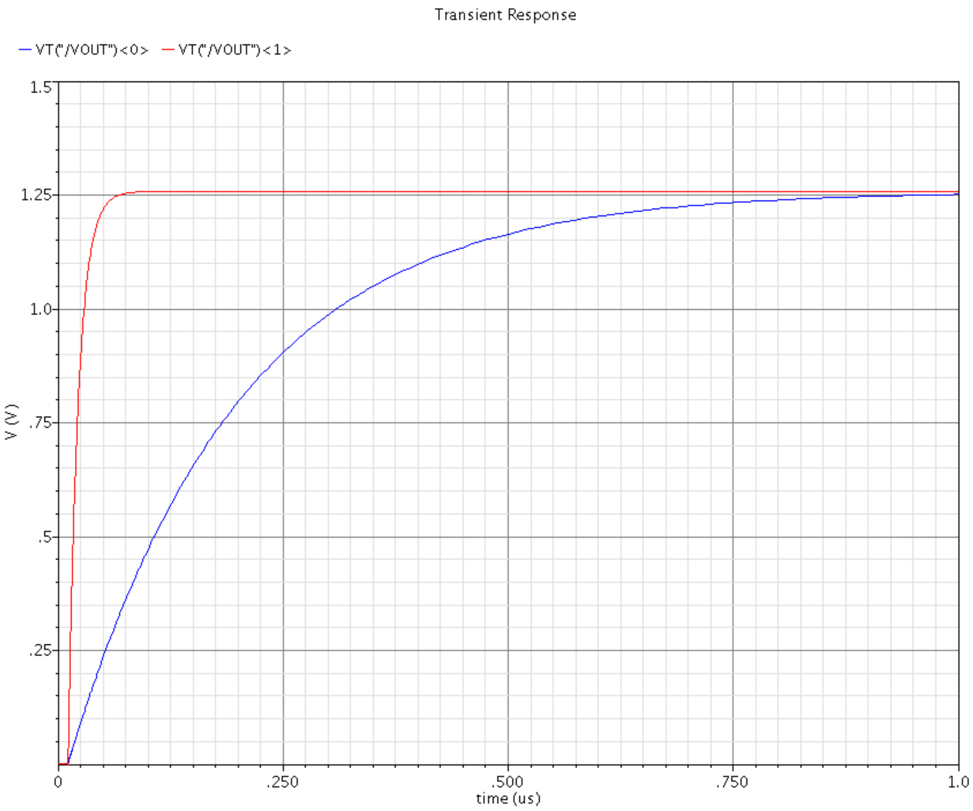


Figure 3.11: Step response of adaptive compensation and standard compensation.

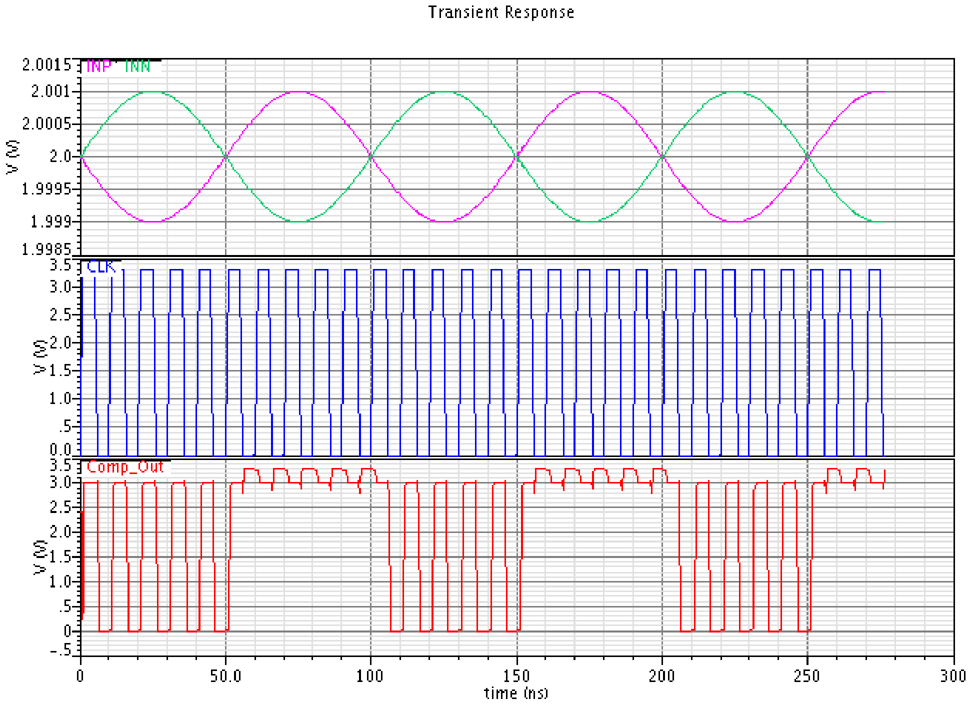


Figure 3.12: Transient simulation results: Opamp in comparator mode.

3.3 Capacitive Array

In the design, an 8-bit binary weighted capacitive array is used as a digital to analog converter. A conventional capacitive divider array is given in Figure 3.13.

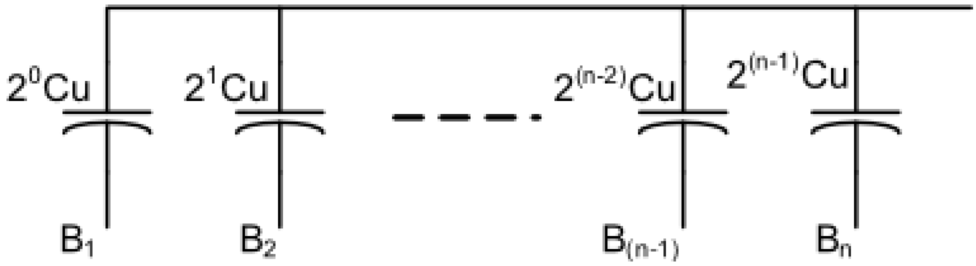


Figure 3.13: Classical binary weighted capacitive DAC.

A drawback of the capacitive divider architectures is that the number of the element increases exponentially with the number of bits. Since technological limits and matching requirements determine the minimum size of the unity capacitance an increase in the number of elements, enlarges the silicon area and capacitive load.

An alternative configuration to solve this problem is to add an attenuator in the middle of the array [15-16]. This way the capacitive spread is reduced. For an 8 bit DAC the capacitance spread reduction is from 2^7 to 2^3 (16 times). The use of attenuator is given in Figure 3.14 [15].

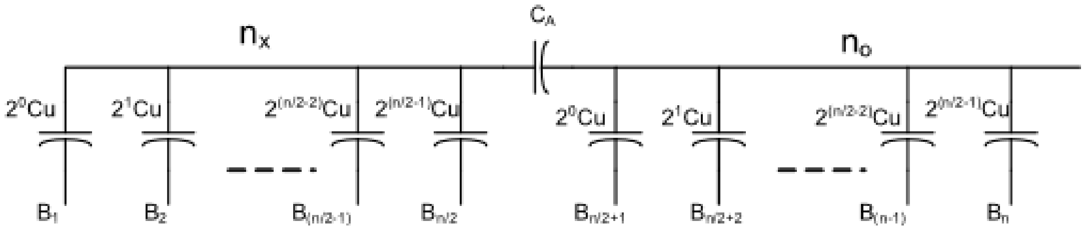


Figure 3.14: The use of an attenuator in the middle of the array.

In order to show that C_A should be exactly equal to Cu, following derivations are conducted assuming zero initial charge for all of the capacitors. When one of the bits (mth bit) goes to logic 1, the output voltage can be found using charge conservation rule:

- If the bit is at the left hand side of the array (m<n/2):

$$Vx\left(2^{n/2}-1-2^{(m-1)}\right)Cu+(Vx-Vp)2^{(m-1)}Cu+(Vx-Vo)Cu=0$$

$$\Rightarrow Vx\left(2^{n/2}-1\right)Cu+(-Vp)2^{(m-1)}Cu+(Vx-Vo)Cu=0$$

$$\Rightarrow Vx2^{n/2}Cu-Vp2^{(m-1)}Cu-VoCu=0$$

$$\Rightarrow Vx2^{n/2}-Vp2^{(m-1)}-Vo=0 \qquad \textbf{(3.3)}$$

$$Vo\left(2^{n/2}-1\right)Cu+(Vo-Vx)Cu=0$$

$$\Rightarrow Vo2^{n/2}Cu-VxCu=0$$

$$\Rightarrow Vx=Vo2^{n/2} \qquad \textbf{(3.4)}$$

Where Vp is the positive logic 1 voltage, whereas Vx is the voltage at node n_x and Vo is the voltage at node n_o, after the change of logic. Equations (3.13) and (3.14) define the charge conservation rule for nodes n_x and n_o, respectively. Using equation (3.4) in equation (3.3):

$$\Rightarrow Vo2^{n/2}2^{n/2}-Vp2^{(m-1)}-Vo=0$$

$$\Rightarrow Vo2^{n}-Vp2^{(m-1)}-Vo=0$$

$$\Rightarrow Vo\left(2^{n}-1\right)-Vp2^{(m-1)}=0$$

$$\Rightarrow Vo=Vp\frac{2^{(m-1)}}{\left(2^{n}-1\right)} \qquad \textbf{(3.5)}$$

- If the bit is at the right hand side of the array (m>n/2):

$$Vx\left(2^{n/2}-1\right)Cu+(Vx-Vo)Cu=0$$

$$\Rightarrow Vx=Vo2^{-n/2} \qquad \textbf{(3.6)}$$

$$Vo\left(2^{n/2}-1-2^{(m-n/2-1)}\right)Cu+(Vo-Vp)2^{(m-n/2-1)}Cu+(Vo-Vx)Cu=0$$

$$\Rightarrow Vo2^{n/2}Cu-Vp2^{(m-n/2-1)}Cu-VxCu=0$$

$$\Rightarrow Vo2^{n/2}-Vp2^{(m-n/2-1)}-Vx=0 \qquad \textbf{(3.7)}$$

Equations (3,6) and (3.7) define the charge conservation rule for nodes n_x and n_o, respectively. Using equation (3.6) in equation (3.7):

$$\Rightarrow Vo2^{n/2} - Vp2^{(m-n/2-1)} - Vo2^{-n/2} = 0$$

$$\Rightarrow Vo = \frac{2^{(m-n/2-1)}}{2^{n/2} - 2^{-n/2}} Vp$$

$$\Rightarrow Vo = Vp\frac{2^{(m-1)}}{2^n - 1} \qquad \textbf{(3.8)}$$

Equations (3.5) and (3.8) agree that if C_A is chosen to be equal to Cu, the coefficient of the mth bit is proportional to $2^{(m-1)}$. Due to the linearity of the circuit, for any particular digital input the output can be found by exploiting the superposition rule.

While realizing the capacitive array, one of the most important considerations is that, the parasitic capacitance on node n_x should be kept at minimum for the equations (3.5) and (3.8) to hold. In order to meet that requirement, the unit capacitor is realized with its top plate completely surrounded by its bottom plate. As a result, all the parasitic capacitances are on or among the bottom plates of the capacitors. However, it should be noted that the parasitic capacitance at the output node does not contribute to the non-linearity, but modify the output function by a gain factor.

The unit capacitor is divided into 4 identical capacity cells, so that the unit capacitor is distributed among the layout to prevent gradient errors. The algorithm of distributing DAC elements against possible gradient errors is presented by Randall L. Geiger [17].

So far, no exact simulation method or tool for the gradient error analysis is known to the author. Neither parasitic extraction nor statistical Monte-Carlo analysis give any information about any possible gradient effects within the layout.

The layout of the capacity cell is given in Figure 3.15.

Top plate of the capacity cell consists of layers poly2 and metal1 (can be seen in the middle of the layout). Top plate is covered by layers poly1 from bottom, metal2 from top, and poly1-poly2-metal1 from around. Top plate connections are allowed through the holes at the three edges.

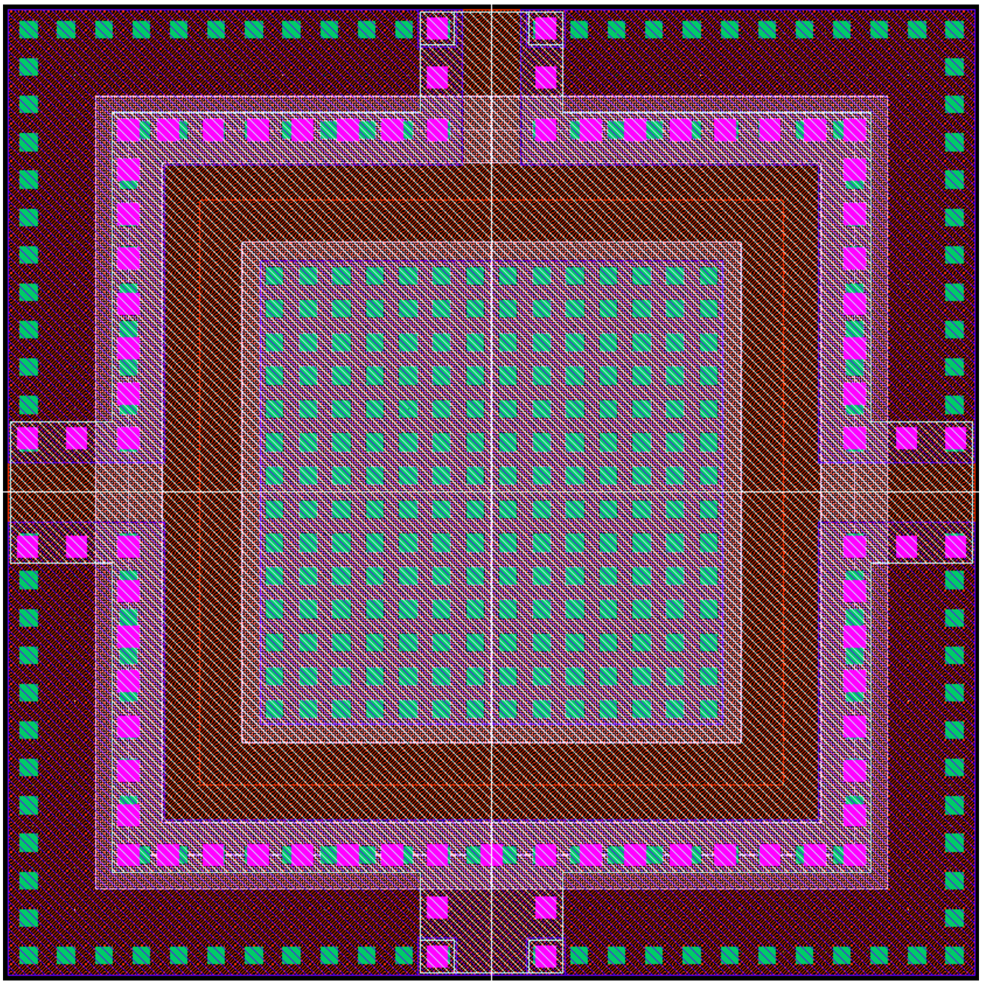


Figure 3.15: Layout of the capacity cell.

The top view layout of the capacitive array surrounded by the bottom plate driving circuitry is given in Figure 3.16. The MSB and LSB parts of positive and negative sides of the differential capacitive array are labeled with red squares around them. Moreover, around the capacitive array, analog reference channel and switches are surrounded by green and blue loops around them. Attenuator capacitors are marked with yellow rectangles.

In order to reduce edge effects, capacitive array is surrounded by a dummy capacitor matrix. The outmost capacitors are dummy.

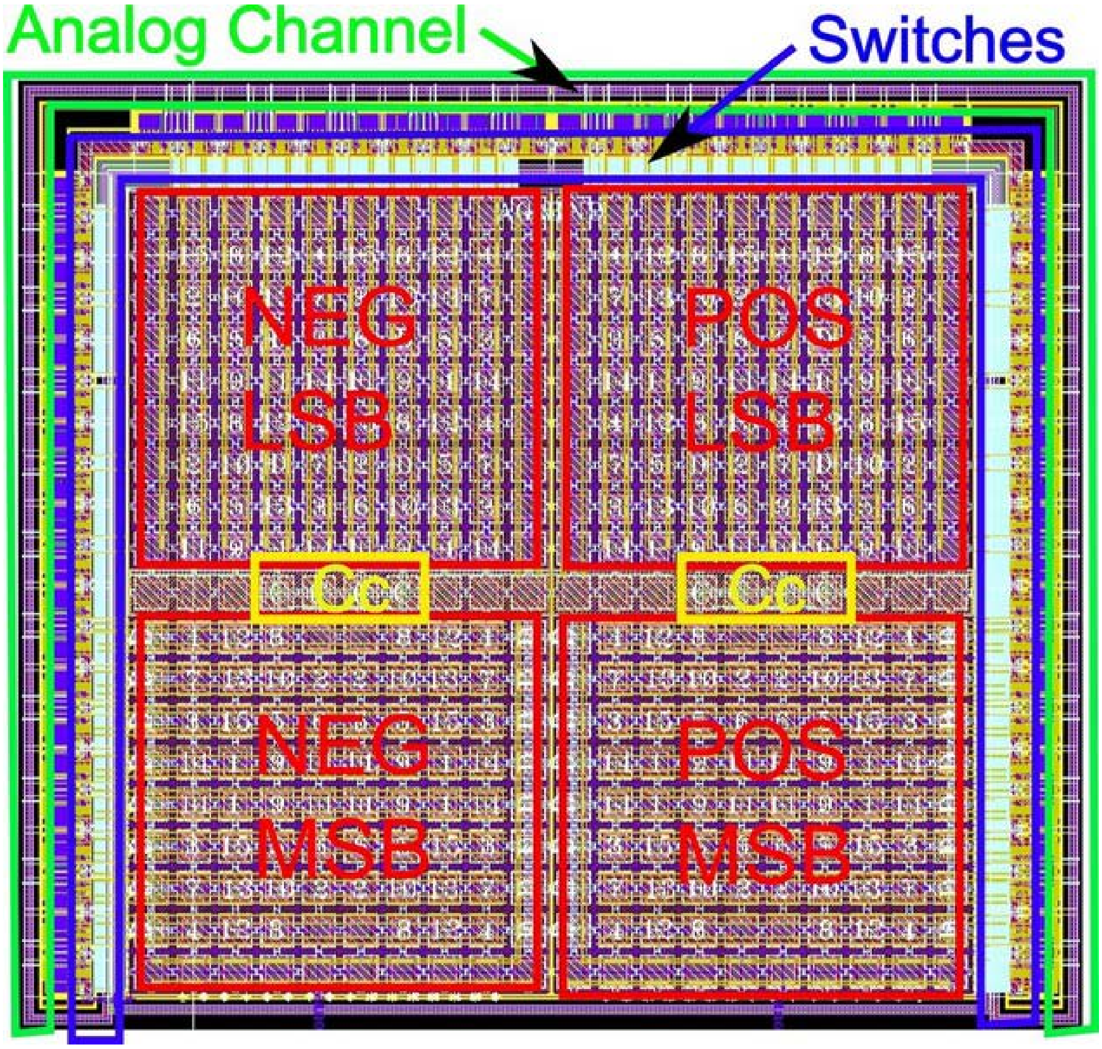


Figure 3.16: Top view layout of the capacitive array with bottom plate driving circuitry around.

3.4 Digital Part

The digital part of the SARADC is composed of an I2C slave and a state machine. State machine is the part that generates control signals for the SARADC algorithm to operate properly. The algorithm will be explained in detail in section 4. ALGORTIHM. The state machine is designed using verilog hardware description language. The code is given in APPENDIX A.2.

In order to utilize in-chip test and control structures and intervene if necessary; I2C slave is designed and integrated. I2C slave controls trim bits and connects necessary internal signals to the test pins of the chip. A detailed test procedure and I2C input configuration will be given in chapter 7.

3.5 Bandgap Reference and Power on Reset (POR)

The bandgap reference is used to generate the bias current of the differential opamp. The circuit details are not provided in this thesis because bandgap reference design is not in the scope of this study. Also a power on reset is used to generate a reset signal to the I2C when the power is ramped, so that the default values of the parameters in the circuit are provided correctly.

Both bandgap and power on reset circuits are directly taken from previous projects. The schematics and layouts of the bandgap reference and power on reset circuits are provided in the APPENDIX A.1 (Figures A.5-8).

4. ALGORITHM

The state flow diagram of the subranging successive approximation ADC is given in Figure 4.1.

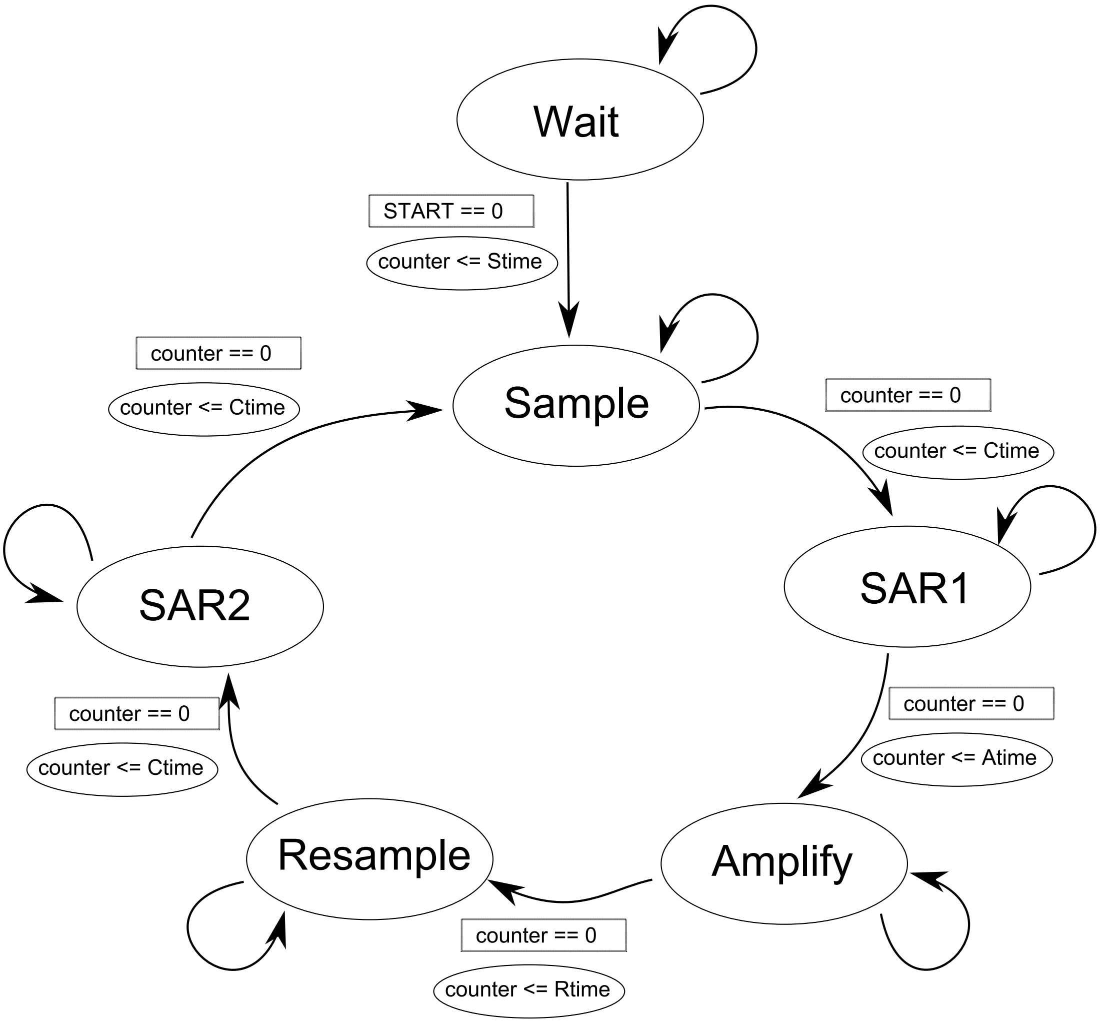


Figure 4.1: State flow diagram.

Each state transition is controlled by the register "counter". This register, as the name implies, is a back counter, with each clock it counts down. At the end of each state, the counter is loaded with the parameter indicating how long the next state will last. Among these parameters, only "Ctime", indicating conversion time, is deterministic and not needed to be changed. Its value is 8, which is the number of bits obtained at each conversion state, SAR1 and SAR2. The other parameters will be loaded through

I2C, in order to optimize the sample time vs. performance trade-off. As an example, the settling time of the amplifier is 100 ns, so with a clock signal of 50 MHz, the amplification state needs to be 5 clock cycles long. Therefore, the parameter “Atime” should be 5. The reason why these parameters are not determined and kept as variables is that, there may be deviations from what is expected by the simulation results. For example, when testing the circuit, the settling time of the amplifier may be longer than expected due to process drifts. Then, the variable “Atime” is simply set to 6 via I2C.

The function of each state can be summarized as follows:

- **Wait**: The state machine waits for the “START” signal to start the conversion algorithm.
- **Sample**: This is the state when analog input signal is sampled on the sampling capacitors. The HVSBS is activated in this state.
- **SAR1**: This is the first conversion state. SAR algorithm is carried out and most significant 8 bits are obtained.
- **Amplify**: This is the state when the residue signal on capacitive array is amplified via the differential amplifier.
- **Resample**: The amplified residue signal is resampled on the capacity matrix.
- **SAR2**: This is the second conversion state. Another 8 bits of data is acquired.

In the SAR1 and SAR2 states, the successive approximation register (SAR) block is activated to generate digital data.

The state flow diagram of the SAR block is given in Figure 4.2.

At the steady state digital data is set to the midcode which is “10000000” for 8 bits. When the algorithm is activated, the comparator decides whether the 1st approximation is greater than (Comp = 1) or smaller than (Comp = 0) the actual signal. If the first approximation is greater, the most significant bit is set to 0, and next significant bit is set to 1 (making a new approximation of “01000000”). If not, the next significant bit is set to 1 (making a new guess of “11000000”). At each clock cycle, another digital bit value is determined.

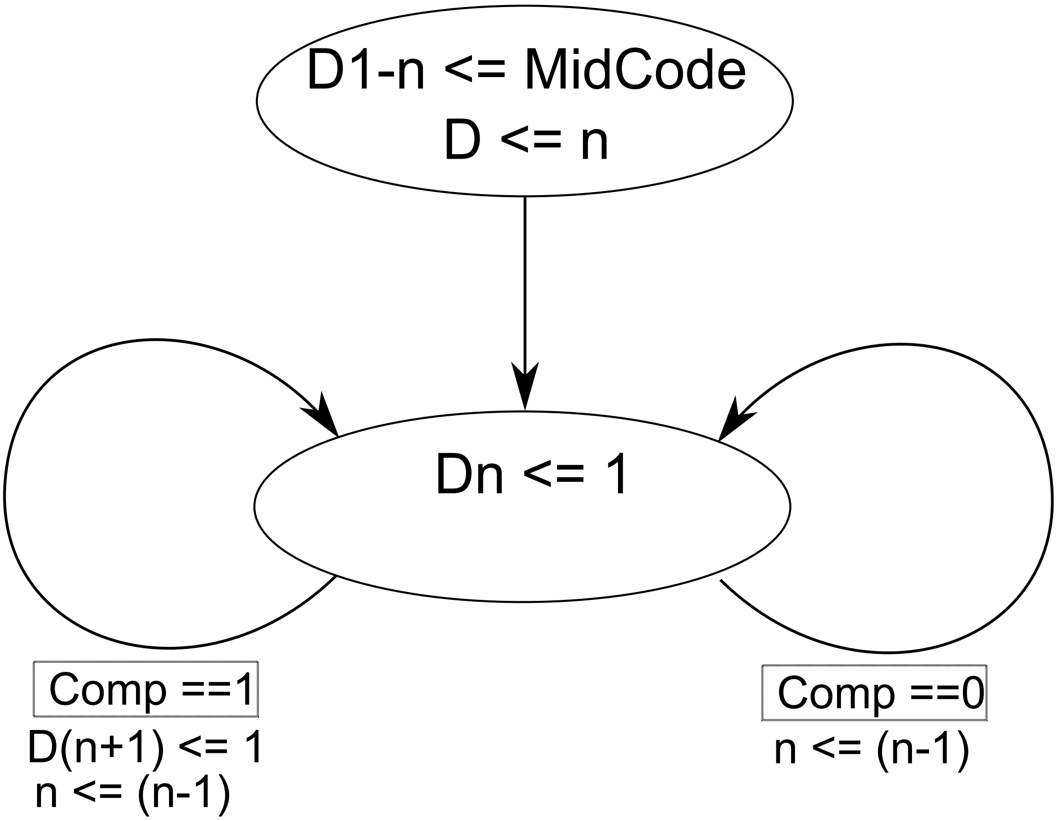


Figure 4.2: SAR algorithm.

The top-level schematic of the designed SARADC is given in Figure 4.3. In the figure, the signals RSMPL, PSMPL, SAMPLE, T&H_B, and FB are generated by the digital state machine who realizes the algorithm given in Figure 4.1. The verilog code is provided in the Appendix A.2. The signals refp, refn, and VCM are DC voltages constituting positive reference, negative reference and common mode voltage, respectively. High voltage differential input signals are INp and INn. COMP signal is the output of the comparator inside the opamp.

In order to have a better understanding of the state flow and circuit configuration, top-level signal values and switch conditions will be presented. In the following illustrations, the logic values of the signals are given in red circles. Moreover, the switches that are “OFF” are red, while “ON” switches are kept black. For simplicity, the whole capacity matrices are not drawn, they are represented with bold capacitors CMp and CMn.

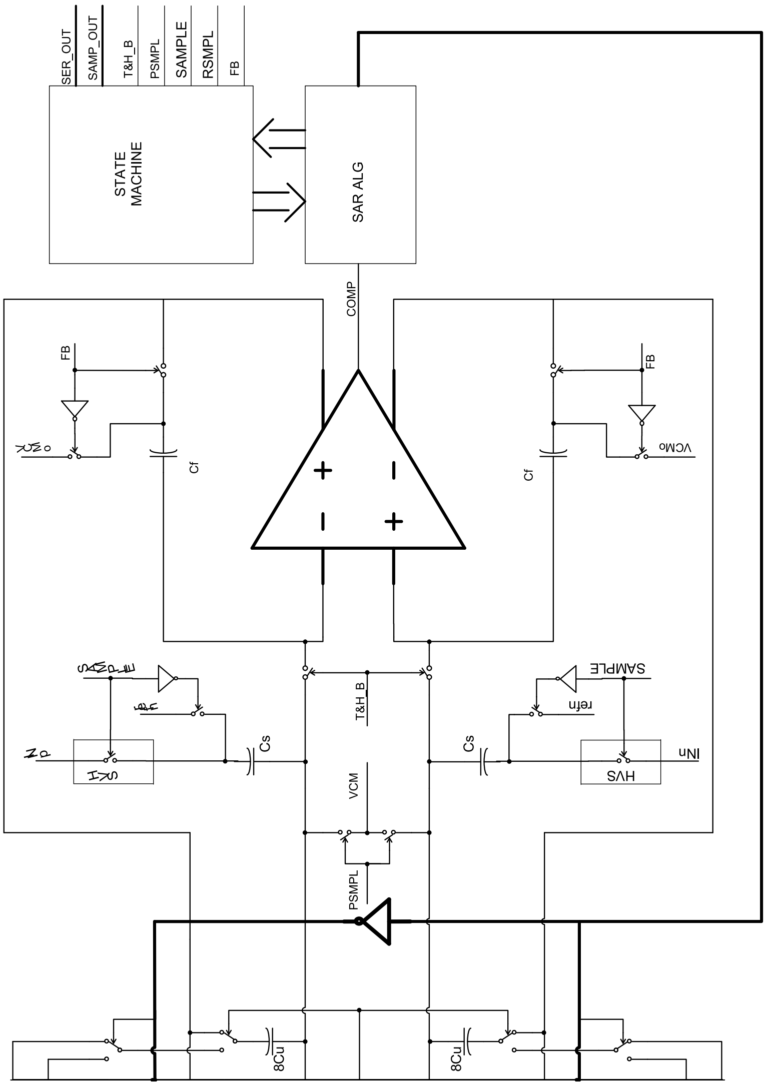


Figure 4.3: Top level schematic of the designed SARADC.

4.1 Sample State

The circuit configuration of the sample state is given in Figure 4.4.

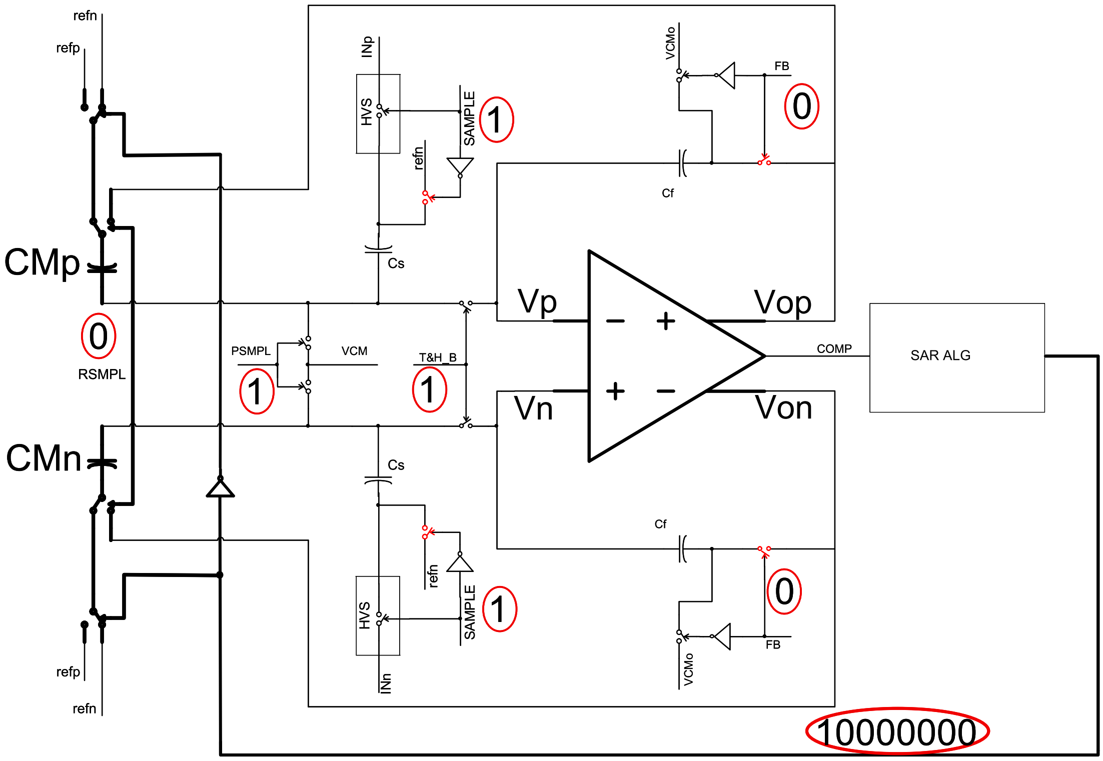


Figure 4.4: Sample state.

In this state, high voltage sampling switches (named HVS on figure) are ON and inputs are sampled on sampling capacitors Cs. The digital signal applied to the capacity matrix is kept at midcode “10000000”.

At the end of the sampling state, first PSMPL signal goes to 0 so that the nodes Vp and Vn are left floating and the input sampling is done at that moment. After that, one clock cycle later, high voltage sampling switches are closed and bottom plates of the sampling capacitors are connected to negative reference voltage refn.

4.2 SAR State

After the sampling state, first conversion state takes place. In this state differential amplifier is in comparator configuration and SAR Algorithm is activated. The digital output is obtained according to the comparator output. The circuit configuration is given in Figure 4.5.

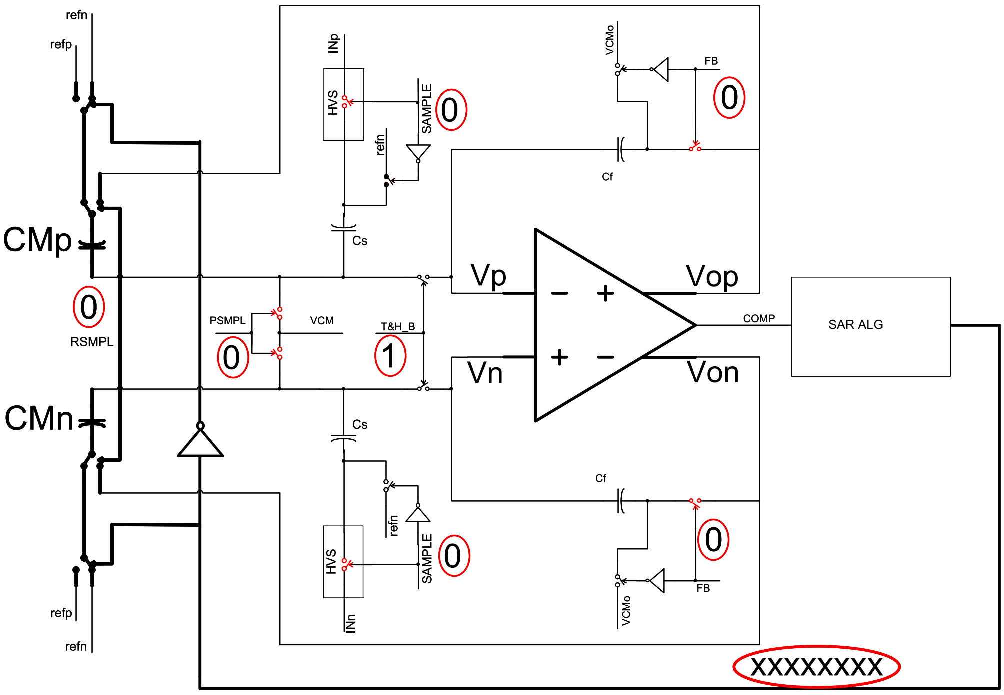


Figure 4.5: SAR state.

4.3 Amplification State

After the first conversion state, the amplification state comes. The circuit configuration for this state can be seen in Figure 4.6.

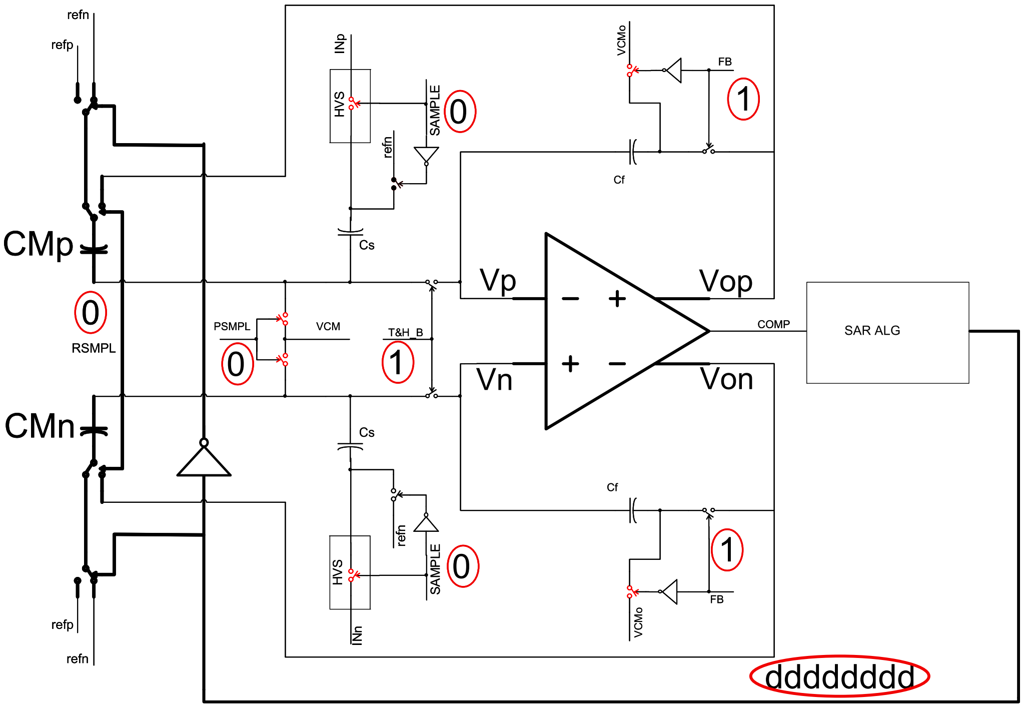


Figure 4.6: Amplification state.

The digital bits determined at the SAR state are kept at the constant at “dddddddd”. At this state, the comparator configuration of the differential amplifier is switched off. In addition, capacitors Cf are connected between the output and input nodes of the differential amplifier to provide a feedback path. Using this feedback path the differential amplifier zeros the residue signal at its input nodes, generating an output voltage of amplified residue signal.

4.4 Resample State

The configuration for this state is given in Figure 4.7. In this state, the differential amplifier is separated from the capacitor matrices by the signal T&H_B. This way the differential amplifier is in hold mode and keeps the amplified signal at its output. Then the output voltages are resampled on the capacity matrix through the signals RSMPL and PSMPL to get another 8 bits of data.

After this state, another SAR algorithm is run and another 8 bits are achieved. The configuration is just the same as the first SAR state.

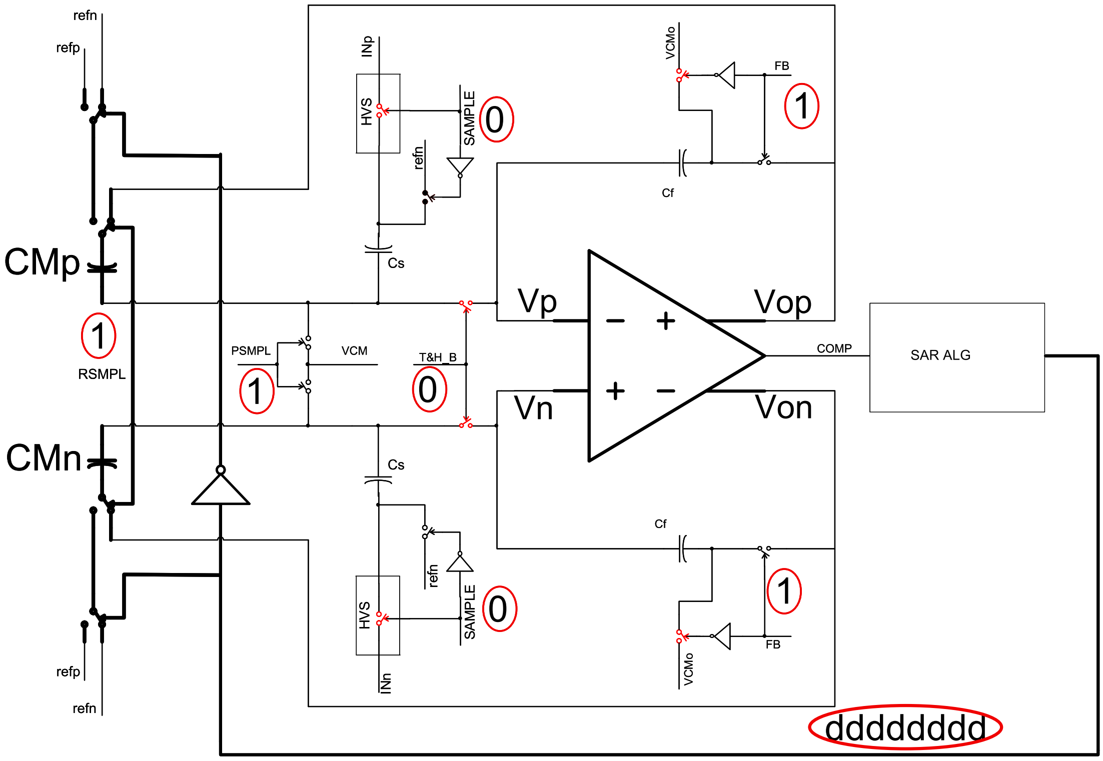


Figure 4.7: Resample state.

4.5 Full Conversion Cycle

A full cycle simulation is conducted to show the node voltages at important nodes of the proposed SARADC. The simulation results are given in **Figure 4.8**. The signals named CS_P-CS_N are bottom plates of the positive and negative sampling capacitors; TOP_P-TOP_N are top plates of the positive and negative capacity arrays; and AMP_P-AMP_N are the outputs of the operational amplifier.

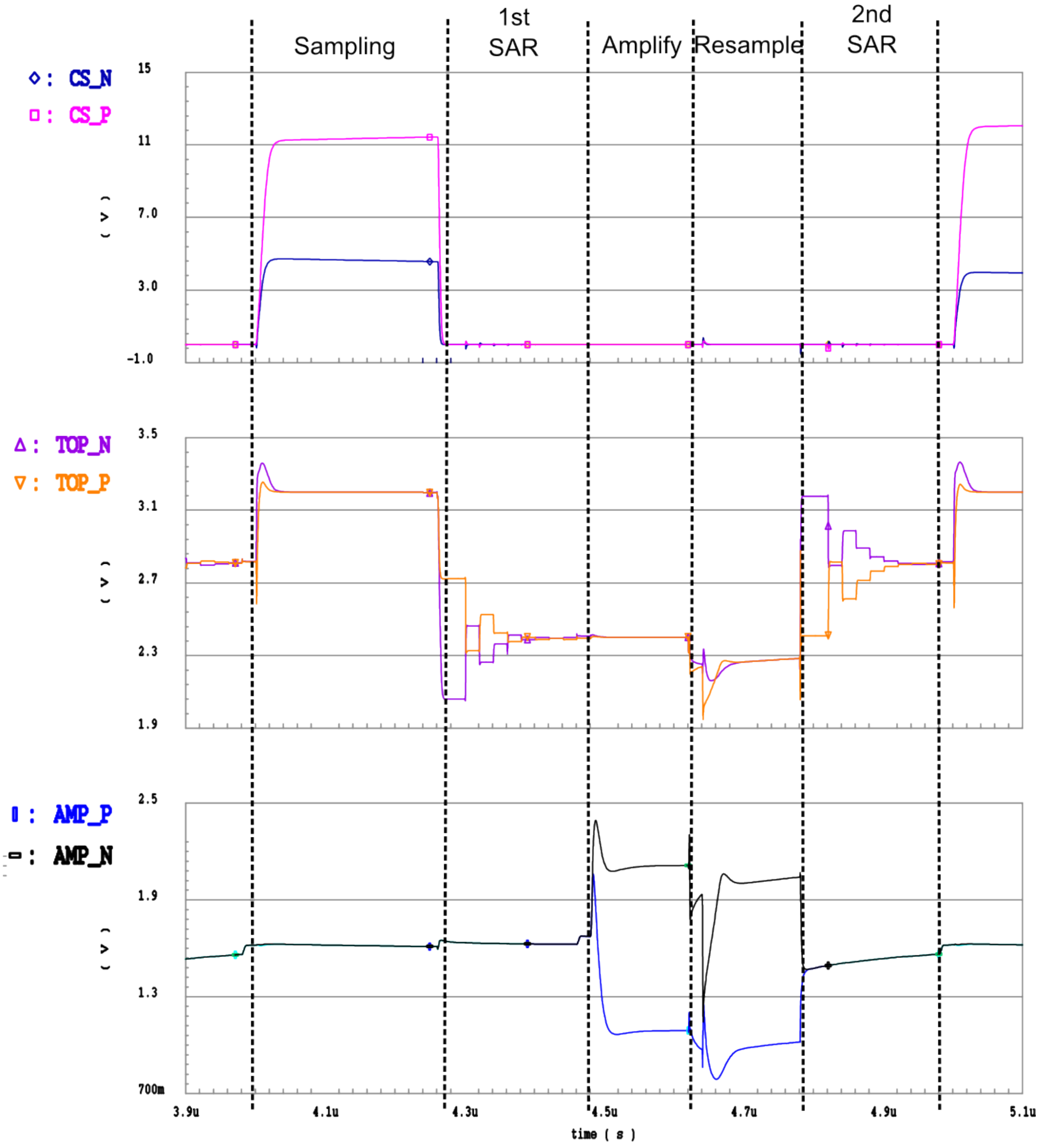


Figure 4.8: Full cycle conversion.

5. OBTAINING TRANSFER FUNCTION OF SARADC

In this section, the mathematical verification of the applied analog to digital conversion method will be discussed. Parasitic capacitances and their effect on the accuracy will be analyzed. The circuit used for the mathematical analysis is given in Figure 5.1.

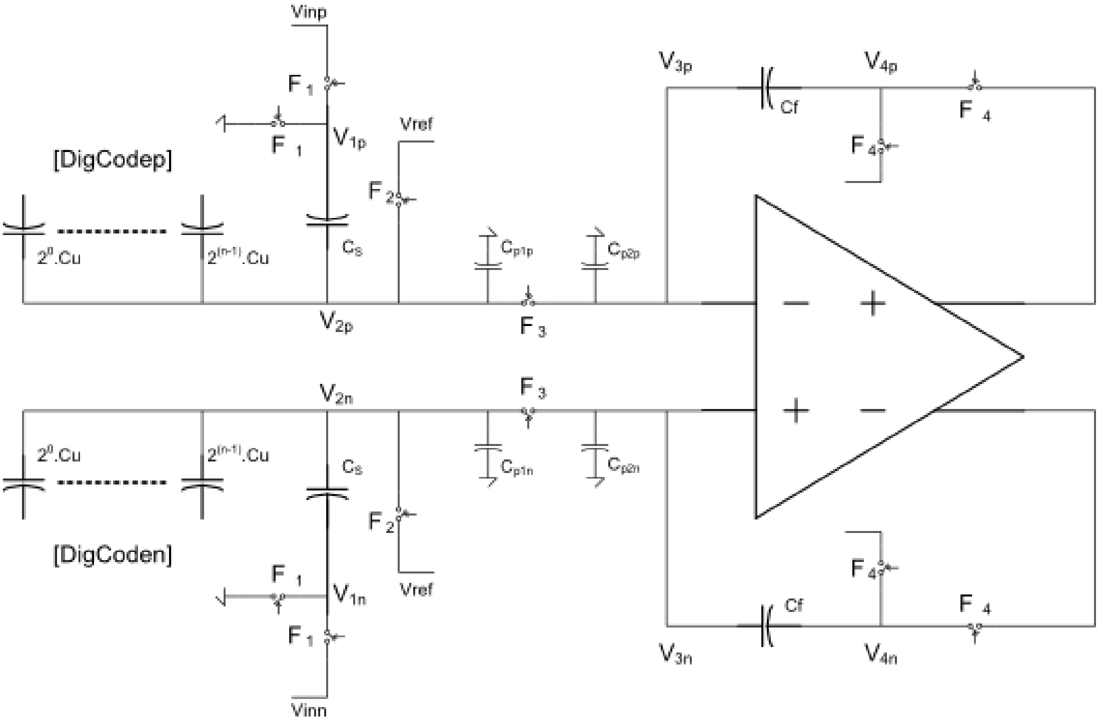


Figure 5.1: SARADC.

The capacitive matrix is modeled as a binary weighted capacitive DAC for simplicity. It was demonstrated that the used DAC structure has exactly the same weight as the binary weighted DAC unless the parasitic capacitance on the node n_x is small enough. This parasitic capacitance is kept at minimum with careful layout, so its effect is neglected throughout the analyses. Moreover, it was discussed that the bottom plate capacitances are not important at all since they are connected to low impedance nodes, just like the parasitic capacitance at the output of the differential amplifier. The other capacitances on the high impedance nodes Cp1p, Cp2p, Cp1n and Cp2n are the parasitic capacitances on the nodes they are connected to. Φ_1, Φ_2,

Φ_3, and Φ_4 illustrates state control signals, whereas Φ_4' is the inverse of Φ_4; and Φ_1' is the inverse of Φ_1.

[DigCodep] and [DigCoden] are the logical input vectors applied to the bottom plates of the capacity matrix. They are 8 bit long and the leftmost bit is applied to the most significant capacitor. [DigCoden] is the 1's complement of the vector [DigCodep], that is if [DigCodep] = [10110010], then [DigCoden] = [01001101]. In that case the minus sign will be applied as follows:

[Digcodep]-[DigCoden] = [10110010] -[01001101] = [1 -1 1 1 -1 -1 1 -1].

Vrefn is accepted as 0 (which is not actually an assumption, since Vrefn is going to be connected to 0 V) through the calculations for the sake of simplicity. IF Vrefn is wanted to be kept as a variable, one could easily replace $[matrix].Vrefp$ with $[matrix].Vrefp + [\overline{matrix}].Vrefn$ throughout the calculations.

5.1 Sampling Phase

The control signals are:

$$\Phi_1 : 1(on)$$
$$\Phi_2 : 1(on)$$
$$\Phi_3 : 1(on)$$
$$\Phi_4 : 1(on)$$

The node voltages are:

$V1p : Vinp$ $\quad V3p : Vref$

$V1n : Vinn$ $\quad V3n : Vref$

$V2p : Vref$ $\quad V4p : VCMo$

$V2n : Vref$ $\quad V4n : VCMo$

And digital data applied to the capacitive DAC array are:

$$[DigCodep] = 10000000$$
$$[DigCoden] = 01111111$$
$$[\overline{DigCodep}] = [DigCoden]$$

In that case, the node pairs V2p-V3p and V2n-V3n are connected together and the charge on those common nodes are given as follows:

$$Q_{Sp} = (V_{2p} - V_{1p}).C_S + (V_{2p} - [DigCodep].Vrefp).Bin.Cu$$
$$+ (V_{3p} - V_{4p}).Cf + (V_{2p} - 0).(C_{p1p} + C_{p2p})$$

$$Q_{Sp} = (Vref - Vinp).C_S + (Vref - [10000000].Vrefp).Bin.Cu$$
$$+ (Vref - VCMo).Cf + (Vref - 0).(C_{p1p} + C_{p2p}) \qquad \textbf{(5.1)}$$

$$Q_{Sn} = (V_{2n} - V_{1n}).C_S + (V_{2n} - [DigCoden].Vrefp).Bin.Cu$$
$$+ (V_{3n} - V_{4n}).Cf + (V_{2n} - 0).(C_{p1n} + C_{p2n})$$

$$Q_{Sn} = (Vref - Vinn).C_S + (Vref - [01111111].Vrefp).Bin.Cu$$
$$+ (Vref - VCMo).Cf + (Vref - 0).(C_{p1n} + C_{p2n}) \qquad \textbf{(5.2)}$$

Where *Bin* is a vertical binary weighted vector:

$$Bin = \begin{bmatrix} 2^{(8-1)} \\ 2^{(7-1)} \\ 2^{(6-1)} \\ 2^{(5-1)} \\ 2^{(4-1)} \\ 2^{(3-1)} \\ 2^{(2-1)} \\ 2^{(1-1)} \end{bmatrix}$$

And the subtraction of a matrix from a variable is done as the following example:

$$V_{2p} - [01100010] = \lfloor (V_{2p} - 0)(V_{2p} - 1)(V_{2p} - 1)(V_{2p} - 0)(V_{2p} - 0)(V_{2p} - 0)(V_{2p} - 1)(V_{2p} - 0) \rfloor$$

5.2 1st Conversion Phase

The control signals are:

1)$\ldots\Phi_2 : 1 \Rightarrow 0$

2)$\ldots\Phi_1 : 1 \Rightarrow 0$

$\ldots\ldots\Phi_3 : 1$

$\ldots\ldots\Phi_4 : 1$

Assuming that the successive approximation method operated without any mistakes, at the end of the first conversion state, the node voltages are:

$V1p : Vinp \Rightarrow 0$
$V1n : Vinn \Rightarrow 0$
$V2p : Vref \Rightarrow Vresp$
$V2n : Vref \Rightarrow Vresn$
$V3p : Vref \Rightarrow Vresp$
$V3n : Vref \Rightarrow Vresn$
$V4p : VCMo$
$V4n : VCMo$

And the digital data applied to the capacitive DAC array are:

$$\begin{aligned}
&[DigCodep] = [DigCode1p] \\
&[DigCoden] = [DigCode1n] \\
&[\overline{DigCode1p}] = [DigCode1n]
\end{aligned}$$

In that case the charges on the common nodes V2p-V3p and V2n-V3n are:

$$\begin{aligned}
Qresp &= (V_{2p} - V_{1p}).C_S + (V_{2p} - [DigCodep].Vrefp).Bin.Cu \\
&+ (V_{3p} - V_{4p}).Cf + (V_{2p} - 0).(C_{p1p} + C_{p2p})
\end{aligned}$$

$$\begin{aligned}
Qresp &= (Vresp - 0).C_S + (Vresp - [DigCode1p].Vrefp).Bin.Cu \\
&+ (Vresp - VCMo).Cf + (Vresp - 0).(C_{p1p} + C_{p2p})
\end{aligned} \tag{5.3}$$

$$\begin{aligned}
Qresn &= (V_{2n} - V_{1n}).C_S + (V_{2n} - [DigCoden].Vrefp).Bin.Cu \\
&+ (V_{3n} - V_{4n}).Cf + (V_{2n} - 0).(C_{p1n} + C_{p2n})
\end{aligned}$$

$$\begin{aligned}
Qresn &= (Vresn - 0).C_S + (Vresn - [DigCode1n].Vrefp).Bin.Cu \\
&+ (Vresn - VCMo).Cf + (Vresn - 0).(C_{p1n} + C_{p2n})
\end{aligned} \tag{5.4}$$

5.3 Amplification Phase

The control signals are:

$\ldots\ldots\Phi_2 : 0$
$\ldots\ldots\Phi_1 : 0$
$\ldots\ldots\Phi_3 : 1$
$\ldots\ldots\Phi_4 : 1 \Rightarrow 0$

At the end of the amplification phase the node voltages are:

$V1p : 0 \Rightarrow 0$
$V1n : 0 \Rightarrow 0$
$V2p : Vresp \Rightarrow Vx$
$V2n : Vresn \Rightarrow Vx$
$V3p : Vresp \Rightarrow Vx$
$V3n : Vresn \Rightarrow Vx$
$V4p : VCMo \Rightarrow Vampp$
$V4n : VCMo \Rightarrow Vampn$

And the digital data applied to the capacitive DAC array are still:

$$
\begin{aligned}
&[DigCodep] = [DigCode1p] \\
&[DigCoden] = [DigCode1n] \\
&\left[\overline{DigCode1p}\right] = [DigCode1n]
\end{aligned}
$$

In that case the charges on the common nodes V2p-V3p and V2n-V3n are:

$$
\begin{aligned}
&Qampp = \left(V_{2p} - V_{1p}\right).C_S + \left(V_{2p} - [DigCodep].Vrefp\right).Bin.Cu \\
&+ \left(V_{3p} - V_{4p}\right).Cf + (V_{2p} - 0).\left(C_{p1p} + C_{p2p}\right)
\end{aligned}
$$

$$
\begin{aligned}
&Qampp = (Vx - 0).C_S + (Vx - [DigCode1p].Vrefp).Bin.Cu \\
&+ (Vx - Vampp).Cf + (Vx - 0).\left(C_{p1p} + C_{p2p}\right)
\end{aligned}
\tag{5.5}
$$

$$
\begin{aligned}
&Qampn = \left(V_{2n} - V_{1n}\right).C_S + \left(V_{2n} - [DigCoden].Vrefp\right).Bin.Cu \\
&+ \left(V_{3n} - V_{4n}\right).Cf + (V_{2n} - 0).\left(C_{p1n} + C_{p2n}\right)
\end{aligned}
$$

$$
\begin{aligned}
&Qampn = (Vx - 0).C_S + (Vx - [DigCode1n].Vrefp).Bin.Cu \\
&+ (Vx - Vampn).Cf + (Vx - 0).\left(C_{p1n} + C_{p2n}\right)
\end{aligned}
\tag{5.6}
$$

At the end of the sampling phase, the nodes V2p-V2n and V3p-V3n are left floating by the signal Φ_1. From that moment up to the end of the amplification phase, the net charges entering those two nodes are 0 (by the charge conservation rule). Thus, the charges given by equations 5.1-5.5 and 5.2-5.6 are equal:

$$
\begin{aligned}
&Q_{Sp} = Qampp \\
&Q_{Sn} = Qampn \\
&\Rightarrow Q_{Sp} - Q_{Sn} = Qampp - Qampn
\end{aligned}
$$

$$
\begin{aligned}
&(Vref - Vinp).C_S + (Vref - [10000000].Vrefp).Bin.Cu + (Vref - VCMo).Cf + (Vref - 0).\left(C_{p1p} + C_{p2p}\right) \\
&- (Vref - Vinn).C_S - (Vref - [01111111].Vrefp).Bin.Cu - (Vref - VCMo).Cf - (Vref - 0).\left(C_{p1n} + C_{p2n}\right) \\
&= (Vx - 0).C_S + (Vx - [DigCode1p].Vrefp).Bin.Cu + (Vx - Vampp).Cf + (Vx - 0).\left(C_{p1p} + C_{p2p}\right) \\
&- (Vx - 0).C_S - (Vx - [DigCode1n].Vrefp).Bin.Cu - (Vx - Vampn).Cf - (Vx - 0).\left(C_{p1n} + C_{p2n}\right)
\end{aligned}
$$

Making proper operations the equation is simplified as follows:

$$
\begin{aligned}
&(Vinn - Vinp).C_S - ([00000001].Vrefp).Bin.Cu \\
&+ Vref.\left(\left(C_{p1p} + C_{p2p}\right) - \left(C_{p1n} + C_{p2n}\right)\right) \\
&= (([DigCode1n] - [DigCode1p]).Vrefp)Bin.Cu \\
&+ (Vampn - Vampp).Cf + Vx.\left(\left(C_{p1p} + C_{p2p}\right) - \left(C_{p1n} + C_{p2n}\right)\right)
\end{aligned}
$$

So the amplified signal can be represented as follows:

$$
\begin{aligned}
&(Vampp - Vampn) \\
&= \left(-\left([DigCode1p]-[DigCode1n]\right).Vrefp\right)Bin.\frac{Cu}{Cf} + (Vinp - Vinn).\frac{C_S}{Cf} \qquad \textbf{(5.7)} \\
&+ \left([00000001].Vrefp\right).Bin.\frac{Cu}{Cf} + (Vx - Vref).\frac{\left((C_{p1p}+C_{p2p}) - (C_{p1n}+C_{p2n})\right)}{Cf}
\end{aligned}
$$

Note that the input common mode voltage seems to appear at the transfer function. However, at the end of the analysis, it will be shown that the last term cancels itself out and does not appear at the total transfer function.

5.4 Resampling Phase

In this phase, signal Φ_3 separates the capacitive array from the differential amplifier and the amplified signal is resampled on the capacitive DACs. The control signals of on this phase are:

$\Phi_1 : 0$
$\Phi_2 : 0 \Rightarrow 1$
$\Phi_3 : 1 \Rightarrow 0$
$\Phi_4 : 0$

At the end of the resampling phase the node voltages are:

$V1p : 0 \Rightarrow 0$	$V3p : Vx \Rightarrow Vx$
$V1n : 0 \Rightarrow 0$	$V3n : Vx \Rightarrow Vx$
$V2p : Vx \Rightarrow Vref$	$V4p : Vampp \Rightarrow Vampp$
$V2n : Vx \Rightarrow Vref$	$V4n : Vampn \Rightarrow Vampn$

The digital signals are not used in this phase; instead, the outputs of the differential amplifier are applied to the capacitive DAC.

Since node pairs V2p-V3p and V2n-V3n are separated from each other, each node will have its own charge equation. In the following equations, the charges on nodes V2p, V3p, V2n, and V3n will be denoted as QRSp2, QRSp3, QRSn2, and QRSn3, respectively:

$$QRSp_2 = (Vref - 0).C_S + (Vref - [11111111].Vampp).Bin.Cu + (Vref - 0).C_{p1p} \qquad \textbf{(5.8)}$$

$$QRSn_2 = (Vref - 0).C_S + (Vref - [11111111].Vampn).Bin.Cu + (Vref - 0).C_{p1n} \qquad \textbf{(5.9)}$$

$$QRSp_3 = (V_{3p} - 0).C_{p2p} + (V_{3p} - Vampp).Cf$$

$$QRSp_3 = (Vx - 0).C_{p2p} + (Vx - Vampp).Cf \tag{5.10}$$

$$QRSn_3 = (V_{3n} - 0).C_{p2n} + (V_{3n} - Vampn).Cf$$

$$QRSn_3 = (Vx - 0).C_{p2n} + (Vx - Vampn).Cf \tag{5.11}$$

5.5 2nd Conversion Phase

At the end of the resampling phase, the control signal Φ_2 goes to 0, leaving the nodes V2p and V2n floating to finish the resampling state. At the second conversion phase, the control signals are as follows:

$\Phi_1 : 0$
$\Phi_2 : 1 \Rightarrow 0$
$\Phi_3 : 0 \Rightarrow 1$
$\Phi_4 : 0 \Rightarrow 1$

Assuming that the successive approximation method operated without any mistakes, at the end of the second conversion state the node voltages are:

$V1p : 0 \Rightarrow 0$
$V1n : 0 \Rightarrow 0$
$V2p : Vref \Rightarrow Vres2p$
$V2n : Vref \Rightarrow Vres2n$
$V3p : Vx \Rightarrow Vres2p$
$V3n : Vx \Rightarrow Vres2n$
$V4p : Vampp \Rightarrow VCMo$
$V4n : Vampn \Rightarrow VCMo$

And the digital data applied to the capacitive DAC array are:

$[DigCodep] = [DigCode2p]$
$[DigCoden] = [DigCode2n]$
$[\overline{DigCode2p}] = [DigCode2n]$

In that case the charges on the common nodes V2p-V3p and V2n-V3n are:

$$\begin{aligned} Qres2p &= (V_{2p} - V_{1p}).C_S + (V_{2p} - [DigCodep].Vrefp).Bin.Cu \\ &+ (V_{3p} - V_{4p}).Cf + (V_{2p} - 0).(C_{p1p} + C_{p2p}) \end{aligned}$$

$$\begin{aligned} Qres2p &= (Vres2p - 0).C_S + (Vres2p - [DigCode2p].Vrefp).Bin.Cu \\ &+ (Vres2p - VCMo).Cf + (Vres2p - 0).(C_{p1p} + C_{p2p}) \end{aligned} \tag{5.12}$$

$$\begin{aligned} Qres2n &= (V_{2n} - V_{1n}).C_S + (V_{2n} - [DigCoden].Vrefp).Bin.Cu \\ &+ (V_{3n} - V_{4n}).Cf + (V_{2n} - 0).(C_{p1n} + C_{p2n}) \end{aligned}$$

$$\begin{aligned} Qres2n &= (Vres2n-0).C_S + (Vres2n-[DigCode2n].Vrefp).Bin.Cu \\ &+ (Vres2n-VCMo).Cf + (Vres2n-0).(C_{p1n}+C_{p2n}) \end{aligned} \tag{5.13}$$

By the charge conservation rule:

$$QRSp_2 + QRSp_3 = Qres2p$$
$$QRSn_2 + QRSn_3 = Qres2n$$

$$(QRSp_2 + QRSp_3) - (QRSn_2 + QRSn_3) = Qres2p - Qres2n$$
$$(QRSp_2 - QRSn_2) + (QRSp_3 - QRSn_3) = Qres2p - Qres2n$$

Utilizing equations (5.8)-(5.13) in the equation above:

$$\begin{aligned} &\begin{pmatrix} ((Vref-0).C_S + (Vref-[11111111].Vampp).Bin.Cu + (Vref-0).C_{p1p}) \\ -((Vref-0).C_S + (Vref-[11111111].Vampn).Bin.Cu + (Vref-0).C_{p1n}) \end{pmatrix} \\ &+ (((Vx-0).C_{p2p} + (Vx-Vampp).Cf) - ((Vx-0).C_{p2n} + (Vx-Vampn).Cf)) \\ &= \begin{pmatrix} (Vres2p-0).C_S + (Vres2p-[DigCode2p].Vrefp).Bin.Cu \\ + (Vres2p-VCMo).Cf + (Vres2p-0).(C_{p1p}+C_{p2p}) \end{pmatrix} \\ &- \begin{pmatrix} (Vres2n-0).C_S + (Vres2n-[DigCode2n].Vrefp).Bin.Cu \\ + (Vres2n-VCMo).Cf + (Vres2n-0).(C_{p1n}+C_{p2n}) \end{pmatrix} \end{aligned}$$

After proper simplifications:

$$\begin{aligned} &(-[11111111].(Vampp-Vampn)).Bin.Cu \\ &+ Vref.(C_{p1p}-C_{p1n}) + Vx.(C_{p2p}-C_{p2n}) - (Vampp-Vampn).Cf \\ &= (Vres2p-Vres2n).C_S \\ &+ ((Vres2p-Vres2n) - ([DigCode2p]-[DigCode2n]).Vrefp).Bin.Cu \\ &+ (Vres2p-Vres2n).Cf + Vres2p.(C_{p1p}+C_{p2p}) - Vres2n.(C_{p1n}+C_{p2n}) \end{aligned}$$

where the matrix operation

$$[11111111].Bin = 2^n - 1$$

Would lead to:

$$\begin{aligned} &(-(Vampp-Vampn)).((2^n-1)Cu + Cf) + Vref.(C_{p1p}-C_{p1n}) + Vx.(C_{p2p}-C_{p2n}) \\ &= (Vres2p-Vres2n).C_S \\ &+ ((Vres2p-Vres2n) - ([DigCode2p]-[DigCode2n]).Vrefp).Bin.Cu \\ &+ (Vres2p-Vres2n).Cf + Vres2p.(C_{p1p}+C_{p2p}) - Vres2n.(C_{p1n}+C_{p2n}) \end{aligned} \tag{5.14}$$

Now inserting equation (5.7) into equation (5.14):

$$
\begin{aligned}
&-\begin{pmatrix}
(-([DigCode1p]-[DigCode1n]).Vrefp)Bin.\dfrac{Cu}{Cf} \\
+(Vinp-Vinn).\dfrac{C_S}{Cf}+([00000001].Vrefp).Bin.\dfrac{Cu}{Cf} \\
+(Vx-Vref).\dfrac{((C_{p1p}+C_{p2p})-(C_{p1n}+C_{p2n}))}{Cf}
\end{pmatrix}.((2^n-1)Cu+Cf) \\
&+Vref.(C_{p1p}-C_{p1n})+Vx.(C_{p2p}-C_{p2n}) \\
&=(Vres2p-Vres2n).C_S+((Vres2p-Vres2n)-([DigCode2p]-[DigCode2n]).Vrefp).Bin.Cu \\
&+(Vres2p-Vres2n).Cf \\
&+Vres2p.(C_{p1p}+C_{p2p})-Vres2n.(C_{p1n}+C_{p2n})
\end{aligned}
$$

Leaving (Vinp-Vinn) at the left hand side of the equation an exact transfer function is given as:

$$
\begin{aligned}
&-(Vinp-Vinn).\frac{C_S}{Cf}.((2^n-1).Cu+Cf) \\
&=(Vres2p-Vres2n).C_S \\
&+((Vres2p-Vres2n)-([DigCode2p]-[DigCode2n]).Vrefp).Bin.Cu \\
&+(Vres2p-Vres2n).Cf+Vres2p.(C_{p1p}+C_{p2p})-Vres2n.(C_{p1n}+C_{p2n}) \\
&-((([DigCode1p]-[DigCode1n]).Vrefp)Bin.\frac{Cu}{Cf}.((2^n-1).Cu+Cf) \\
&+([00000001].Vrefp).Bin.\frac{Cu}{Cf}.((2^n-1).Cu+Cf) \\
&+(Vx-Vref).\frac{((C_{p1p}+C_{p2p})-(C_{p1n}+C_{p2n}))}{Cf}.((2^n-1).Cu+Cf) \\
&-Vref.(C_{p1p}-C_{p1n})-Vx.(C_{p2p}-C_{p2n})
\end{aligned}
\tag{5.15}
$$

In order to simplify the exact transfer functions some assumptions should be made.

- $C_{p2p} \cong C_{p2n} \cong C_{p2}$ **&** $C_{p1p} \cong C_{p1n} \cong C_{p1}$

This is an assumption due to the symmetrical properties of the circuit. Special effort was made during the layout for this condition to hold. Another reason why this assumption is correct is that, these capacitances are small themselves, and their difference would be much smaller compared to the other capacitors in the circuit. As a result, this assumption leads to a reduced equation:

$$
\begin{aligned}
&-(Vinp-Vinn).\frac{C_S}{Cf}.\left((2^n-1).Cu+Cf\right)\\
&=(Vres2p-Vres2n).C_S\\
&+\left((Vres2p-Vres2n)-([DigCode2p]-[DigCode2n]).Vrefp\right).Bin.Cu\\
&+(Vres2p-Vres2n).Cf+(Vres2p-Vres2n).\left(C_{p1}+C_{p2}\right)\\
&-\left(([DigCode1p]-[DigCode1n]).Vrefp\right)Bin.\frac{Cu}{Cf}.\left((2^n-1).Cu+Cf\right)\\
&+\left([00000001].Vrefp\right).Bin.\frac{Cu}{Cf}.\left((2^n-1).Cu+Cf\right)
\end{aligned}
\qquad \textbf{(5.16)}
$$

- $Vres2p \cong Vres2n \cong Vres2$

This assumption is correct because the conversion algorithm itself tries to equate the voltage at these nodes. Thus, the equation is further reduced:

$$
\begin{aligned}
&-(Vinp-Vinn).\frac{C_S}{Cf}.\left((2^n-1).Cu+Cf\right)\\
&=\left(0-([DigCode2p]-[DigCode2n]).Vrefp\right).Bin.Cu\\
&-\left(([DigCode1p]-[DigCode1n]).Vrefp\right)Bin.\frac{Cu}{Cf}.\left((2^n-1).Cu+Cf\right)\\
&+\left([00000001].Vrefp\right).Bin.\frac{Cu}{Cf}.\left((2^n-1).Cu+Cf\right)
\end{aligned}
\qquad \textbf{(5.17)}
$$

$$
\begin{aligned}
&(Vinp-Vinn)\\
&=\left(([DigCode2p]-[DigCode2n]).Vrefp\right).Bin.\frac{Cu.Cf}{\left((2^n-1).Cu+Cf\right).Cs}\\
&+\left(([DigCode1p]-[DigCode1n]).Vrefp\right)Bin.\frac{Cu}{Cs}\\
&-\left([00000001].Vrefp\right).Bin.\frac{Cu}{Cs}
\end{aligned}
\qquad \textbf{(5.18)}
$$

$$
\begin{aligned}
&(Vinp-Vinn)\\
&=\begin{pmatrix}([DigCode2p]-[DigCode2n]).Bin\dfrac{Cf}{\left((2^n-1).Cu+Cf\right)}\\
+([DigCode1p]-[DigCode1n]).Bin-([00000001].Bin)\end{pmatrix}.Vrefp.\frac{Cu}{Cs}
\end{aligned}
\qquad \textbf{(5.19)}
$$

For a clearer point of view, another definition is made:

$$
\begin{aligned}
&[DigCode2]=([DigCode2p]-[DigCode2n])\\
&[DigCode1]=([DigCode1p]-[DigCode1n])
\end{aligned}
$$

It should be noted that exchanging 0’s of $[DigCode1p]$with -1’s would directly give $[DigCode1]$.

And equation (5.19) becomes:

$$(Vinp - Vinn) = \begin{pmatrix} [DigCode2].Bin \dfrac{Cf}{((2^n - 1).Cu + Cf)} \\ + [DigCode1].Bin - ([00000001].Bin) \end{pmatrix} .Vrefp.\frac{Cu}{Cs} \qquad \textbf{(5.20)}$$

It is clear that the transfer function has a systematic offset, yet it does not contribute the nonlinearity of the ADC. Moreover, the coefficient $\dfrac{Cf}{((2^n - 1).Cu + Cf)}$ determines the “weight” of the second 8-bit digital data. In the design Cf is chosen to be equal to $4 \times Cu$ 4. As a result, the value of the coefficient comes out to be $1/64.75$. This value will be used to get the actual 14 bits data, from the raw 16 bits of data. The operation will be done outside the chip, since the mismatch of Cf will cause a large nonlinearity. The actual value will be determined experimentally.

After that, the input range can be defined by assuming a maximum digital output is achieved. It should also be noted that the maximum value of the digital output will be:

$$[DigCode1] = [11111111]$$
$$[DigCode2] = [00111111]$$

The two most significant bits of the output of the second conversion will be used for digital error correction. If no errors occur during conversion, the two bits are expected to be 0. Placing the maximum digital values in equation (5.20):

$$(Vinp - Vinn) = \begin{pmatrix} [00111111].Bin \dfrac{1}{64.75} \\ + [11111111].Bin - ([00000001].Bin) \end{pmatrix} .Vrefp.\frac{Cu}{Cs}$$

$$(Vinp - Vinn) = \left((2^6 - 1)\frac{1}{64.75} + (2^8 - 1) - 1 \right).Vrefp.\frac{Cu}{Cs}$$

$$(Vinp - Vinn) \cong (1 + (2^8 - 1) - 1)Vrefp.\frac{Cu}{Cs}$$

$$(Vinp - Vinn) \cong (2^8 - 1)Vrefp.\frac{Cu}{Cs} = 255.Vrefp.\frac{Cu}{Cs}$$

For a maximum peek voltage of 16 V and a positive reference voltage of 2V, Cs is found to be:

$$Cs = 255.2.\frac{Cu}{16} = 31.875$$

Since the sampling capacitors will be exposed to high voltage stress, they cannot be realized by poly-poly capacitances, metal-metal capacitance is used instead. Since the matching of the metal-metal capacitance to poly-poly capacitance is not expected to be good, either the exact input signal range will be determined experimentally, or the positive reference voltage will be adjusted so that the input signal range is 16 V_{peak}. As a result, the input signal range is 32 V_{pp}.

6. EFFECTS OF NONIDEALITIES

In this section, non-ideality effects on the linearity of the SARADC will be investigated. The main non-idealities are: comparator and differential amplifier input offset voltage, element mismatch in the capacitor array, noise, and amplification error.

6.1 Comparator and Differential Amplifier Offset Voltage

As previously explained, the SARADC algorithm makes use of both amplifier and comparator. Thus both amplifier and comparator offsets have influence on the transfer function of the circuit. For the rest of the analysis, comparator and amplifier offsets will be represented with $Voff_{(c)}$ and $Voff_{(a)}$, respectively.

At the end of the first conversion the input output relation can be given with the following equation:

$$Vin = D1 * V_{LSB(DAC)} + res1 \tag{6.1}$$

where Vin is the analog input voltage, D1 is the first 8-bit data, $V_{LSB(DAC)}$ is least significant bit voltage of the capacitive DAC, and res1 is the residue voltage at the end of the first conversion.

After that, amplification phase comes, and residue is amplified and the amplified signal (Vamp) can be given as:

$$Vamp = K.\left(res1 - Voff_{(a)}\right) \tag{6.2}$$

where K is the amplification factor. This signal is resampled on the capacitive DAC. Then, second conversion takes place and the amplified signal is converted into 8-bit digital data D2 as in the equation:

$$Vamp = D2.V_{LSB(DAC)} + res2 \tag{6.3}$$

where D2 is the second 8 bit data. After two conversions the two 8-bits data are combined to get the actual digital output, D_{tot}, as in the following equation:

$$D_{tot} = D1 + \frac{D2}{K} \tag{6.4}$$

Replacing D1 and D2 in equation (6.4) with their equivalents from equation 6.1-3:

$$D_{tot}.V_{LSB(DAC)} = Vin - res1 + \frac{K.\left(res1 - Voff_{(a)}\right) - res2}{K}$$

After necessary simplifications, the transfer function of the SARADC is given by:

$$D_{tot}.V_{LSB(DAC)} = Vin + Voff_{(a)} - \frac{res2}{K} \tag{6.5}$$

From equation 6.5 it is obvious that the relation between the digital data and input voltage is linear. The effect of the offsets is to shift the transfer function. Res2/K is the quantization error, where K is 64. The comparator offset is included within the res2 term and causes only a shift, just as the amplifier offset does.

6.2 Mismatch at the Capacity Array

Mismatch of the elements of the capacitor array directly contribute to the non-linearity of the SARADC. There are three different factors contributing to the mismatch in the capacitor array: gradients, edge effects, and statistical scattering.

Gradient errors are the process parameter changes on the wafer. Especially insulator thickness change in any direction would deteriorate INL. Another problem about gradient error is, there is not a proper way to simulate and foresee its effect. A random walk scheme[19] was applied in placing capacitors to the layout. This technique does not reduce gradients, whereas it reduces the effect of gradient errors on the INL.

The non-continuous behavior at the boundaries of a homogeneous layout would cause edge-effects, which stems from mask accuracy and the edging accuracy. The capacitor array is such a homogeneous layout (Figure 3.16). The capacitor values at the boundaries could be different from the ones at the middle due to edge effects. In order to avoid that happen, the main capacitor matrix is encircled with a row of dummy unit capacitor cell. Just like gradient errors, the prediction of edge-effects by simulation is not possible.

The last contributor to the mismatch is the statistical scattering which is independent from the layout placement or routing. The magnitude of the mismatch is inversely proportional to the square root of the area of the capacitor and is given by the equation:

$$\sigma C / C = {}^{Ac}\!\big/_{\sqrt{W * L}} \qquad \textbf{(6.6)}$$

Where Ac is a parameter determined by the technology, W and L are the width and length of the capacitor, and σC/C is the ratio of the standard deviation to the nominal value of the capacitance. The technology used, provides the parameter Ac = 0.45%. The statistical scattering can be simulated with Monte Carlo analyses tool and statistical parameters provided by the technology. On the other hand, using Monte Carlo analysis for the entire system and obtaining DNL and INL is extremely time consuming. An easier way to model the SARADC for DNL and INL, is to measure the DNL and INL parameters of its capacitive DAC, because the linearity of the SARADC is determined by the capacitive DAC.

Assuming that the transfer function of the implemented capacitive DAC composes of a linear and an error element, equation 6.7 is achieved.

$$TF_{DAC} = [I] + [E] \qquad \textbf{(6.7)}$$

Where [I] is the binary weighted vertical vector, and [E] is the vertical error vector. It should be noted that, the linear element [I] is the ideal case (zero error) transfer function.

At the end of the first conversion, sampled input (Vin) is converted into an 8-bit digital data [D] and a residue signal (res) as given in equation 6.8.

$$Vin = [D].[I] + res \qquad \textbf{(6.8)}$$

For the non-ideal case, the eq.2 is converted into equation 6.9.

$$Vin = [D].([I] + [E]) + res_D \qquad \textbf{(6.9)}$$

Where res_D is the residue signal of the implemented DAC.

From equations 2 and 3, the result in equation 6.10 can be obtained.

$$res - res_D = [D].[E] \qquad \textbf{(6.10)}$$

This phenomenon is illustrated in **Figure 6.1**.

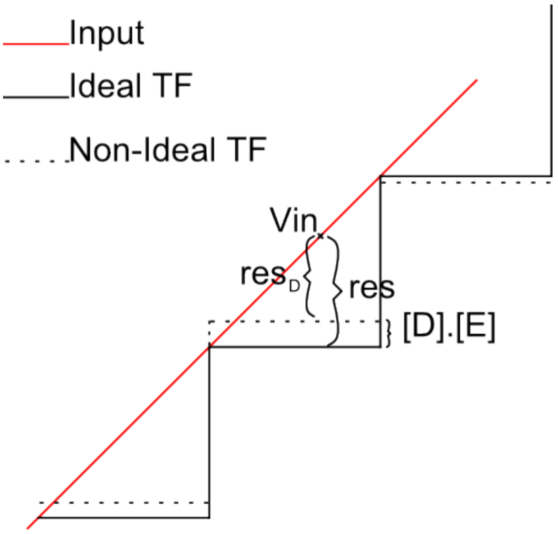


Figure 6.1: Ideal and non-ideal transfer functions.

After that the residue signal will be amplified by a factor of K (nominal value is 64 for K) to obtain a second 8 bit data. After these steps, the resultant error signal due to non-linear DAC is given in equation 6.11.

$$Err = K.[D].[E] \tag{6.11}$$

And the resultant analog-to-digital transfer function is:

$$Vin = K.[D] + [D2] + K.[D].[E] \tag{6.12}$$

Where [D2] is the ideal second 8 bit digital data.

It is required that the error signal is less than 1 LSB:

$$|K.[D].[E]| < 1LSB$$

$$|[D].[E]| < 1LSB / K \tag{6.13}$$

Where 1LSB is equal to 2^{-8}. Placing the value of K (2^6) into the equation it is concluded that the implemented DAC should be of 14 bits resolution.

In order to investigate if this condition holds for the proposed DAC, it is simulated separately. A digital ramp signal is applied to the DAC so that the input is increased by “1” each time. The analog output voltage change for each step is observed by subtracting the shifted analog signal from the original one. That is the observed signal is:

$$Y[n] = X[n] - X[n-1] \qquad \textbf{(6.14)}$$

where X[n] is the analog output signal for the nth digital input. 10 simulations were run to observe the statistical scattering in Monte Carlo analyses with the option “mismatch only” selected since process scattering would not contribute to the nonlinearity. The results of the simulation are given in Figure 6.2. DNL (in units of LSB) of the capacitive DAC can be deduced from the simulation results according to the following formula:

$$DNL[n] = \frac{Y[n] - \frac{\sum Y[n]}{n}}{\frac{\sum Y[n]}{n}} \qquad \textbf{(6.15)}$$

The mean value of Y[n] is observed to be around 3.5mV. In order to find the maximum DNL, maximum deviation from the mean value is measured to be 40 μV. Thus, the maximum DNL of the capacitive DAC is found to be:

$$DNL(\max) = \frac{40\mu V}{35mV} = 0.0115LSB_{(8)} \qquad \textbf{(6.16)}$$

where LSB$_{(8)}$ is the least significant bit unit for 8 bits. In order to convert this DNL(max) into 14 bits (the resolution of the DAC) the value should be multiplied by $2^{(14-8)}$, and the result is:

$$DNL(\max) = 0.0115 \times 2^6 = 0.736LSB_{(14)} \qquad \textbf{(6.17)}$$

That is the linearity of the implemented DAC is better than 14 bits.

In addition, the proposed SARADC is modeled in matlab to investigate DNL and INL characteristics. Matlab codes are provided in APPENDIX A.3. The model used the parameter provided by the AMS together with the capacitor sizes to simulate the capacitive DAC according to the formula given in equation 6.6. The simulation is run 10 times, and for each simulation, a different color is used to plot DNL and INL. DNL and INL plots are given in Figure 6.3 and Figure 6.4 respectively. The results approve that the DNL is less than 1 LSB (no missing codes) and INL is slightly over 1 LSB for each run.

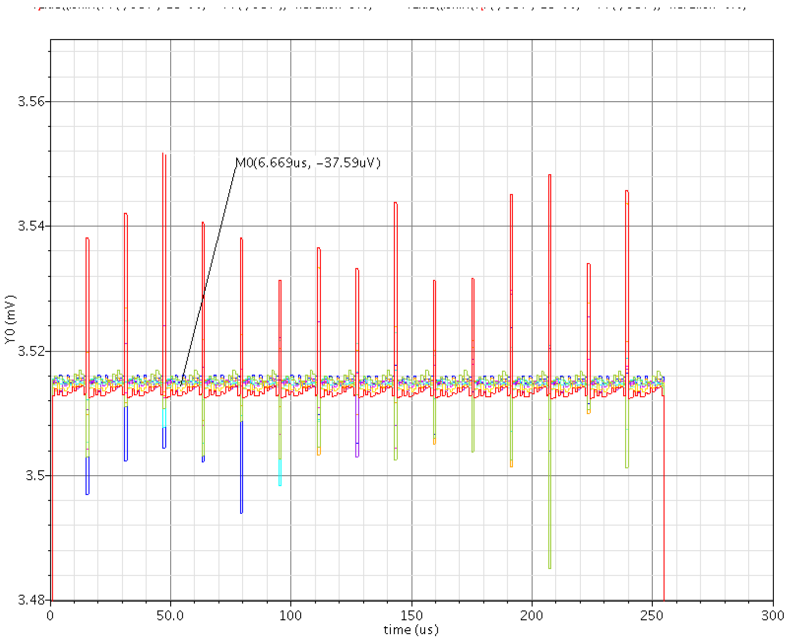


Figure 6.2: Monte Carlo analyses for capacitor array mismatch.

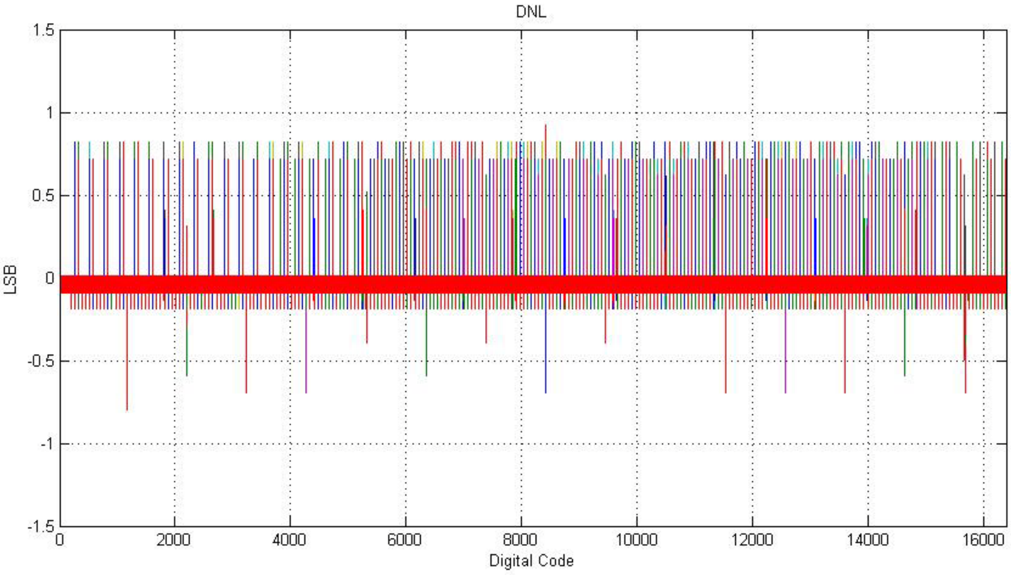


Figure 6.3: DNL plots for 10 runs of SARADC matlab model.

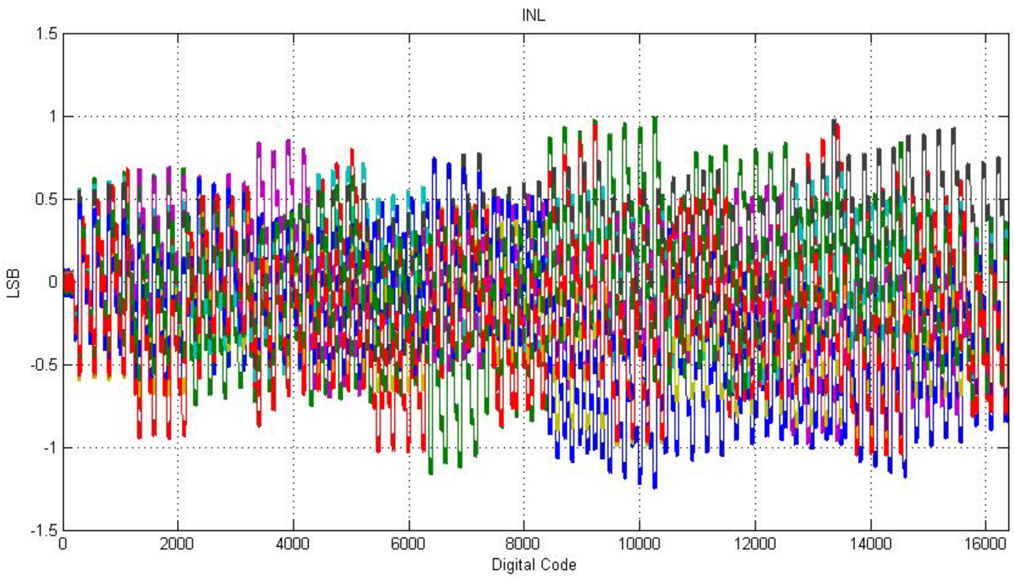


Figure 6.4: INL Plots for 10 runs of SARADC matlab model.

6.3 Noise

For noise analysis, the noise contribution of each state should be determined. First state is sampling state and the sampling noise (SN) can be expressed as:

$$SN^2 = Nsw^2 + Nref^2 \qquad \textbf{(6.18)}$$

Where Nsw is switch noise and Nref is the reference voltage noise. According to the algorithm, sampled noise is amplified and then converted to digital data. Thus, the contribution of the sampling noise is:

$$SNo^2 = K^2.SN^2 \qquad \textbf{(6.19)}$$

After sampling, first conversion takes place and first 8-bit digital data D1 is achieved. It should be noted that the comparator noise is cancelled by the digital error correction included in the algorithm, as long as comparator noise does not exceed 1 LSB of the first conversion, that is 15,6 mV ($4/2^8$ V). Such a high noise level is very unlikely to occur for the comparator used, so the assumption is easily satisfied.

After the first conversion, amplification phase comes. The noise sources at the amplification phase are switches, references, and amplifier itself. Noting that the switch and reference noises are multiplied by the amplification factor K, the output referred amplification noise (AN) can be expressed as:

$$AN^2 = (K.Nsw)^2 + (K.Nref)^2 + NampOR_{(AC)}{}^2 \quad \textbf{(6.20)}$$

where NampOR$_{(AC)}$ is the ouput referred amplifier noise for amplification configuration.

Then, resampling phase is used to resample the amplified signal. The resampling noise (RN) can be formulated as:

$$RN^2 = NampOR_{(UGC)}{}^2 + Nsw^2 \quad \textbf{(6.21)}$$

Where NampOR$_{(UGC)}$ is the output referred amplifier noise for the unity gain configuration.

Finally, second conversion generates the second 8-bit digital data to finish the conversion. The noise contribution of this state (C2N) is:

$$C2N^2 = Ncomp^2 + Nsw^2 + Nref^2 \quad \textbf{(6.22)}$$

where Ncomp is the comparator noise.

Total noise generated through the whole algorithm (Ntot) can be found by adding the contribution from all states:

$$Ntot^2 = SNo^2 + AN^2 + RN^2 + C2N^2 \quad \textbf{(6.23)}$$

$$\begin{aligned} Ntot^2 &= K^2.(Nsw^2 + Nref^2) + (K.Nsw)^2 + (K.Nref)^2 + NampOR_{(AC)}{}^2 \\ &+ NampOR_{(UGC)}{}^2 + Nsw^2 + Ncomp^2 + Nsw^2 + Nref^2 \end{aligned} \quad \textbf{(6.24)}$$

$$\begin{aligned} Ntot^2 &= Nsw^2(2.K^2 + 2) + Nref^2(2.K^2 + 1) \\ &+ NampOR_{(AC)}{}^2 + NampOR_{(UGC)}{}^2 + Ncomp^2 \end{aligned} \quad \textbf{(6.25)}$$

If the noise of each source is found the total noise power can be found.

The first noise source is switch noise. The switch noise for is defined as:

$$SN^2 = \frac{k.T}{C} \quad \textbf{(6.26)}$$

Where k is Boltzman constant and is equal to 1,38x10^{-23}J/K. T is the absolute temperature and can be accepted as 300 K (room temperature) for the analysis. And C is the sampling capacity. In this case, total sampling capacity (Ctot) is equal to the sum of sampling capacity (Cs) and equivalent capacity of the DAC looking from the output node (Ceq$_{(DAC)}$).

$$Ctot = Cs + Ceq_{(DAC)} \tag{6.27}$$

$$Ctot \cong 1pF + 8pF = 9pF \tag{6.28}$$

Applying the parameter values to the equation 6.26, the switch noise can be found:

$$SN^2 = \frac{1{,}38x10^{-23}x300}{9x10^{-12}}.V^2 = 4{,}6x10^{-10}V^2 \tag{6.29}$$

The second noise source is the reference voltage source. The noise from this source is determined externally from out the chip and can not be calculated. Thus this variable is left in the equation as it is.

The third noise contributor is the amplifier noise referred to the output nodes. This value is found from the simulations. The simulation is conducted with amplification configuration and the noise at the output nodes are is integrated from 10 Hz to 1GHz. The resulting noise is:

$$NampOR_{(AC)}{}^2 = 172x10^{-6}V^2 \tag{6.30}$$

The same simulation is run for unity gain configuration and the result is:

$$NampOR_{(UGC)}{}^2 = 4.8x10^{-6}V^2 \tag{6.31}$$

Note that the amplification configuration of the amplifier introduces a much higher noise than unity gain configuration, as expected. This is due to the high gain (64 nominally).

Since the comparator used is dynamic, the noise analysis of the comparator is quite complicated and will not be covered in this study. On the other hand, the value of this noise is expected to be a value much less than $NampOR_{(AC)}$. Thus, it can be neglected.

Integrating the values given in equations 29-31 into equation 6.25 the total noise can be found:

$$\begin{aligned}Ntot^2 &= 4.6x10^{-10}\left(2.64^2+2\right)V^2 + Nref^2\left(2.64^2+1\right)\\ &+172x10^{-6}V^2 + 4.8x10^{-6}V^2 + Ncomp^2\end{aligned} \tag{6.32}$$

$$\begin{aligned}Ntot^2 &= 3.77x10^{-6}V^2 + Nref^2\left(2.64^2+1\right)\\ &+172x10^{-6}V^2 + 4.8x10^{-6}V^2 + Ncomp^2\end{aligned} \tag{6.33}$$

Assuming Ncomp is negligible and Nref is zero, the output noise is

$$Ntot^2 = 3.77x10^{-6}V^2 + 172x10^{-6}V^2 + 4.8x10^{-6}V^2 = 180.57x10^{-6}V^2 \quad \textbf{(6.34)}$$

$$Ntot = 13.44mV \quad \textbf{(6.35)}$$

The total noise should be below 1 LSB (15.6 mV), which is the case. Yet, Ntot is actually equal to one σ of the total noise. Thus, when a signal is sampled many times, larger noise values are more likely to occur statistically and may deteriorate the performance of the SARADC. Moreover, reference voltage noise must be quite less than amplifier noise at amplification configuration not to increase noise level any more.

In order to reflect these numbers to the input stage both LSB and noise levels should be divided by 64. That is 1 LSB is equal to 243.75 μV and 1 sigma noise is equal to 210μV. As a result the noise contribution of the reference voltage should be less than the 243.75-210=33.75 μV.

6.4 Amplification Factor Error

When composing the digital data from the results of two 8-bit conversions, it should be taken into consideration that, the mismatch between the feedback capacitor and capacitive DAC would cause deviations from the nominal amplification factor 64. As previously mentioned, the total digital data (Dtot) is obtained from the digital data of first and second conversions (D1 and D2) as follows:

$$Dtot = K.D1 + D2 \quad \textbf{(6.36)}$$

In order not to “misinterpret” the raw digital data (16 bits), the actual amplification factor (K) should be found precisely.

The effect of the amplification factor error is illustrated in Figure 6.5. The black steps show the transfer function of the first digital data D1, and red steps show the addition of second digital data (D2) to generate the transfer function of the SARADC. The blue line is the ideal transfer function.

When digital data is calculated using actual amplification factor, the transfer function of the SARADC is equal to the ideal transfer function. Otherwise, the deviations from the ideal transfer function (err1 and err2 in the figure) occur.

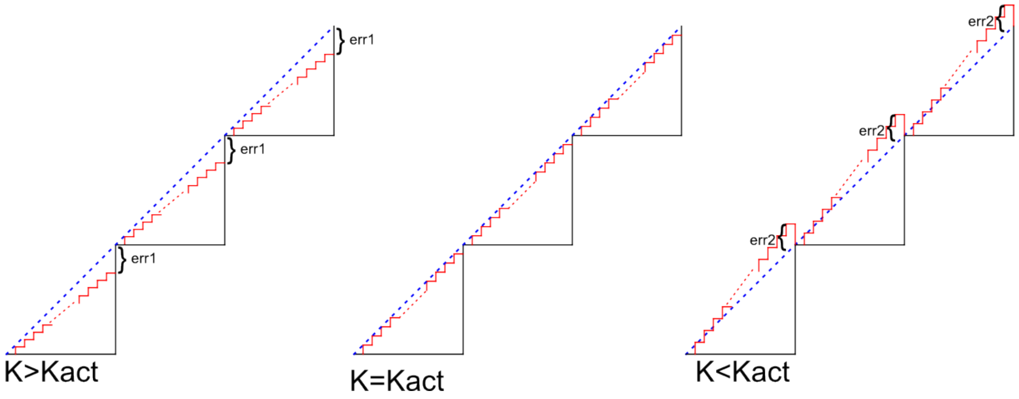


Figure 6.5: Amplification factor error affect on transfer function.

When testing, the amplification factor should be found experimentally. The tolerable error for the value of K should be found so that no missing codes occur and monotonicity is satisfied.

It is obvious from Figure 6.5 that the errors occur at the transition of:

$$K_{id}.D1 + D2_{\max} \Rightarrow K_{id}.(D1+1) \quad \textbf{(6.37)}$$

where K_{id} is the ideal value of amplification factor K.

$$K_{id}.D1 + D2_{\max} + 1 = K_{id}.(D1+1) \quad \textbf{(6.38)}$$

removing K_{id}.D1 from both sides:

$$D2_{\max} + 1 = K_{id} \quad \textbf{(6.39)}$$

The error is then equal to

$$Error = K - D2_{\max} - 1 \quad \textbf{(6.40)}$$

Boundary condition for the error is 1 LSB can be expressed as

$$-1 < Error = K - D2_{\max} - 1 < 1 \quad \textbf{(6.41)}$$

Placing equation 6.39 into equation 6.41

$$-1 < K - K_{id} < 1 \quad \textbf{(6.42)}$$

So the boundary condition for the amplification factor K is:

$$K_{id} - 1 < K < K_{id} + 1 \quad \textbf{(6.43)}$$

That is, for the proposed SARADC, if the capacitors determining the value of amplification factor K are matched better than 6-bits, monotonicity and no missing codes is guaranteed.

7. MEASUREMENTS

In this section, the measurements done so far and to be done are going to be presented. First subsection describes the measurements conducted on the previously implemented switched capacitor attenuator (SC attenuator). Second subsection describes the test procedure for the proposed SARADC.

7.1 SCATTENUATOR

Before implementing the whole SARADC, a switched capacitor attenuator is designed in order to investigate the reliability and performance of the proposed high voltage sampling bootstrapped switch experimentally. The SC Attenuator is implemented by using the high voltage sampling switch presented in section 0 and operational amplifier presented in section 3.2. The block diagram of the SC attenuator is given in Figure 7.1.

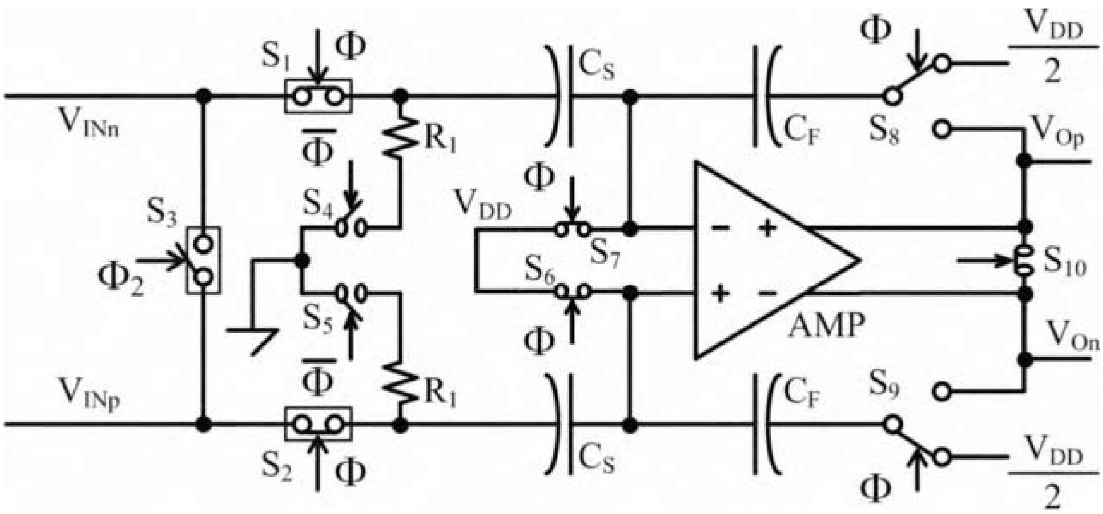


Figure 7.1: Switched capacitor attenuator schematic.

In the figure, the high voltage sampling bootstrapped switches S1-3 are enclosed with a rectangle around them. S1 and S2 are used to sample differential ends of the input signal while S3 is included in the system for stand-alone characterization. The switches S4 and S5 are DNMOS transistors for ground connection. The resistors are placed in series configuration to limit the pull-down speed. The remaining switches are simple transmission gates implemented with 3.3V CMOS devices, because the nodes they are connected to do not experience high voltage stress.

The sampling capacitors, i.e. C_{SP} and C_{SN}, are implemented using metal-insulator-metal (MIM) capacitors and they can withstand to 65V. The feedback capacitors, i.e. C_{FP} and C_{FN}, are implemented using 5V poly capacitors and the attenuation factor, i.e. C_F / C_S is set to be 8.

Φ is the control signal of the SC Attenuator.

If charge conservation equations are written down for the steady states of sample and hold, it is easy to deduce the transfer function of the switched capacitor attenuator:

When Φ is equal to supply voltage V_{DD}, the switches S1, S2, S5, S6, and S10 are on; and switches S8 and S9 connect the bottom plates of the capacitors to half the supply voltage $V_{DD}/2$, just as it is in the figure. In that case the charge on the shorted input terminals of the opamp is:

$$\begin{aligned} &Qtotal = \\ &(V_{DD} - V_{INn}) \times C_{SP} + (V_{DD} - V_{INp}) \times C_{SN} \\ &+ \left(V_{DD} - {}^{V_{DD}}\!/_{2}\right) \times C_{FP} + \left(V_{DD} - {}^{V_{DD}}\!/_{2}\right) \times C_{FN} \end{aligned} \quad \textbf{(7.1)}$$

When Φ goes to 0, for hold state, the switches S1, S2, S5, S6, and S10 are off, and S4 and S5 are on to transfer the charges on sampling capacitor on feedback capacitors. In that state, the charges on the input terminals are:

$$Qp = Vx \times C_{SP} + (Vx - V_{Op}) \times C_{FP} \quad \textbf{(7.2)}$$

$$Qn = Vx \times C_{SN} + (Vx - V_{On}) \times C_{FN} \quad \textbf{(7.3)}$$

where Vx is the final voltage on the input terminals of the opamp. By the charge conservation rule, the sum of the charges given by equations (7.2) and (7.3) must be equal to the initial charge:

$$Qp + Qn = Qtotal \quad \textbf{(7.4)}$$

On the other hand, it is known that

$$\begin{aligned} C_{SP} &= C_{SN} = C_S \\ C_{FP} &= C_{FN} = C_F \end{aligned} \quad \textbf{(7.5)}$$

Moreover, Input and output voltages can be defined in terms of a common voltage and a differential voltage:

$$\begin{aligned} V_{INn} &= V_{INCM} - V_{IN}/2 \\ V_{INp} &= V_{INCM} + V_{IN}/2 \\ V_{On} &= V_{OCM} - V_O/2 \\ V_{Op} &= V_{OCM} + V_O/2 \end{aligned} \qquad \textbf{(7.6)}$$

Placing the equations (7.1-7.3) on (7.4) with the newly defined variables on (7.5) and (7.6):

$$\begin{aligned} & Vx \times C_S + (Vx - V_{Op}) \times C_F + Vx \times C_S + (Vx - V_{On}) \times C_F \\ & = (V_{DD} - V_{INn}) \times C_S + (V_{DD} - V_{INp}) \times C_S + \left(V_{DD} - V_{DD}/2\right) \times C_F + \left(V_{DD} - V_{DD}/2\right) \times C_F \end{aligned}$$

$$\Rightarrow 2Vx \times C_S + (2Vx - V_{Op} - V_{On}) \times C_F = (2V_{DD} - V_{INn} - V_{INp}) \times C_S + V_{DD} \times C_F$$

$$\Rightarrow 2Vx \times C_S + (2Vx - 2V_{OCM} - V_O) \times C_F = (2V_{DD} - 2V_{INCM} - V_{IN}) \times C_S + V_{DD} \times C_F$$

$$\begin{aligned} & \Rightarrow 2Vx(C_S + C_F) - 2V_{OCM} \times C_F - V_O \times C_F \\ & = -2V_{INCM} \times C_S - V_{IN} \times C_S + V_{DD}(C_F + 2C_S) \end{aligned} \qquad \textbf{(7.7)}$$

In equation (7.7), there are two orthogonal variables: DC variables and AC variables, so that the equation can be separated into 2 equations:

$$-V_O \times C_F = -V_{IN} \times C_S \qquad \textbf{(7.8)}$$

$$\Rightarrow 2Vx(C_S + C_F) - 2V_{OCM} \times C_F = -2V_{INCM} \times C_S + V_{DD}(C_F + 2C_S) \qquad \textbf{(7.9)}$$

Where equation (7.9) defines the relation of the DC voltages; and equation (7.8) defines the relation of the AC input and output voltages. Thus, the attenuation factor is given as:

$$V_O = V_{IN} \times C_S / C_F \qquad \textbf{(7.10)}$$

The linearity simulation results of the implemented SC attenuator are given in Figure 7.2. The output signal is 4Vpp obtained after attenuating 32Vp p input signal 8 fold. The SC attenuator simulations show 107dB linearity for 1 MHz clock and 7 kHz 32Vpp differential input. As it is clear, the even order harmonics are canceled by the differential circuit topology. The linearity stays above 100 dB for 100 kHz input signal.

The linearity errors originating from the high-voltage switch channel charge injection are eliminated using top plate sampling technique. To that end, the top plate switches, i.e. S6 and S7, are turned off slightly before the high voltage switches S1 and S2. The linearity errors originating from the switch transistor gate overdrive modulation is minimized by choosing large flying capacitor value.

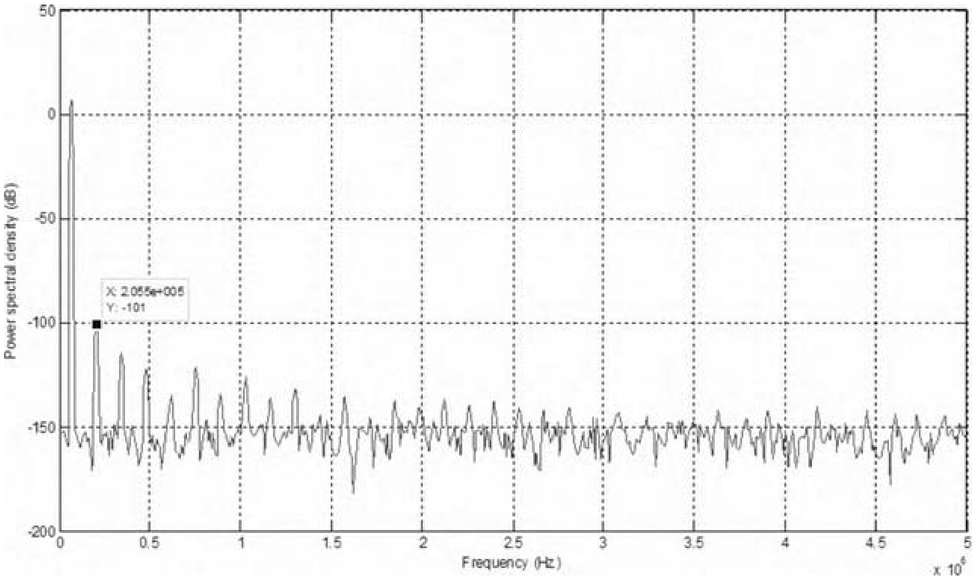


Figure 7.2: FFT results of SC attenuator simulation.

The die micrograph of the implemented SCA is given in Figure 7.3. In the figure the three switches S1-S3, the amplifier AMP, sampling capacitors CS, and feedback capacitors CF are given together with the silicon area they consume. The rest of the circuit is the biasing circuitry.

Figure 7.4 shows the measurement results of S3 as a track and hold switch. The switch is loaded with the oscilloscope probe and sampled 20Vpp signal on $15V_{DC}$ reliably with 2.2V supply voltage without forward biasing any diode. Achieved maximum input signal to supply voltage ratio is 11. The clock frequency is 1 MHz. Although safe operating input signal range is far above than the achieved 25V maximum input signal level, the sampling switch becomes very slow above 25V due to the lack of switch transistor gate overdrive voltage for high input levels. In order to increase this level further, it is necessary to increase the value of the flying capacitor C3 given in Figure 3.1.

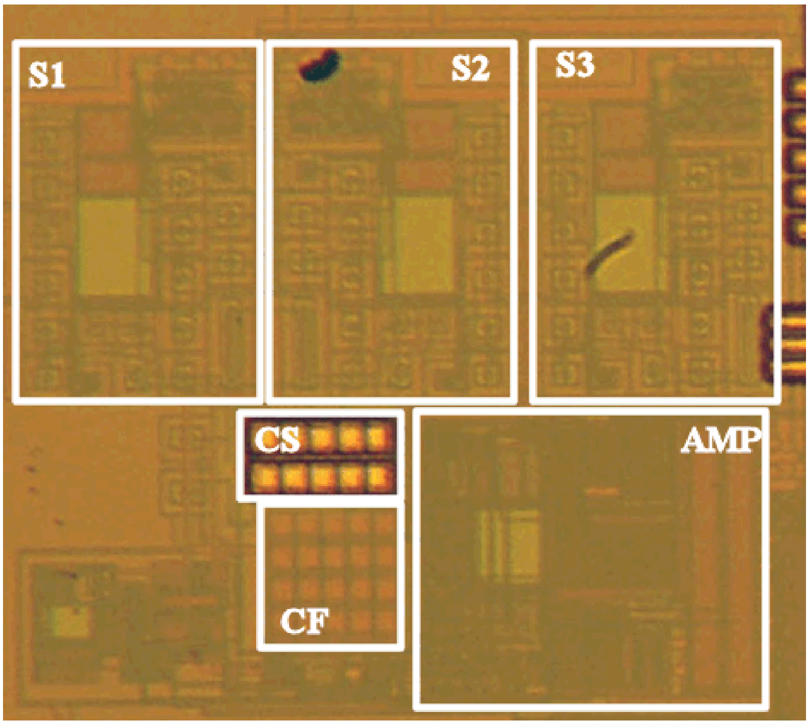


Figure 7.3: Die micrograph of SCA.

Figure 7.5 shows the measurement results of the implemented switched capacitor attenuator. Only positive output terminal voltage of the amplifier and the positive input signal are given in the figure. The differential input signal is 1 kHz 20Vpp on 10VDC. The supply voltage is 2.2V.

In order to test the reliability of the switch, the stand-alone high voltage sampling switch is left for continuous sampling of 25VDC. The output node of the switch is reset to ground after each sampling. The switch kept operating normally after 5 days of continuous high-voltage sampling. No significant current flow measured from the input terminal throughout the measurements proving that none of the parasitic diode is turned on.

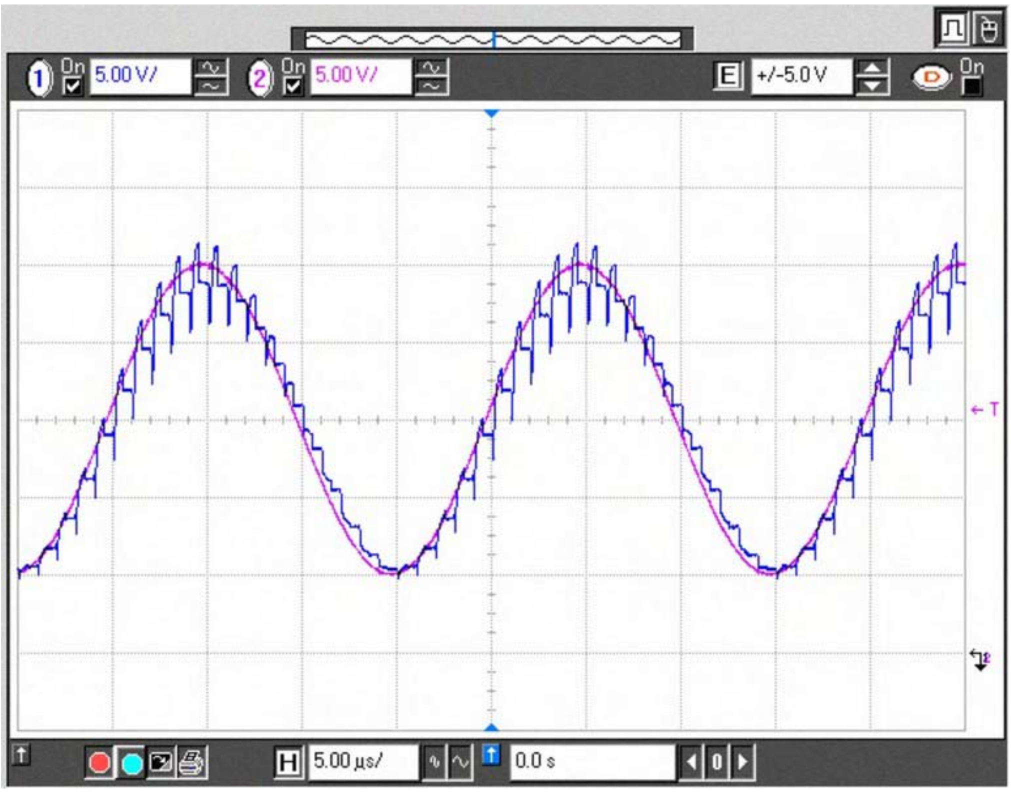
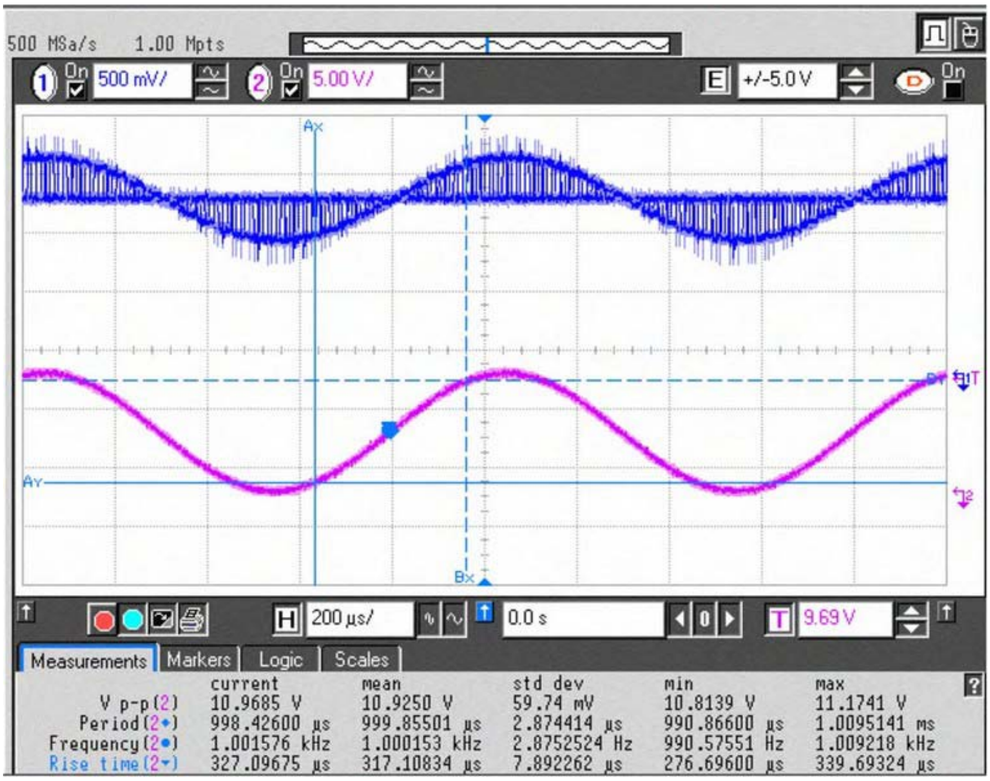


Figure 7.4: Measurement of high-voltage sampling switch (S3) in track and hold configuration. Input 20 Vpp + 15 V_{DC}, clock 1 MHz.



```
                    current          mean             std dev          min              max
       V p-p(2)     10.9685 V        10.9250 V        59.74 mV         10.8139 V        11.1741 V
     Period(2•)     998.42600 µs     999.85501 µs     2.874414 µs      990.86600 µs     1.0095141 ms
  Frequency(2•)     1.001576 kHz     1.000153 kHz     2.8752524 Hz     990.57551 Hz     1.009218 kHz
  Rise time(2•)     327.09675 µs     317.10834 µs     7.892262 µs      276.69600 µs     339.69324 µs
```

Figure 7.5: Measurement result of SC attenuator showing V_{INP} and V_{OP} terminal voltages, test condition 20Vpp differential input on $10V_{DC}$.

7.2 SARADC Simulation Results

Implemented SARADC was simulated to find the spurious free dynamic range (SFDR) performance. An input signal of 32 Vpp which satisfies coherent sampling conditions was sampled at 1 MSps by the proposed SARADC. A transient simulation is run. 16-bit digital data is collected and exported into matlab. The raw 16-bit digital data is converted into the “actual” 14-bit digital data as explained in section 5. Then, FFT method is used to find the spectrum of the output signal. The results of the FFT analysis is given for input frequencies of 15.6 KHz and 474.6 KHz in Figure 7.6 and Figure 7.7. The SFDR is performance is 95 dB and 86 dB, respectively.

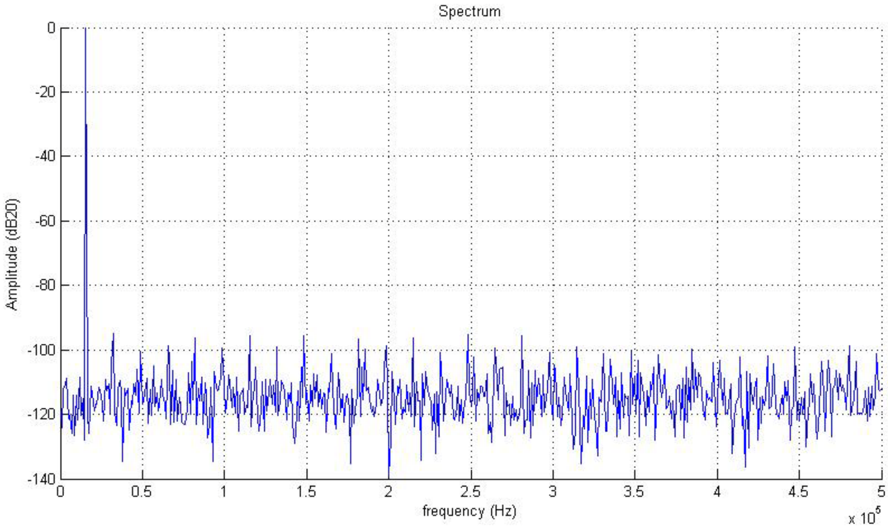


Figure 7.6: FFT analysis of SARADC top level simulation Fin = 15.6KHz.

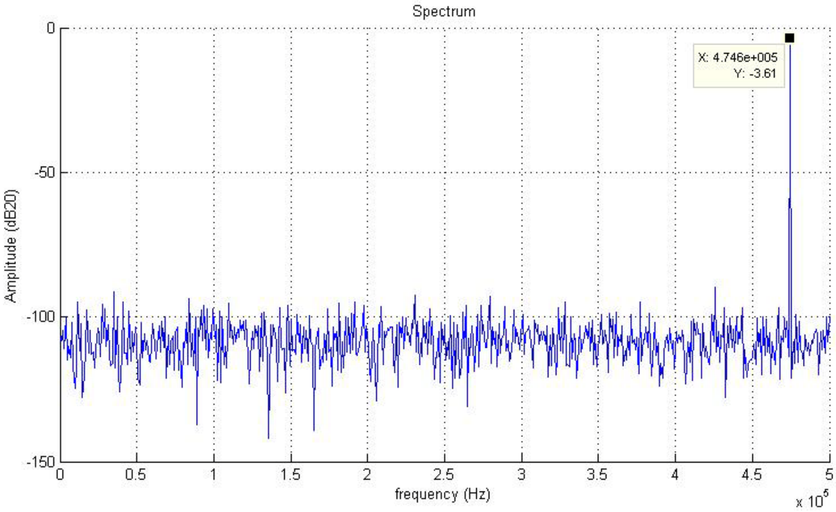


Figure 7.7: FFT analysis of SARADC top level simulation Fin = 475 KHz.

7.3 SARADC Tests

A test procedure to measure the performance of the proposed SARADC has been prepared. In order to ease the understanding of the following test procedure the top level symbol is presented in Figure 7.8 with all the pins chip includes. Besides, the top level layout and bonding diagram of the chip are given in APPENDIX A.1 (Figure A.9 and Figure A.10, respectively). The pins are explained in the Table 5 in detail.

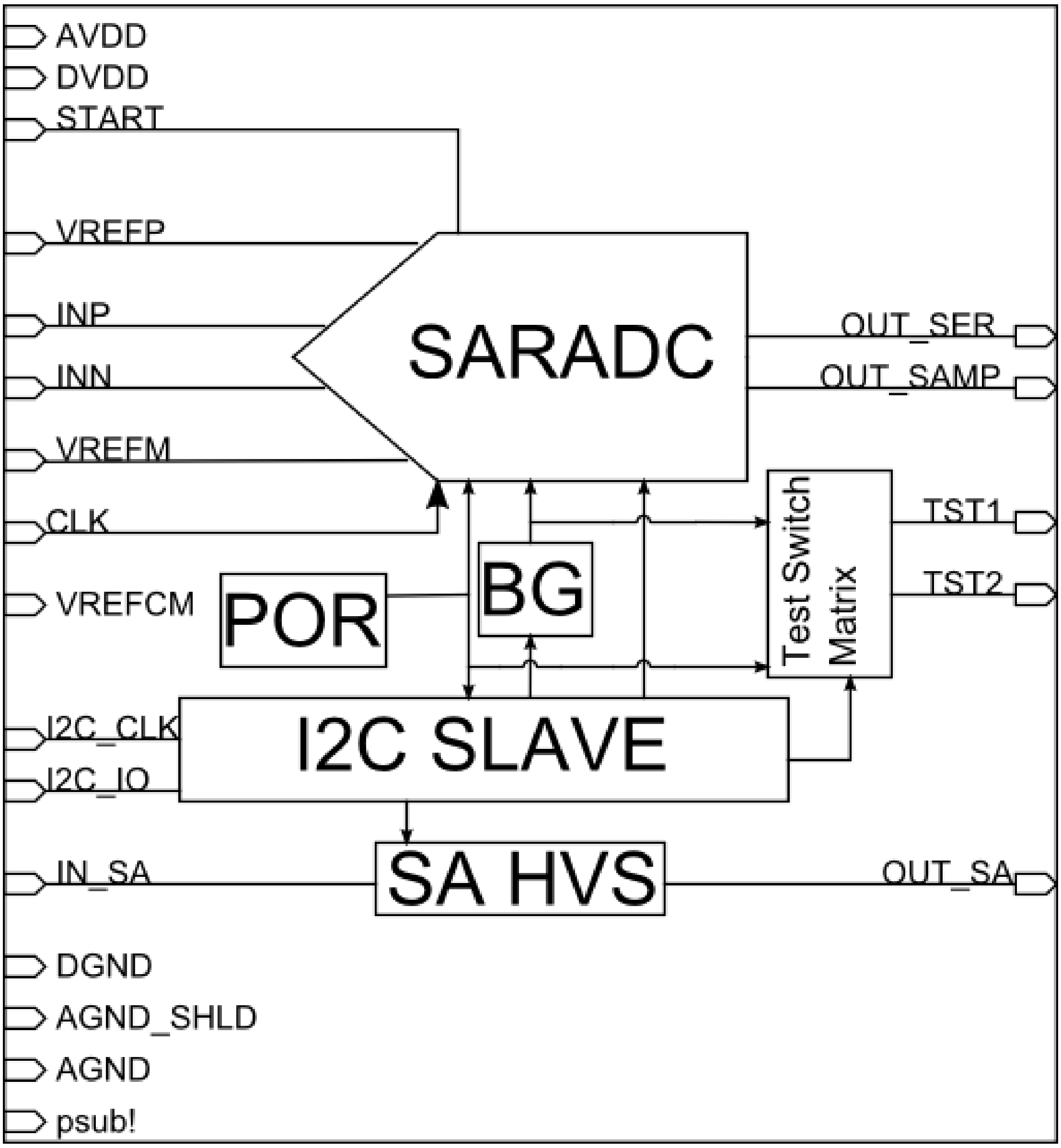


Figure 7.8: Top level symbol for test chip.

Table 5: Pin explanation of the test chip.

Pin Name	Explanation	Pin Number	Voltage Range	Selected IO
AVDD	Analog Supply	40	3.3	AVDD3ALLP_HV
AGND	Analog Ground	37	0	AGND3ALLP_HV
AGND_SHLD	Analog Shield	38	0	AGND3ALLP_HV
psub!	Substrate	39		AVSUB3ALLP_HV
DVDD	Digital Supply	48	3.3	AVDD3ALLP_HV
DGND	Digital Ground	47	0	AGND3ALLP_HV
VREFP	Positive Reference	34	2-3	APRIO50P_HV
VREFM	Negative Reference	33	0	APRIO50P_HV
VREFCM	Top plate reference at Sampling	32	0-3.3	APRIO50P_HV
INP	Positive Analog Input	25	0-16	POWER20
INN	Negative Analog Input	24	0-16	POWER20
IN_SA	Stand Alone HVS Input	9	0-16	POWER20
OUT_SA	Stand Alone HVS Output	8	0-16	POWER20
CLK	Clock	46	0-3.3	ICCK2P_HV
I2C_CLK	I2C clock	45	0-3.3	ISP_HV
I2C_IO	I2C input-output	44	0-3.3	BBS24SMP_HV
START	Start Conversion	41	0-3.3	ISP_HV
OUT_SER	Digital Serial Output	43	0-3.3	BU4SMP_HV
OUT_SAMP	Output Sample	42	0-3.3	BU4SMP_HV
TST1	Test Pin 1	35	0-3.3	APRIO50P_HV
TST2	Test Pin 1	36	0-3.3	APRIO50P_HV
HVGND	Stand Alone HVS ground	5-21-28	0	POWER20

The tests to be conducted include main performance criteria of ADCs such as SFDR, SNR, THD, DNL, INL, power consumption, etc. Besides, in order to separate any possible performance limiting block, an I2C slave is integrated into the chip. This block enables the observation of internal signals from outside the chip. Moreover, performance optimization and reference voltage and current trimming are done via the I2C interface. The top level test setup and control signals are given in Figure 7.9.

The blocks composing the SARADC are encircled with a dotted rectangle. SARADC I2C is the module that applies the given test and control signals to the relevant block. The Switch Matrix block is controlled by the I2C slave and connects the internal signals to the two test pins, i.e. TST1 and TST2. A Stand Alone High Voltage Switch can be enabled if the proposed high voltage switch's performance is to be measured.

It has its own input and output pins, i.e. IN_SA and OUT_SA. The bus width and default value of the control signals applied via I2C are given in Table 6.

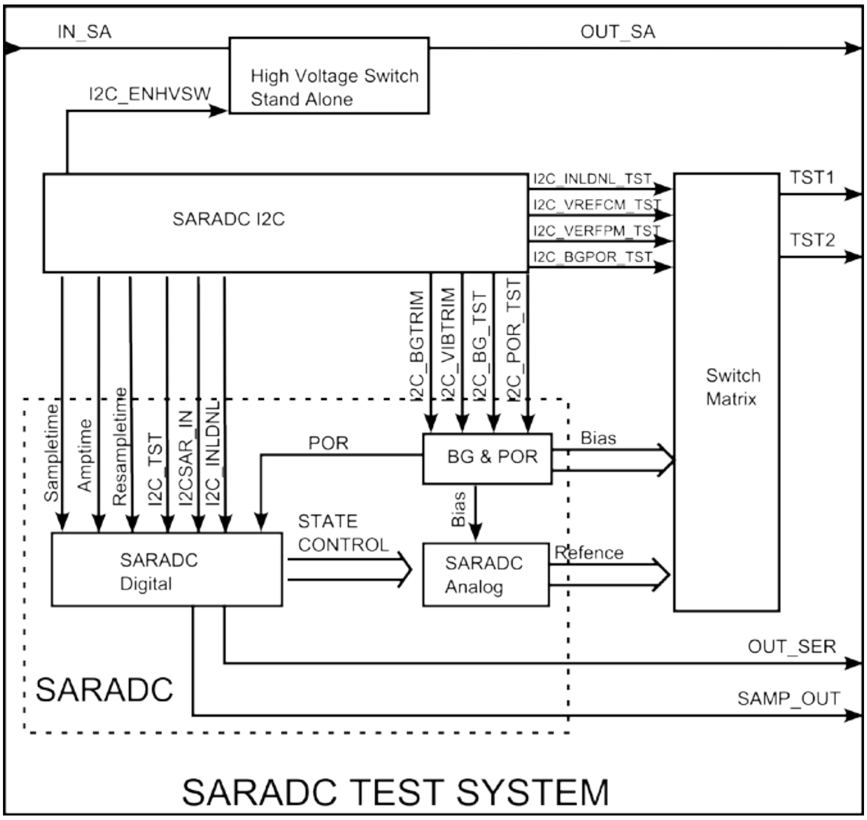


Figure 7.9: Top level test control signal.

Table 6: I2C controlled test signals.

Signal Name	Bus Width	Default Value	Explanation
Sampletime	6	000110	Sets sampling period length
Amptime	6	001000	Sets amplification period length
Resampletime	6	000110	Sets resampling period length
I2C_TST	6	000000	Connects two of the state control signals to either pins, TST1 and TST2
I2C_SAR_IN	8	00000000	Digital Input for measuring resolution of capacitive DAC
I2C_INLDNL	1	0	Enables measuring resolution of capacitive DAC
I2C_INLDNL_TST	1	0	Enables measuring resolution of capacitive DAC
I2C_VREFCM_TST	1	0	Connects reference signals VREFCM and VCMo to pins TST1 and TST2
I2C_VREFPM_TST	1	0	Connects reference signals VREFP and VREFM to pins TST1 and TST2
I2C_BGPOR_TST	1	0	Connects outputs of bandgap and POR to pins TST1 and TST2
I2C_POR_TST	1	0	Enables POR test
I2C_BG_TST	3	000	Determines either the voltage or current references are to be measured
I2C_BGTRIM	4	1000	Bandgap trim
I2C_VIBTRIM	6	100000	Reference current trim
I2C_ENHVSW	1	0	Enables Stand Alone High Voltage Switch
I2C_EN_ANA	1	1	Enables Differential Operational Amplifier
TOTAL	53		

7.3.1 Spurious free dynamic range and total harmonic distortion (SFDR and THD)

In order to measure SFDR and THD values the I2C controlled signals are set to their nominal values. A differential input signal of 32 Vpp sine wave is applied with a clock input of 50 MHz. The linearity of the applied input signal should be well above 110 dB. Else the linearity of the SARADC can not be tested properly, because the non-linearity of the SARADC itself can not be distinguished from the non-linear input itself. After applying proper input signals, the generated digital data is collected via Logic Analyzer. The PCB designed to test the proposed SARADC has a complex programmable logic device (CPLD) at its serial output to generate 16-bits of parallel data. As a result the output sampling rate is reduced from the clock rate of 50 MSps (nominal) to the actual sampling rate of 1 MSps; and the handling of the digital data is easier via Logic Analyzer. Discrete Fourier Transform method is used to analyze the output signal in frequency domain. The test results should be compared with the simulation results. If they match, the tests can be diversified for different input frequencies and different ambient temperatures. If the measured performance is below expected, each of the Sampletime, Amptime and Resampletime parameters should be adjusted to increase sampling, amplification and resampling periods.

7.3.2 Differential and integral non-linearity (DNL and INL)

Just like measuring SFDR and THD, a sine wave at nominal input conditions is applied and the digital data is collected via Logic Analyzer. The histogram of the digital data is obtained and statistical distribution of the codes is analyzed [18]. The number of hits of each code is compared with the mathematically expected value to determine the deviation. As a result, DNL and INL values are achieved.

7.3.3 Signal to noise ratio (SNR)

In order to measure SFDR and THD values the I2C controlled signals are set to their nominal values. A low noise DC input is applied and sampling is started. The digital data is collected via Logic Analyzer. The statistical distribution of the collected digital data is calculated by the histogram method to determine the noise of the designed SARADC. If no distribution is observed, it means that the noise floor is below 1 LSB. In that case, the voltage difference of the externally applied reference signals should be decreased. Decreasing the reference voltage reduces the input signal range covered by 14 bits. Thus, the step of 1 LSB reduces as well and sub-LSB noise can be measured.

7.3.4 Matching of the capacitive DAC

In order to determine the resolution of the capacitive DAC, the operation of the SARADC is reconfigured by the I2C block. During this test, analog inputs are not used and can be connected directly to ground. However, the clock signal should be applied throughout the test, yet the frequency can be reduced to a few KHz, since this is a DC measurement. To activate this mode, I2C_INLDNL and I2C_INLDNL_TST controls should be set to 1 via I2C. When this mode is activated, the digital input of I2C_SAR_IN is sampled on the capacitive array. After that, the top plates of the differential capacitive arrays are left connected to the inputs of the differential amplifier and their bottom plates are connected to the output nodes of the differential amplifier, constituting a unity gain negative feedback. As a result, an analog signal is observed at the output nodes of the differential amplifier, that is the applied digital signal is converted to an analog signal. In this measurement mode, the output nodes of the differential amplifier are connected to TST1 and TST2 pins of the system to measure the analog value. The transfer function of this conversion is equal to the transfer function of the pseudo-differential capacitive DAC. Thus, the transfer function of the implemented capacitive DAC can be determined by applying all possible 8-bit digital inputs manually and sampling the respective differential analog output data from outside the chip.

The resolution of the expected transfer function is above 14 bits. If the transfer function deviates from the expected results, the linearity of the SARADC is limited by the capacitive DAC.

7.3.5 Power consumption

The total power consumption of the SARADC can be calculated by measuring the total ground current with an ammeter at nominal operating conditions. Moreover, the total power consumption of the differential amplifier, bandgap reference circuit, and POR can be calculated by measuring the ground current of the SARADC, with clock signal disabled. After that if I2C_ANA_EN is set to 0, the ground current of the amplifier is cut off, and the resultant ground current is that of bandgap and reference. Subtracting this value from the previously measured one, the total power consumption of the differential amplifier can be measured.

7.3.6 Bandgap reference voltage and reference current

The output reference voltage of the bandgap circuit is directly connected to the output pin TST1 by setting I2C_BGPOR_TST to 1 and I2C_BG_TST to 010. The I2C_BGTRIM should be used to trim the bandgap reference circuit until 1.2V is observed at the output node. Moreover, the reference current generated by the bandgap reference circuit can be observed at the output by setting I2C_BGPOR_TST to 1 and I2C_BG_TST to 001. The reference current is sourced from the supply voltage (not sinked to ground) and should be handled accordingly. The reference current should be trimmed using I2C_VIBTRIM until 5 μA is achieved.

7.3.7 Power on reset operation

The operation of the Power on Reset circuit can be investigated as well. Setting both I2C_POR_TST and I2C_BGPOR_TST to 1, connects the POR output to the output pin TST2. In order to measure the turn off voltage of the POR, the analog supply voltage AVDD should be decreased slowly starting from 3.3. At the turn off voltage, the output of the POR disables the system. The turn off voltage sets a lower limit to the minimum supply voltage needed. Although, due to the hysteresis characteristic of the POR, a higher turn on voltage is needed.

7.3.8 Integrity of the reference signals inside the chip

If the reference signals are disturbed by factors such as clock and other digital signals, the performance of the SARADC is reduced. Although the analog reference signals are shielded and handled with great care on the layout, an unexpected disturbance may still affect the integrity of the signals. In order to detect such a disturbance, the reference signals are connected to output pins TST1 and TST2. Setting I2C_VREFCM_TST to 1, connects the analog signals VCMo to TST1 and VREFCM to TST2 pins. Or, setting I2C_VREFPM to 1, connects VREFP and VREFM to pins TST1 and TST2, respectively.

7.3.9 Stand alone high voltage switch

In order to investigate the operation of the proposed high voltage sampling switch, a third switch is integrated into the chip. It shares the supply voltage and clock signal with the rest of the circuit while its input and output nodes are connected to separate pins of the chip, IN_SA and OUT_SA. This high voltage sampling switch is activated by setting I2C_ENHVSW to 1. Then, the operating conditions such as sampling frequency and input signal range are determined by the applied clock and input signals respectively. The output waveform can be observed via an oscillator.

7.3.10 Observation of state control signals

Although not very likely, a problem may occur on the digital part of the system, state machine. In that case the output created by the SARADC would probably be nonsense and no meaningful data can be acquired. In order to make sure that the state machine operates properly, the system is designed so that the control signals created by the state machine can be observed from outside the chip. I2C_TST inputs are used to connect the control signals of the state machine to the serial output and output sample pins.

The output configuration controlled by the I2C_TST pin is summarized in Table **7**. In the default configuration (000000), "serial data out" (OUT_SER) and "output sample" (OUT_SAMP) signals are connected to the output pins. The first 3 bits control the OUT_SER pin and the second 3 bits control the SAMP_OUT pin. En_sar is the control signal that enables each of the conversion states. Amplify is used to set the differential amplifier to amplify state, not comparator state. As an example, an input of I2C_TST = 011100 connects SAMPLE and PSMPL signals to OUT_SER and SAMP_OUT pins, respectively; so that, the timing of these signals can be observed via an oscilloscope at the output of the chip.

Table 7: I2C_TST control configuration.

I2C_TST	OUT_SER	SAMP_OUT
000000	OUT_SER	SAMP_OUT
001xxx	En_sar	
010xxx	En_sar	
011xxx	SAMPLE	
100xxx	PSMPL	
101xxx	Amplify	
110xxx	T&H_B	
111xxx	RSMPL	
xxx001		En_sar
xxx010		En_sar
xxx011		SAMPLE
xxx100		PSMPL
xxx101		Amplify
xxx110		T&H_B
xxx111		RSMPL

8. CONCLUSIONS

In this study, a wide input range 14 bits, 1MSps successive approximation register analog to digital converter is implemented using AMS 0.35 µm high voltage CMOS process with drain extended NMOS (DNMOS) capability.

A novel high voltage sampling bootstrapped switch (HVSBS) has been designed and implemented. The design details are provided and simulation results are given. The implemented switch is integrated within a switched capacitor attenuator for experimental testing. Test results proved that the switch can sample input voltages up to 25V (highest voltage available within the lab equipment) with a supply voltage of as low as 2.2V. Moreover, no reliability problems were faced after long periods of continuous sampling (5 days).

After that, the proposed switch is used to implement a high voltage sampling SARADC. As a result, the implemented high voltage sampling SARADC can directly sample high voltages with standard supply voltage. Up to the date of the study, no ADC was reported or implemented to sample such high voltages without extra high supply voltage or input attenuation.

In this report, the algorithm used for the operation of the SARADC is presented with circuit configurations for each state. The design of the sub-blocks realizing the algorithm is explained and their performance are illustrated with simulation results. The operation of the system is analyzed mathematically, and governing equations are derived. The non-idealities of the devices are discussed and their possible affects on the system performance are investigated. Finally, the tests to be conducted are described. The problems that may be faced during testing are explained and the solution is presented if possible.

The proposed high voltage sampling technique can be implemented in different ways to design analog to digital converters with different specifications. For example, removing subranging from the algorithm and using a 16 bit capacitive DAC, a 16 bit SARADC may be achieved. Moreover, since the subranging is removed, amplification and resample periods are eliminated and sampling rate increases as well. Furthermore, since the differential amplifier is replaced with a comparator, the power consumption reduces as well.

Because of the difficulty in simulating the reliability, the input signal range is kept conservative in this application. Once the reliability and performance is characterized, the input signal range may be increased. The upper limit of the sampling voltage is determined only by the drain break-down of the DNMOS, i.e. 50 V, making up a total input signal range of 100 Vpp in differential mode.

Another drawback of the implemented system is its test modes, which complicates the circuitry and adds additional pads to the layout. If the results of the performance tests are as expected, these test modules may be removed. As a result, the digital part will be less then half of its present value, both in terms of layout and power consumption.

To sum up, the proposed high voltage sampling technique enables a completely different method of handling wide input range analog signals. The method can be used to implement different ADCs with different specifications. Yet, the performance of the SARADC must be proved by the measurement and tests.

REFERENCES

[1] **Ball, W. W. R. and Coxeter**, **H. S. M.,** 1987**.** *Mathematical Recreations and Essays,* Thirteenth Edition, Dover Publications, pp. 50- 51.

[2] **Schelleng, J. C.,** 1948. Code Modulation Communication System, *U.S. Patent,* No: 2453461 dated 09.11.1948.

[3] **Gordon, B. M. and Talambiras, R. P.,** 1963. Signal Conversion Apparatus, *U.S. Patent*, No: 3108266 dated 22.10.1963.

[4] **Cho, T.B. and Gray, P. R.,** 1994: A 10-bit, 20-MS/s, 35-mW pipeline A/D converter, *Proceedings of the IEEE Custom Integrated Circuits Conference.*

[5] **Abo, A. M and Gray, P. G.,** 1999. A 1.5-V, 10-bit, 14.3-MS/s CMOS Pipeline Analog-to-Digital Converter. *IEEE Journal of Solid-State Circuits*, Vol. **34**, no. 5.

[6] **Url-1** *http://en.wikipedia.org/wiki/Bootstrapping,* accessed at 13.03.2010

[7] **Aksin, D. Y., Al-Shyoukh, M. and Maloberti, F.** 2006. Switch Bootstrapping for Precise Sampling Beyond Supply Voltage. *IEEE Journal of Solid-State Circuits,* Vol. **41**, no. 8.

[8] Linear Technology Chronicle, 2006,. Vol:15 No:02 p:1-3

[9] Chip Sensors, 2009, AN004, Application Note, Applications for the Auxiliary Sensor ADC of the CS124, p:3

[10] Linear Technology Design Notes, 2008,. Design Note 452, Power Monitor for Automotive and Telecom Applications Includes ADC and I2C Interface p:1

[11] **Url-2** *http://en.wikipedia.org/wiki/Hall_effect_sensor*, accessed at 09.09.2010

[12] **Aksin, D. Y. and Ozkaya, I.** 2009: 25V Sampling Switch for Power Management Data Converters in 0.35 μm CMOS with DNMOS, *Proceedings of ESSCIRC.*

[13] **Mitros, J. C., Chin-Yu, T., Shichijo, H., Kunz, M., Morton, A., Goodpaster, D., Mosher, D. and Effland. T. R.,** 2001. High voltage drain extended mos transistors for 0.18-μm logic CMOS process. *IEEE Journal of Transactions on Electron Devices,* Vol. **48**, no. 8, pp. 1751–1755.

[14] **Banu, M., Khoury, J. and Tsividis, Y.,** 1988. Fully differential operational amplifiers with accurate output balancing. *IEEE Journal of Solid-State Circuits,* Vol. **23**, no. 6, pp. 1410-1414.

[15] **Yee, Y.S., Terman, L.M. and Heller, L.G.,** 1979. A two-stage weighted capacitor network for D/A-A/D conversion. , *IEEE Journal of Solid-State Circuits,* Vol. **14** , Issue: 4.

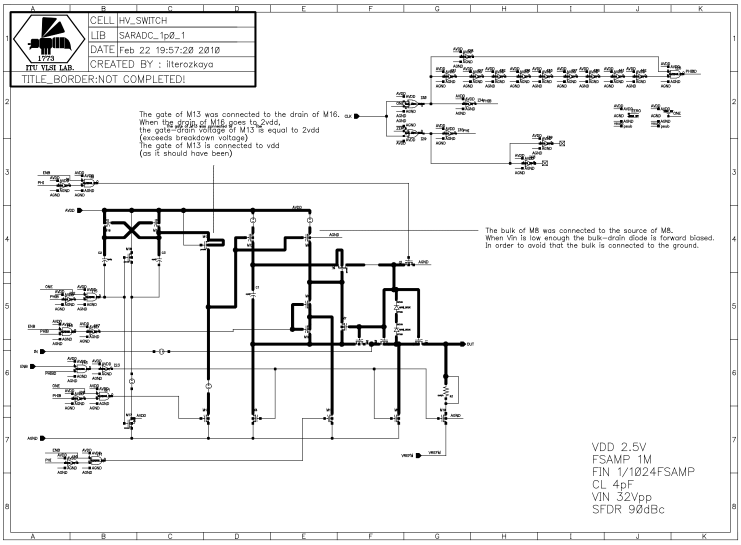
[16] **Maloberti, F., 2007.** *Data Converters,* Springer.

[17] **Cong, Y. and Geiger, R. L.,** 2000. Switching Sequence Optimization for Gradient Error Compensation in Thermometer-Decoded DAC Arrays. *IEEE Journal of Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* Vol. **47**, no. 7.

[18] **Kuyel, T.,** 1999: Linearity Testing Issues of Analog to digital Converters, *Proceedings of International Test Conference*, page(s): 747 - 756

APPENDICES

APPENDIX A.1 : Schematics and Layout



The gate of M13 was connected to the drain of M16.
When the drain of M16 goes to 2vdd,
the gate–drain voltage of M13 is equal to 2vdd
(exceeds breakdown voltage)
The gate of M13 is connected to vdd
(as it should have been)

The bulk of M8 was connected to the source of M8.
When Vin is low enough the bulk–drain diode is forward biased.
In order to avoid that the bulk is connected to the ground.

VDD 2.5V
FSAMP 1M
FIN 1/1024FSAMP
CL 4pF
VIN 32Vpp
SFDR 90dBc

Figure A.1: High voltage switch schematics

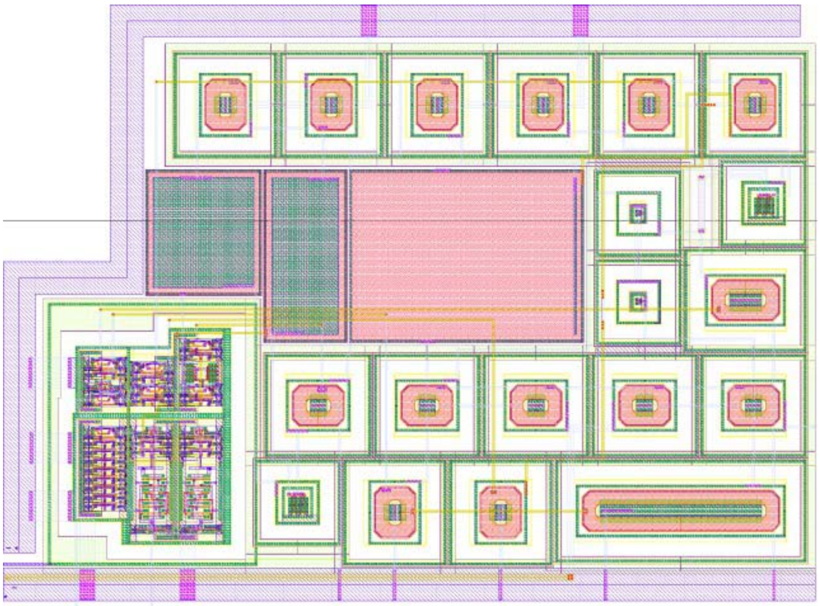


Figure A.2: Layout of high voltage sampling switch.

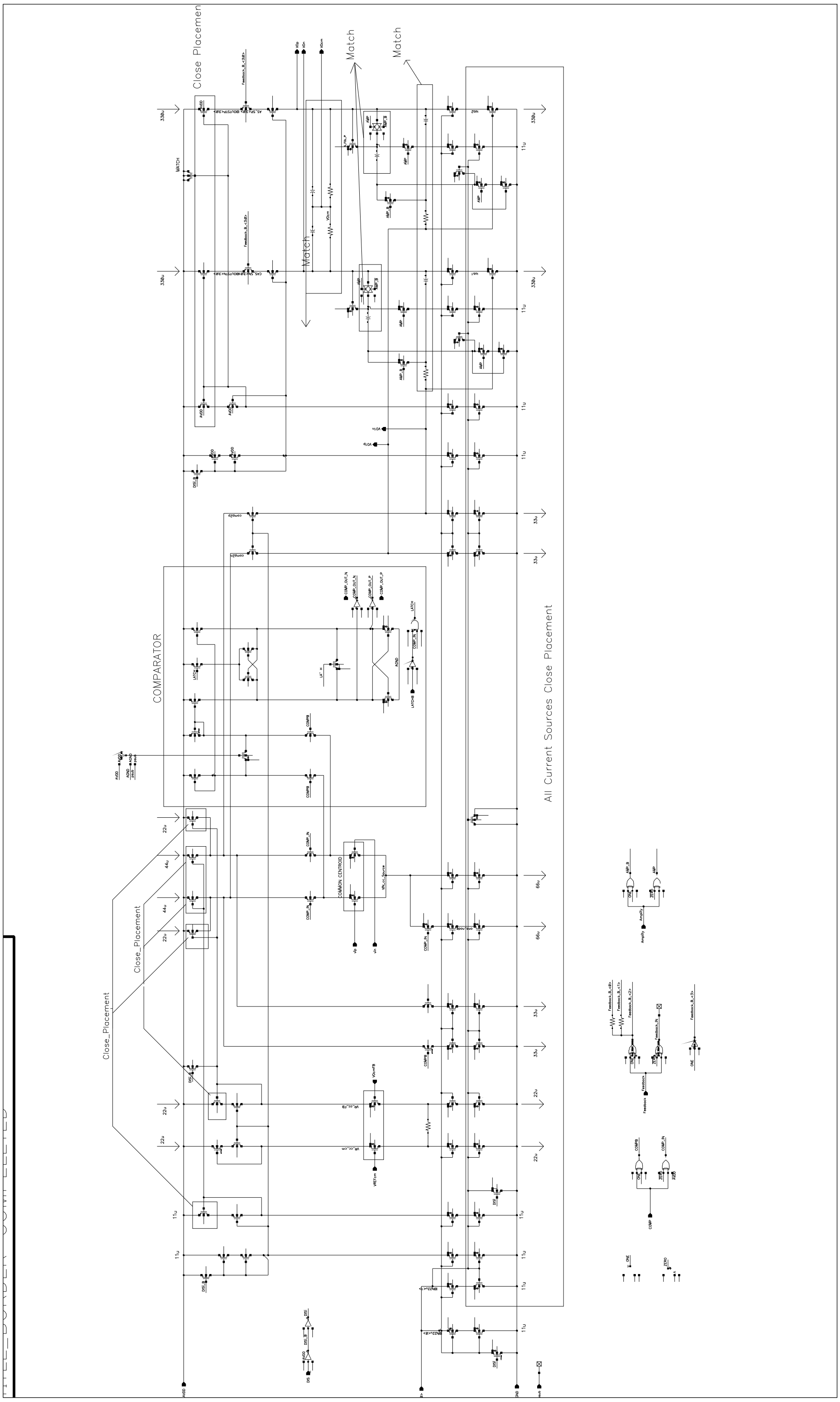


Figure A.3: Schematics of fully differential operational amplifier.

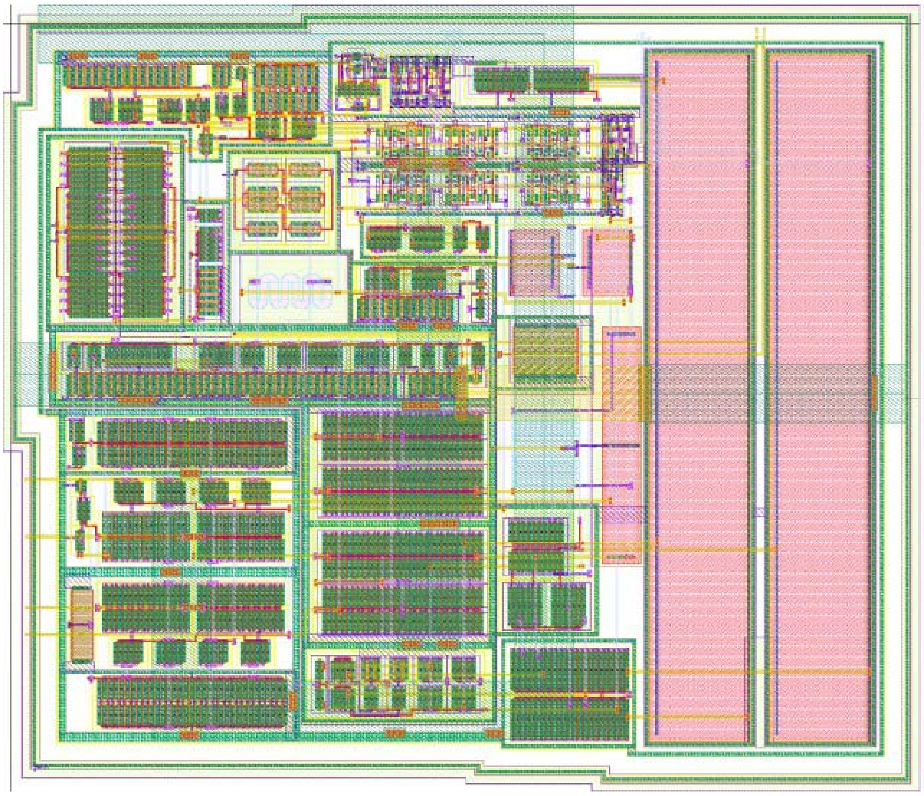


Figure A.4: Layout of the differential amplifier.

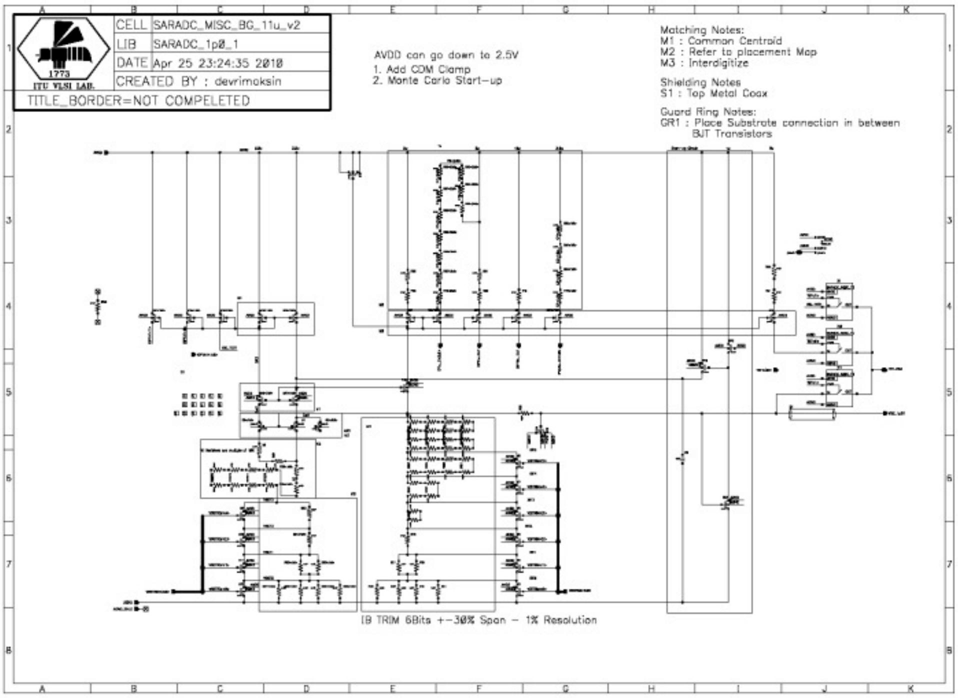


Figure A.5: Schematics of the bandgap reference.

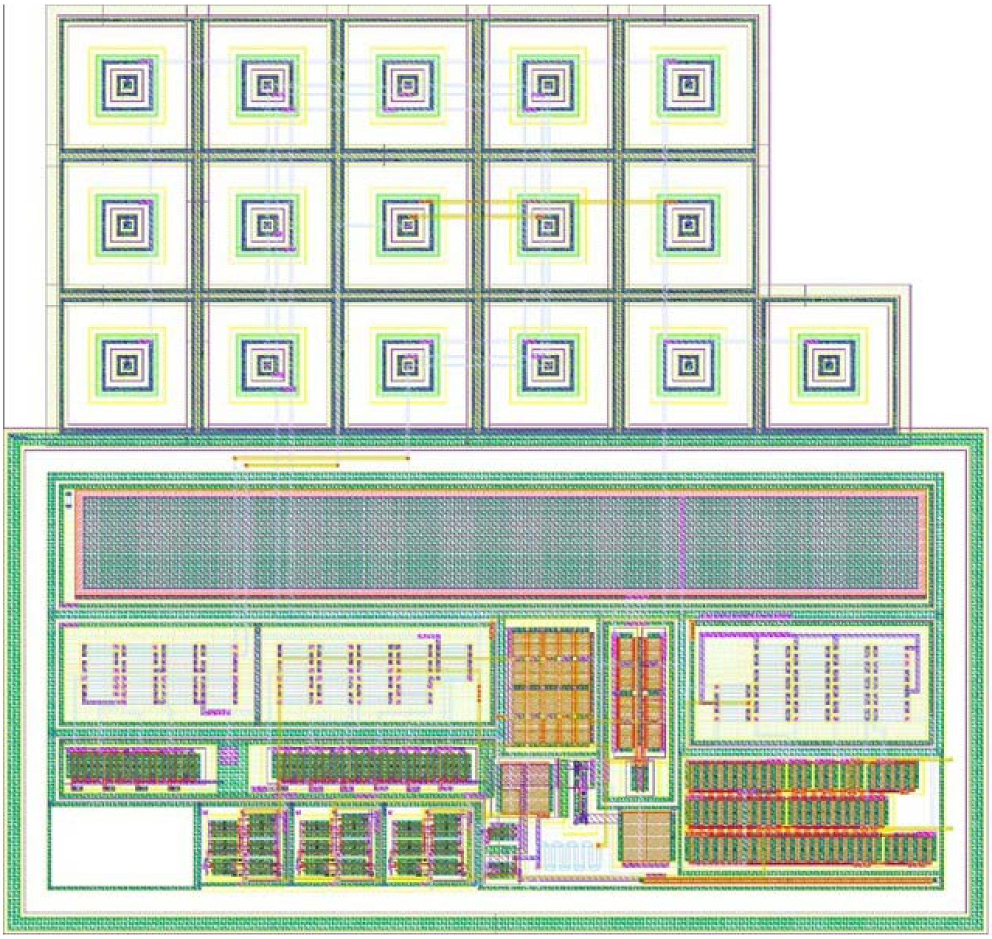


Figure A.6: Layout of the bandgap reference.

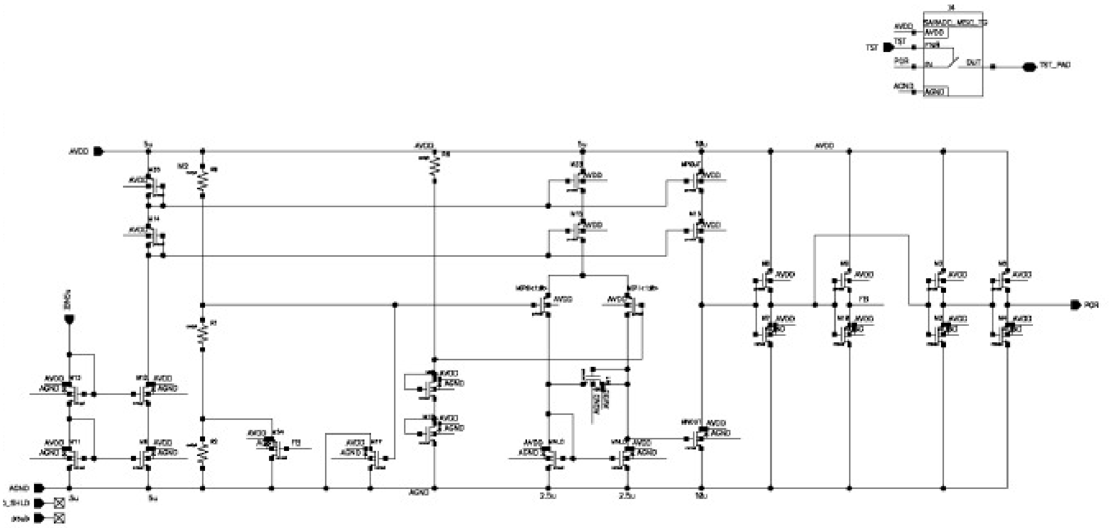


Figure A.7: Schematics of power on reset.

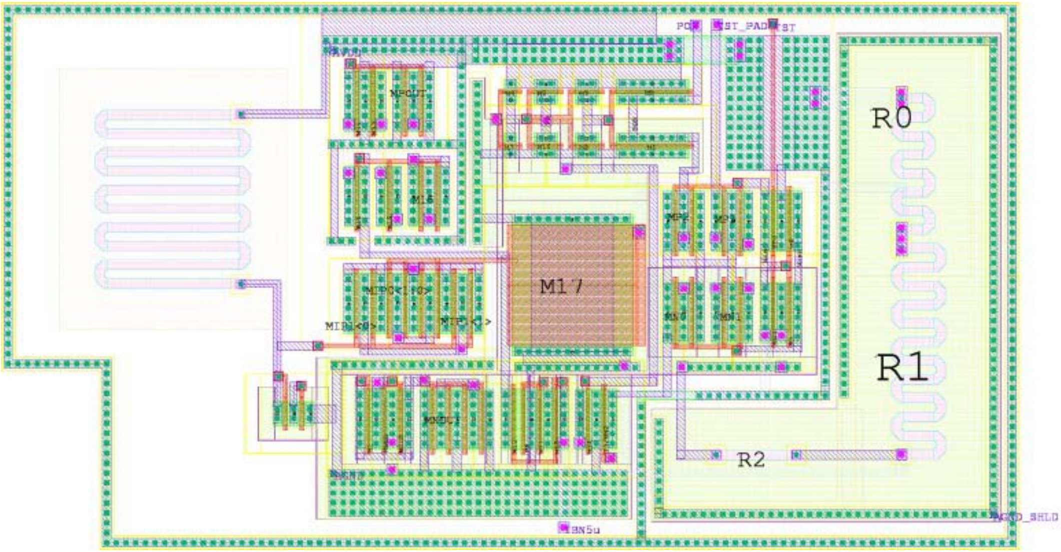


Figure A.8: Layout of the power on reset.

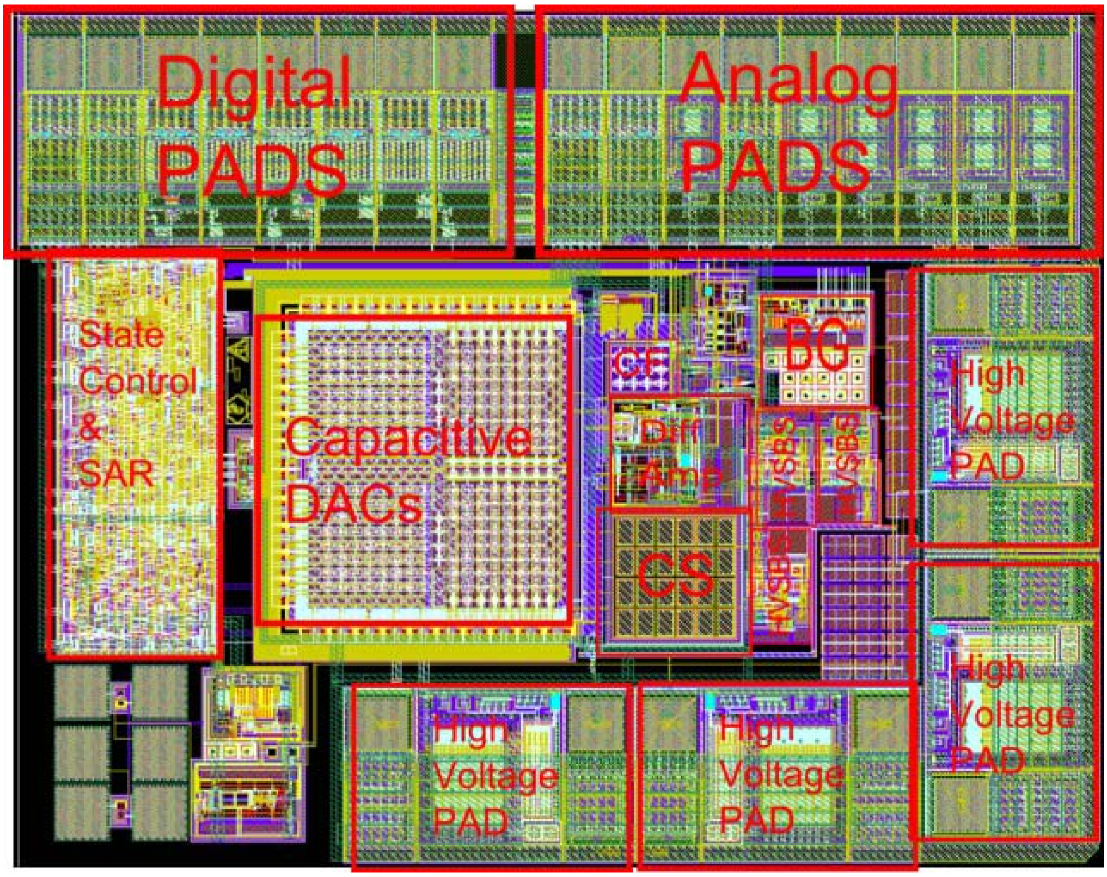


Figure A.9: Layout of the top level chip.

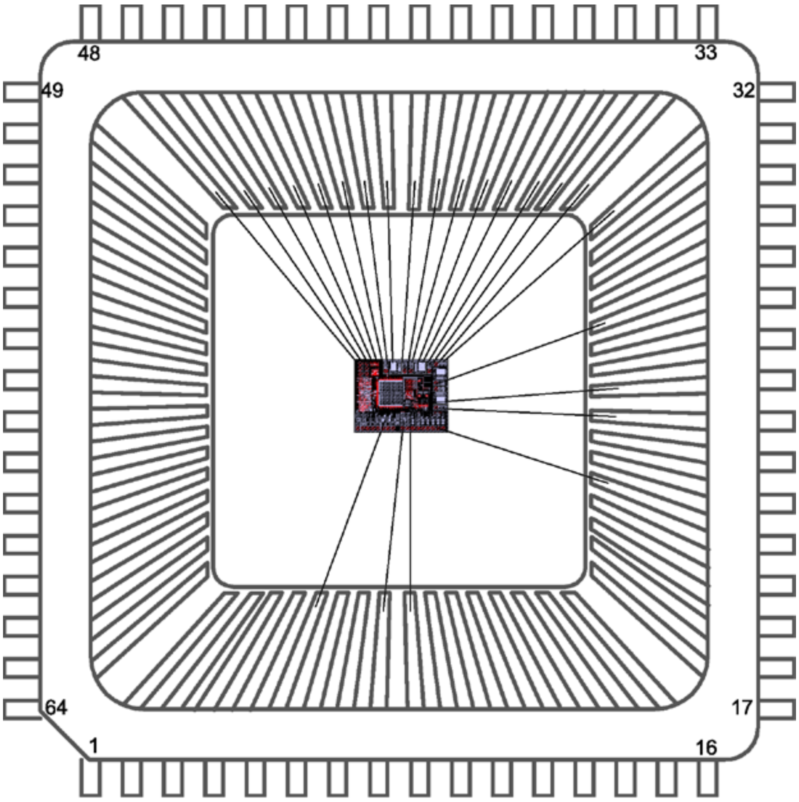


Figure A.10: Bonding diagram.

APPENDIX A.2 : Verilog Codes

//Verilog Code for SAR Algorithm for 8 Bits

```
`timescale 1ns / 1ps
module SAR1(RSTB, START, CLK, comp, comp_EN_OUT, sample_OUT, resample_OUT, SARREAD);
// Pins
        input RSTB;                             // Reset input
        input START;
        input CLK;                              // Clock input
        input comp_EN_OUT;                      // enable conversion
        input comp;                             // Input from comparator
        input sample_OUT;
        input resample_OUT;
        output [7:0] SARREAD;           // SAR Output
// Output Registers
        reg [7:0] SARREAD;
// Internal Registers
        reg [3:0] CONVBIT;

always @(posedge CLK or negedge RSTB) begin
        if(!RSTB)                       CONVBIT <= 0;
        else if (START == 0)    CONVBIT <= 0;
        else begin
                if(comp_EN_OUT == 1)
                        CONVBIT <= CONVBIT + 1;
                else
                        CONVBIT <= 0;
        end
end
always @(posedge CLK or negedge RSTB) begin
        if(!RSTB)                       SARREAD <= 8'b10000000;
        else if (START == 0) SARREAD <= 8'b10000000;
        else begin
                if (comp_EN_OUT == 1) begin
                        case(CONVBIT)
                                0: begin
```

```
                    SARREAD[7] <= comp;
                    SARREAD[6] <= 1;            end
                1: begin
                    SARREAD[6] <= comp;
                    SARREAD[5] <= 1;            end
                2: begin
                    SARREAD[5] <= comp;
                    SARREAD[4] <= 1;            end
                3: begin
                    SARREAD[4] <= comp;
                    SARREAD[3] <= 1;            end
                4: begin
                    SARREAD[3] <= comp;
                    SARREAD[2] <= 1;            end
                5: begin
                    SARREAD[2] <= comp;
                    SARREAD[1] <= 1;            end
                6: begin
                    SARREAD[1] <= comp;
                    SARREAD[0] <= 1;            end
                7: begin
                    SARREAD[0] <= comp;         end
                default: begin
                SARREAD <= SARREAD;             end
            endcase
        end
        else if (sample_OUT == 1) begin
            SARREAD <= 8'b10000000;
        end
        else if (resample_OUT == 1) begin
            SARREAD <= 8'b10000000;
        end
        else begin
            SARREAD <= SARREAD;
        end
    end
end
endmodule
```

// Verilog Code for SAR State Machine

```
`timescale 1ns / 1ps
module SSM(RSTB, START, CLK, sampletime, amptime, resampletime, SARREAD, sample, psample, resample, en_sar, tah,
amplify, feedback, SERDIGOUT, OUT_SAMPLE);
// Inputs
    input RSTB;                     // Reset input
    input CLK;                      // Clock input
    input START;                    // Start Sampling

    input [5:0] sampletime;
    input [5:0] amptime;
    input [5:0] resampletime;
    input [7:0] SARREAD;
//Outputs
    output sample, psample, resample, tah, amplify, feedback, en_sar, SERDIGOUT, OUT_SAMPLE;
// Registers
    reg [2:0] state;
    reg [2:0] nx_state;
    reg [5:0] counter;
    reg [5:0] nx_counter;
    reg [15:0] DIGITAL_OUT;
    reg [15:0] DIGOUT_BUF;
    reg [4:0] SEROUT_CNT;
    reg EN_OUT, loadSAR1, loadSAR2, SERDIGOUT, sample, psample, resample, tah, feedback;
    reg amplify, en_sar, OUT_SAMPLE;
//parameters
    parameter convtime              = 9;
    parameter WAIT                  = 0;
    parameter SAMPLE_LOOP           = 1;
    parameter SAR1                  = 2;
    parameter AMPLIFY_LOOP          = 3;
    parameter RESAMPLE_LOOP         = 4;
    parameter SAR2                  = 5;
always @(posedge CLK or negedge RSTB) begin
```

```
        if (!RSTB) begin
                counter <= 0;
                state   <= WAIT;
        end
        else if (START == 0)begin
                counter <= 0;
                state   <= WAIT;
        end
        else begin
                if (counter == 0) begin
                        counter <= nx_counter;
                        state   <= nx_state;
                end
                else begin
                        counter <= counter - 1;
                        state   <= state;
                end
        end
end
always @(*) begin
        case(state)
                WAIT: begin
                        nx_state        = SAMPLE_LOOP;
                        nx_counter = sampletime;        end
                SAMPLE_LOOP: begin
                        nx_state        = SAR1;
                        nx_counter= convtime;           end
                SAR1: begin
                        nx_state        = AMPLIFY_LOOP;
                        nx_counter= amptime;            end
                AMPLIFY_LOOP: begin
                        nx_state        = RESAMPLE_LOOP;
                        nx_counter= resampletime;       end
                RESAMPLE_LOOP: begin
                        nx_state        = SAR2;
                        nx_counter= convtime;           end
                SAR2: begin
                        nx_state        = SAMPLE_LOOP;
                        nx_counter= sampletime;         end
                default: begin
                        nx_state        = WAIT;
                        nx_counter = sampletime;        end
        endcase
end
always @(*) begin
        tah             = 0;
        sample          = 0;
        psample         = 0;
        resample        = 0;
        en_sar          = 0;
        amplify         = 0;
        feedback        = 0;
        loadSAR1        = 0;
        loadSAR2        = 0;
        case(state)
                SAMPLE_LOOP: begin
                        sample  = 1;
                        if (counter > 0)        psample = 1;
                        else                    psample = 0;            end
                SAR1: begin
                        en_sar  = 1;
                        if (counter == 0)       loadSAR1 = 1;
                        else                    loadSAR1 = 0;           end
                AMPLIFY_LOOP: begin
                        amplify = 1;
                        feedback = 1;
                        if (counter < 1) begin
                        tah = 1;
                        amplify = 0;
                        end
                        else amplify = 1;                               end
                RESAMPLE_LOOP: begin
                        tah     = 1;
                        resample = 1;
                        feedback = 1;
                        if (counter > 0)        psample = 1;
```

```
            else                psample = 0;                end
        SAR2: begin
            en_sar      = 1;
            if (counter == 0)   loadSAR2 = 1;
            else                loadSAR2 = 0;               end
        default: begin
            tah                 = 0;
            sample              = 0;
            psample             = 0;
            resample            = 0;
            en_sar              = 0;
            amplify             = 0;
            feedback            = 0;
            loadSAR1            = 0;
            loadSAR2            = 0;                        end
    endcase
end
always @(posedge CLK or negedge RSTB) begin
    if (!RSTB) begin
        DIGITAL_OUT     <= 0;
    end
    else if (START == 0) begin
        DIGITAL_OUT     <= 0;
    end
    else begin
        if (loadSAR1 == 1) begin
            DIGITAL_OUT [15:8]      <= SARREAD [7:0];
            DIGITAL_OUT [7:0]       <= DIGITAL_OUT [7:0];
        end
        else if (loadSAR2 == 1) begin
            DIGITAL_OUT [15:8]      <= DIGITAL_OUT [15:8];
            DIGITAL_OUT [7:0]       <= SARREAD [7:0];
        end
        else    DIGITAL_OUT     <= DIGITAL_OUT;
    end
end
always @(posedge CLK or negedge RSTB) begin
    if (!RSTB) begin
        SEROUT_CNT      <= 0;
    end
    else if (START == 0) begin
        SEROUT_CNT      <= 0;
    end
    else begin
        if (loadSAR2 == 1) begin
            SEROUT_CNT      <= 17;
        end
        else if (SEROUT_CNT > 0) begin
            SEROUT_CNT      <= SEROUT_CNT - 1;
        end
        else begin
            SEROUT_CNT      <= 0;
        end
    end
end
always @(posedge CLK or negedge RSTB) begin
    if (!RSTB) begin
        EN_OUT          <= 0;
        DIGOUT_BUF  <= 0;
    end
    else if (START == 0) begin
        EN_OUT          <= 0;
        DIGOUT_BUF  <= 0;
    end
    else begin
        if (loadSAR2 == 1) begin
            EN_OUT          <= 1;
            DIGOUT_BUF      <= DIGITAL_OUT;
        end
        else if (SEROUT_CNT > 0) begin
            EN_OUT          <= 1;
            DIGOUT_BUF      <= DIGOUT_BUF << 1;
        end
        else begin
            EN_OUT          <= 0;
            DIGOUT_BUF      <= 0;
```

```
                end
        end
end
always @ (*) begin
        if (EN_OUT == 1) begin
                OUT_SAMPLE              = ~CLK;
                SERDIGOUT               = DIGOUT_BUF[15];
        end
        else begin
                OUT_SAMPLE              = 0;
                SERDIGOUT               = 0;
        end
end

endmodule
```

// Verilog Code for Top Level SARADC

```
`timescale 1ns / 1ps

module SM_N_SAR(RSTB, START, CLK, sampletime, amptime, resampletime, comp, TST, INL_DNL, SAR_IN, SER_OUT,
tah_OUT, psample_OUT, feedback_OUT, amplify_OUT, resample_OUT, sample_OUT, SEROUT_SAMP, SARREAD_OUT,
comp_EN_OUT, I2C_TEST_MODE, I2C_counter_TAH);
// Pins
        input RSTB;                             // Reset input
        input START;
        input CLK;                              // Clock input
        input [5:0] sampletime;                 // Desired number of input sampling period
        input [5:0] amptime;                    // Feedback time
        input [5:0] resampletime;
        input [5:0] TST;                        // TEST enable bit
        input comp;                             // comp output

        input INL_DNL;
        input [7:0] SAR_IN;
        input [15:0] I2C_counter_TAH;
        input  I2C_TEST_MODE;

        output SER_OUT, psample_OUT, SEROUT_SAMP,  tah_OUT, amplify_OUT, feedback_OUT;
        output resample_OUT, sample_OUT, comp_EN_OUT;
        output [7:0] SARREAD_OUT;
// Wires
        wire [7:0] SARREAD;
        wire [1:0] SAR_NO;
        wire resample, sample, psample, en_sar, feedback, amplify, OUT_SAMPLE, SERDIGOUT;
//internal registers
        reg [15:0] counter_TAH;
        reg [7:0] SARREAD_INL;
        reg [7:0] SARREAD_OUT;
        reg [3:0] counter_INL;
        reg psample_INL, resample_INL, feedback_INL, tah_INL, feedback_OUT, comp_EN_OUT,  tah_OUT,
amplify_OUT, psample_OUT, resample_OUT, sample_OUT, SEROUT_SAMP, SER_OUT, psample_OUT_1,
resample_OUT_1, sample_OUT_1, feedback_OUT_1, comp_EN_OUT_1, tah_OUT_1, amplify_OUT_1;
        always @ (posedge CLK or negedge RSTB) begin
                if(!RSTB) begin
                        counter_INL <= 0;
                        counter_TAH <= 0;
                end
                else if(START == 0) begin
                        counter_INL <= 0;
                        counter_TAH <= 0;
                end
                else if (INL_DNL == 0) begin
                        counter_INL <= 0;
                        counter_TAH <= 0;
                end
                else begin
                        case (counter_INL)
                                0: counter_INL <= 1;
                                1: counter_INL <= 2;
                                2: counter_INL <= 3;
                                3: counter_INL <= 4;
                                4: counter_INL <= 5;
                                5: counter_INL <= 6;
```

```
            6: counter_INL <= 7;
            7: counter_INL <= 8;
            8: begin
                if(I2C_TEST_MODE == 0)      counter_INL <= 8;   // wait
                else counter_INL    <= 9;                       // disturbed
            end
            9: begin
                if(counter_TAH == I2C_counter_TAH)      counter_TAH <= 0;
                else
                counter_TAH <= counter_TAH + 1;
                counter_INL <= 9;
                end
            default: begin
                counter_INL <= 0;
                counter_TAH<= 0;
            end
        endcase
    end
end
always @ (*) begin
    if(!INL_DNL)begin
        SARREAD_INL     = 0;
        psample_INL     = 0;
        resample_INL    = 0;
        feedback_INL    = 0;
        tah_INL         = 0;
    end
    else begin
        SARREAD_INL     = SAR_IN;
        psample_INL     = 0;
        resample_INL    = 0;
        feedback_INL    = 0;
        tah_INL         = 0;
        case (counter_INL)
            0: begin
                tah_INL         = 1;
                psample_INL     = 1;        end
            1: begin
                tah_INL         = 0;
                psample_INL     = 0;        end
            2: begin
                tah_INL         = 1;
                psample_INL     = 1;        end
            3: begin
                tah_INL         = 0;
                psample_INL     = 0;        end
            4: begin
                tah_INL         = 1;
                psample_INL     = 1;        end
            5: begin
                tah_INL         = 0;
                psample_INL      = 0;       end
            6: begin
                psample_INL     = 1;
            end
            7: begin
                psample_INL     = 0;        end
            8: begin
                feedback_INL    = 1;
                resample_INL    = 1;        end
            9: begin
                if (counter_TAH == 0)       tah_INL = 1;
                else                        tah_INL = 0;
                feedback_INL    = 1;
                resample_INL    = 1;        end
            default:begin
                SARREAD_INL     = SAR_IN;
                psample_INL     = 0;
                resample_INL    = 0;
                feedback_INL    = 0;
                tah_INL         = 0;        end
        endcase
    end
end
always @ (*) begin
    if (INL_DNL == 0) begin
```

```
                        SARREAD_OUT                    = SARREAD;
                        psample_OUT_1                  = psample;
                        resample_OUT_1                 = resample;
                        sample_OUT_1                   = sample;
                        feedback_OUT_1                 = feedback;
                        comp_EN_OUT_1                  = en_sar;
                        tah_OUT_1                      = tah;
                        amplify_OUT_1                  = amplify;
                end
                else begin
                        SARREAD_OUT                    = SARREAD_INL;
                        psample_OUT_1                  = psample_INL;
                        resample_OUT_1                 = resample_INL;
                        sample_OUT_1                   = 0;
                        feedback_OUT_1                 = feedback_INL;
                        comp_EN_OUT_1                  = 0;
                        tah_OUT_1                      = tah_INL;
                        amplify_OUT_1                  = 0;
                end
        end
        always @ (posedge CLK or negedge RSTB) begin
                if(!RSTB) begin
                        psample_OUT          <= 0;
                        resample_OUT         <= 0;
                        sample_OUT           <= 0;
                        feedback_OUT         <= 0;
                        comp_EN_OUT          <= 0;
                        tah_OUT              <= 0;
                        amplify_OUT          <= 0;
                end
                else begin
                        psample_OUT          <= psample_OUT_1;
                        resample_OUT         <= resample_OUT_1;
                        sample_OUT           <= sample_OUT_1;
                        feedback_OUT         <= feedback_OUT_1;
                        comp_EN_OUT          <= comp_EN_OUT_1;
                        tah_OUT              <= tah_OUT_1;
                        amplify_OUT          <= amplify_OUT_1;
                end
        end
        always @(*)begin
                case(TST[5:3])
                        0:SEROUT_SAMP = OUT_SAMPLE;
                        1:SEROUT_SAMP = en_sar;
                        2:SEROUT_SAMP = en_sar;
                        3:SEROUT_SAMP = sample_OUT;
                        4:SEROUT_SAMP = psample_OUT;
                        5:SEROUT_SAMP = amplify_OUT;
                        6:SEROUT_SAMP = tah_OUT;
                        7:SEROUT_SAMP = resample_OUT;
                endcase
                case(TST[2:0])
                        0:SER_OUT = SERDIGOUT;
                        1:SER_OUT = en_sar;
                        2:SER_OUT = en_sar;
                        3:SER_OUT = sample_OUT;
                        4:SER_OUT = psample_OUT;
                        5:SER_OUT = amplify_OUT;
                        6:SER_OUT = tah_OUT;
                        7:SER_OUT = resample_OUT;
                endcase
        end
SSM     U_SSM(RSTB, START, CLK, sampletime, amptime, resampletime, SARREAD, sample, psample, resample,
en_sar, tah, amplify, feedback, SERDIGOUT, OUT_SAMPLE);
SAR1    U_SAR1(RSTB, START, CLK, comp, comp_EN_OUT, sample_OUT, resample_OUT, SARREAD);
endmodule
```

APPENDIX A.3 : Matlab Codes

%Statistical Sim of SARADC for Mismatch (10 Runs)

```
clc
clear
```

```
Bin_Wgh = 10;
No_of_bits = 14;
Gain_Err = 1;
offset = 0;
C_2nd = 0;
C_sin = 0;
step_1 = 512;

clip_high = 2^No_of_bits * 1.01;
clip_low = - 2^No_of_bits * 0.01;
for i=1:10
   tic
   Vin = clip_low : 1/Bin_Wgh : clip_high;
   [Digout Dig_Data Ratio] = SARADC_Model(Vin, No_of_bits, 65);
   Step_Wgh = hist(Digout, [0:2^No_of_bits-1]);

   First_Hist = Step_Wgh(1);
   Last_Hist = Step_Wgh(2^No_of_bits);

   Ideal_Hits = (sum(Step_Wgh) - First_Hist - Last_Hist) / (2^No_of_bits-2);

   Step_Wgh(1) = 0;
   Step_Wgh(length(Step_Wgh))=0;
   DNL = (Step_Wgh - Ideal_Hits)/Ideal_Hits;
   DNL(1) = 0;
   DNL(length(DNL))=0;

   toc;
   time = toc
   i=i
   DNL_all(:,i)=DNL;

   INL = cumsum(DNL);
   INL_max(i) = (max(INL)-min(INL))/2;
   INL_all(:,i)=INL(2:length(INL)-1);
end
figure(5)
plot(DNL_all)
figure(6)
plot(INL_all)
INL_max = INL_max;

function [Digout, Dig_Data, Ratio] = SARADC_Model(Vin, No_of_bits, N)

VCMin = 8;

Range = 16 - (-16);

Vref = 3.3;
Vrefp = 2;
VCMo = 1.65;

Cu = 1;
Cs = 2*Cu;
Cf = 0.25*Cu;

Cp1p = 0.01*Cu;
Cp1n = 0.01*Cu;
Cp1 = 0.01*Cu;

Cp2p = 0.01*Cu;
Cp2n = 0.01*Cu;
Cp2 = 0.01*Cu;

W = 20;
L = 20;

C_mismatch = (0.45/100)/((W*L)^(0.5));

[Binp Binn] = Gen_Cap_Arr_3(Cu, C_mismatch)

N = (sum(Binp) + sum(Binn) + 2*Cf)/(2*Cf)

Vin_Act = (Vin * Range / 2^No_of_bits) - Range/2;
```

```
Vinp = VCMin + Vin_Act/2;
Vinn = VCMin - Vin_Act/2;

Digout = zeros(1, length(Vin));
DigCode1 = zeros(length(Vin), 8);
DigCode2 = zeros(length(Vin), 8);
Dig_Data = zeros(length(Vin), 16);
residue = zeros(1, length(Vin));
resampled = zeros(1, length(Vin));

for i=1:length(Vin)
   % Sample And Divide
   [V_divp V_divn] = Smp_N_Div(Vinp(i), Vinn(i), Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp, Binn);
   % Conversion
   [DigCode1(i,:) V_res1p V_res1n] = Conversion(V_divp, V_divn, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n,
Binp, Binn);
   residue(i) = V_res1p-V_res1n;
   % Amplification
   [V_ampp V_ampn Vx] = Amplify(V_res1p, V_res1n, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp,
Binn);
   % Resampling
   [V_2p V_2n] = Resample(V_ampp, V_ampn, Vx, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp, Binn);
   resampled(i) = V_2p - V_2n;

   % 2nd Conversion
   [DigCode2(i,:) V_res2p Vres2n] = Conversion(V_2p, V_2n, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n,
Binp, Binn);

   % Digital Signal Processing
   Digout(i) = Dig_Sig_Pro(DigCode1(i,:), DigCode2(i,:), N);
   Dig_Data(i,:) = [DigCode1(i,:)  DigCode2(i,:)];

end

Ratio = resampled ./residue;

function [V_divp V_divn] = Smp_N_Div (Vinp, Vinn, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp, Binn)

%Sampling Phase:
Qsp = (Vref - Vinp)*Cs + (Vref - [1 0 0 0 0 0 0 0]*Vrefp)*Binp + (Vref-VCMo)*Cf + (Vref - 0)*(Cp1p + Cp2p);
Qsn = (Vref - Vinn)*Cs + (Vref - [0 1 1 1 1 1 1 1]*Vrefp)*Binn + (Vref-VCMo)*Cf + (Vref - 0)*(Cp1n + Cp2n);
%Capacitive Division:
V_divp = (Qsp + [1 0 0 0 0 0 0 0]*Vrefp*Binp + VCMo*Cf)/(Cs + sum(Binp) + Cf + Cp1p + Cp2p);
V_divn = (Qsn + [0 1 1 1 1 1 1 1]*Vrefp*Binn + VCMo*Cf)/(Cs + sum(Binn) + Cf + Cp1n + Cp2n);

function [DigCode V_res1p V_res1n] = Conversion (V_divp, V_divn, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p,
Cp2n, Binp, Binn)

n = 8;
DigCodep = [1 0 0 0 0 0 0 0];
DigCoden = [0 1 1 1 1 1 1 1];

Qsp = V_divp*(Cs + sum(Binp) + Cf + Cp1p + Cp2p) - DigCodep*Vrefp*Binp - VCMo*Cf;
Qsn = V_divn*(Cs + sum(Binn) + Cf + Cp1n + Cp2n) - DigCoden*Vrefp*Binn - VCMo*Cf;

for i =1:(n-1)
   if (V_divp > V_divn)
      DigCodep(i) = 0;
      DigCodep(i+1) = 1;
      DigCoden(i) = 1;
      DigCoden(i+1) = 0;
   else
      DigCodep(i+1) = 1;
      DigCoden(i+1) = 0;
   end
   V_divp = (Qsp + DigCodep*Vrefp*Binp + VCMo*Cf)/(Cs + sum(Binp) + Cf + Cp1p + Cp2p);
   V_divn = (Qsn + DigCoden*Vrefp*Binn + VCMo*Cf)/(Cs + sum(Binn) + Cf + Cp1n + Cp2n);

end

if (V_divp > V_divn)
   DigCodep(n) = 0;
   DigCoden(n) = 1;
else
   DigCodep(n) = 1;
   DigCoden(n) = 0;
```

```
end

V_divp = (Qsp + DigCodep*Vrefp*Binp + VCMo*Cf)/(Cs + sum(Binp) + Cf + Cp1p + Cp2p);
V_divn = (Qsn + DigCoden*Vrefp*Binn + VCMo*Cf)/(Cs + sum(Binn) + Cf + Cp1n + Cp2n);

V_res1p = V_divp;
V_res1n = V_divn;

DigCode = DigCodep;

function [V_ampp V_ampn Vx] = Amplify(V_divp, V_divn, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp,
Binn)

Vx = (V_divn*Cf + V_divn*(sum(Binn) + Cs + Cp1n + Cp2n) + V_divp*Cf + V_divp*(sum(Binp)+Cs + Cp1p + Cp2p))...
    /(sum(Binp) + sum(Binn) + 2*Cs + 2*Cf + Cp1p + Cp2p + Cp1n + Cp2n);

Vamp = (- V_divp*Cf - V_divp*(sum(Binp)+Cs + Cp1p + Cp2p) + Vx*(sum(Binp)+Cs + Cp1p + Cp2p + Cf))/Cf;

V_ampp = VCMo + Vamp;
V_ampn = VCMo - Vamp;

function [V_2p V_2n] = Resample(V_ampp, V_ampn, Vx, Vref, Vrefp, VCMo, Cu, Cs, Cf, Cp1p, Cp1n, Cp2p, Cp2n, Binp,
Binn)

V_2p = ((Vref - V_ampp)*sum(Binp) + Vref*(Cs + Cp1p) + (Vx - V_ampp)*Cf + Vx*Cp2p + [1 0 0 0 0 0 0 0]*Vref*Binp +
VCMo*Cf) ...
    /(sum(Binp) + Cs + Cp1p + Cp2p + Cf);

V_2n = ((Vref - V_ampn)*sum(Binn) + Vref*(Cs + Cp1n) + (Vx - V_ampn)*Cf + Vx*Cp2n + [0 1 1 1 1 1 1 1]*Vref*Binn +
VCMo*Cf) ...
    /(sum(Binn) + Cs + Cp1n + Cp2n + Cf);

function [DAC_out] = Dig_Sig_Pro(DigCode1, DigCode2, N);

Bin_Line = [2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0];

DAC_out = [DigCode1 DigCode2]*[N*Bin_Line Bin_Line]';
```

CURRICULUM VITAE

Candidate's full name: İlter ÖZKAYA

Place and date of birth: Bulgaria/03.12.1983

Permanent address: İstanbul/TURKEY

Universities and colleges attended: Middle East Technical University

Publications:

- Aksın, D. Y. and **Özkaya, İ**., 2009: 25V Sampling Switch for Power Management Data Converters in 0.35 µm CMOS with DNMOS, *Proceedings of ESSCIRC*.