Implementing a new Analog to Digital Convertor Signature testing method and Estimating Aliasing.

By

Tajinderpal Singh Toor

Bachelor of Engineering

A thesis

presented to Toronto Metropolitan University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2024

Authors Declaration

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis,

including any required final revisions, as accepted by my examiners.

I authorize Toronto Metropolitan University to lend this thesis to other institutions or individuals for the

purpose of scholarly research.

I further authorize Toronto Metropolitan University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public

Abstract

By moving testing on chip using additional hardware, Built in Self Testing (BIST) can be implemented. Online BIST is a methodology which detects errors using additional bits, specifically parity bits. Analog to Digital Convertors (ADC) are commonly tested by introducing a duplicate secondary ADC. This “secondary” ADC performs testing by producing a compressed output. Due to a compressed range of values available for testing, undetected errors occur. This is known as Aliasing and can be described as the probability of undetected errors. Residue Number Systems(RNS) is a mathematical topic that uses remainders to provide benefits such as speed and hardware reduction. To reduce aliasing and benefit from RNS, a new signature testing method using dual Successive Approximation Analog to Digital Convertors was introduced. With this method two ADCS are used, where one produces a binary represented modolus. Furthermore, an inequality is presented to estimate the aliasing this method may produce.

Acknowledgements

First and foremost, I would like to thank my supervising professor, Dr. Vadim Geurkov for his guidance and support throughout my time as a graduate student. Dr. Vadim Geurkov was always there to provide advice and help in any way possible. I am more than grateful to have been able to learn from him and have him as my mentor.

I would like to thank my friends, Anoth, Rylan and Wrichiek. I am grateful to have met you guys during my time as a student at Toronto Metropolitan University. Thank you for always being proud of me and encouraging me to pursue my graduate studies. A special thanks is also due to my good friend, Rohit. Thank you for always supporting me when times get tough. You were always there with encouraging words even when I may not have believed in myself.

I would also like to thank Rajwant, Hajit, Amunpreet and Amratpaul my mom, dad, sister, and brother-in law for supporting me and encouraging me over the duration of my graduate studies. Whenever times got tough, you guys were always there with kind words. I am grateful for your love, encouragement, and belief in me.

Finally, I would like to say thanks to everyone involved. Thank you for being a part of one of my biggest accomplishments, without your support this would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ADC Analog to Digital Convertor

ATE Automatic Test Equipment

BIST Built in Self Test

CRC Cyclic Redundancy Check

DAC Digital to Analog Convertor

DNL Differential Non-Linearity

INL Integral Non-Linearity

LSB Least Significant Bit

MSB Most Significant Bit

SAR Successive Approximation Register

SNR Signal to Noise Ratio

1 Introduction

1.1. Motivation

With the multitude of continuous and discrete signals available, the ability to efficiently process or transform these signals and extract information is crucial. Analog to digital converters are data converters that can convert an analog signal that is continuous in both time and amplitude to a digital signal that is discrete in both time and amplitude. Analog to Digital converters have played a vital role in the rapid evolution of digital integrated circuits due to their ability to convert a variety of analog signals such as speech, medical imaging, sonar, and radar. There are a variety of different Analog to Digital Convertors, where each architecture is better suited for specific applications. There is no “one size fits all” architecture, one must be chosen carefully based on the application's requirements.

Aside from data conversion, error free reliable transmission of digital signals is a crucial factor in ensuring that information is processed and output correctly. Error Control coding is a technique often seen in communication systems to detect errors and provide a secure method of data transmission. A common implementation of error control coding or error detection is using redundancy bits. Redundant Bits are simply extra bits that are appended to a message and can be used to detect errors.

Built in self-test or BIST is a methodology commonly used to test analog to digital converters. Built in self-testing methodology involves adding additional hardware to move all testing functions onto the chip itself. One of the advantages of Built in Self Testing is that testing can be completed during normal operation. Online testing is a form of Built in Self Testing and involves taking in information, generating a parity, and using this parity bit to detect errors. Parity Bits are one of the many examples of redundancy bits in error control coding.

Another form of testing Analog to Digital Convertors is by using a dual analog to digital convertor setup. This process involves using two analog to digital converters where one converter (primary converter) is responsible for converting the input and the other is used for testing and detecting errors in the primary converter. Furthermore, this secondary converter is typically seen to be a copy of the primary converter, but produces a compressed output also known as a signature. This compressed output leads to undetected errors. Compressing the output of the testing or second ADC leads to a smaller range of values, hence lowering the range of values that can be used for testing. Compression therefore leads to undetected errors. Undetected errors occur when erroneous values produce an output that cannot be detected or is deemed error free. This introduces the concept of Aliasing. Aliasing can be defined as the ratio between the number of undetected errors and the total number of errors. In certain cases, when there is an error somewhere throughout the process, the error at the output would not be detected, commonly known as undetected errors. Undetected errors occur when there is a corrupted or erroneous value, for which the specific input and modulus combination results in a syndrome value that is error free. Undetected errors drop the accuracy of converters or systems and cause unreliable processing of information. The probability of undetected errors, also known as aliasing rate, can be calculated for a specific system by taking the number of undetected errors and dividing it by the total number of errors.

It is therefore crucial to implement a system which reduces the probability of undetected errors or aliasing rate so that errors can be lowered, and reliable conversion of signals can occur. Furthermore, using two analog to digital converters to perform testing may lead to an increase in hardware as the secondary or testing ADC is essentially a copy of the first. So, it is essential to find an implementation of this system which reduces hardware yet still provides efficient and reliable error detection.

1.2. Thesis Objective

The objective of this work is to introduce a new signature testing method using dual successive approximation analog to digital converters. To be more specific, a dual successive approximation analog to digital converter system involves using two analog to digital converters where one ADC will be used to test the other. This new system is based on online BIST testing and cyclic

redundancy check error detection. Instead of using parity check like regular online testing, the proposed method takes advantage of remainders and implements cyclic redundancy check. To be more specific, the second Successive Approximation Analog to Digital Convertor obtains the modulus of the system input. This new testing method also presents a modified way of calculating Differential and Integral Non-Linearity using signatures, where signatures are obtained using the modolus. The introduction of a secondary Analog to Digital converter with output compression results in the issue of aliasing. The thesis will also present a probability inequality that allows one to estimate the probability of undetected errors or aliasing before developing the system. Using the intended modulus for error control coding, the probability can be determined. Furthermore, using the size of the input range and modulus further characteristics such as total errors and error free syndromes can be found. The inequality was developed using properties of residue number systems. All objectives will be further explained in the following sections.

1.3. Thesis Organization

This thesis consists of six chapters. Chapter 2 features all the background information required to understand the material discussed in this paper. Furthermore, it features a literature review of different Analog to Digital Convertor Architectures, measuring performance, Analog to Digital Convertor testing methods and more. Chapter three discusses the theory presented in this paper, introduces the probability inequality to predict aliasing rate along with proof. Chapter four discusses the design implemented for testing and a breakdown of the individual components. Chapter 5 discusses the results, signature testing method using Successive Approximation Analog to Digital Convertors and a working proof of the probability inequality presented in this paper. Chapter 6 concludes this work by providing a summary and future work.

2 Background

2.1. Analog to Digital Convertors

With the variety of continuous time signals, such as: speech, medical imaging, sonar, radar, etc. the need for efficient and confident signal processing systems is ever increasing. Furthermore, signal processing systems such as data converters, that convert continuous time analog signals into discrete time analog signals play a vital role in many technologies and therefore have seen a rapid evolution. An ADC or Analog to Digital Convertor essentially converts analog signals into discrete signals. Analog signals are continuous in time and can be thought of as an infinite or uncountable sequence of numbers. On the other hand, digital signals are discrete in time and are countable and finite. Computers operate using discrete time and digital signals, so it is essential to have a method of conversion.

To properly convert or digitize an analog signal, two important functions need to be completed: sampling and quantization. Sampling is essentially taking samples or values at regular intervals of your analog signal. Sampling is done on the x-axis, meaning you are discretizing your time values. To sample a signal, a sampling period or frequency is provided, which is the time in between different values. For example, if you are given a sampling period of 10 hertz, you would sample a signal every 0.1 second. Once sampling is completed, the signal undergoes quantization. Quantization is the process of discretizing your y-axis or amplitude values. During quantization you are essentially assigning discrete amplitude values to the chosen intervals in sampling.

In an analog to digital converter where your input is a continuous time signal such as a voltage and your output is a binary weighted discrete number, you would sample every certain voltages and during quantization you would assign discrete values to the samples.

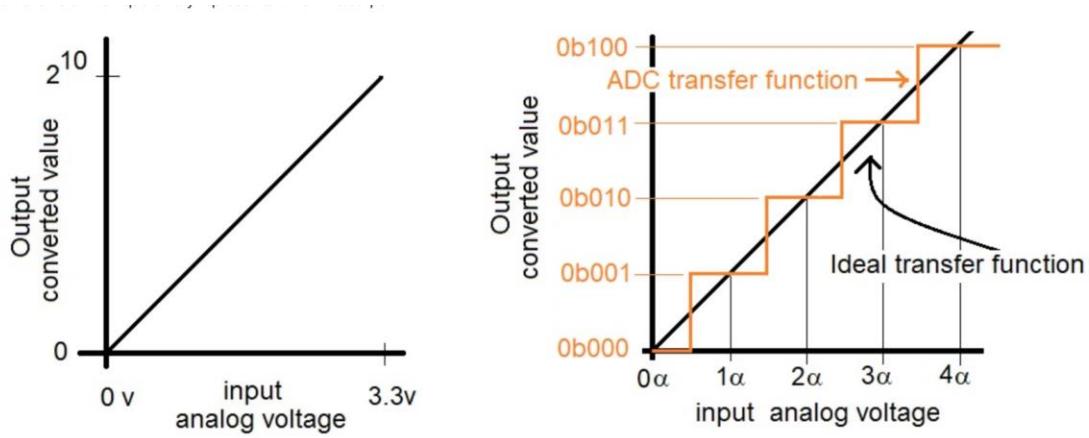


Figure 1. *Sampling and Quantization of ideal transfer function [10]*

In the figure above, assume the alpha (α) is one. For every sample through the process of quantization we are assigning a three-bit binary value for each sample. If you look carefully at the image above, you will notice that when you start increasing the voltage from 0 to 0.5, the binary output remains 000 or if you increase the voltage from 0.5 to 1, the binary output will remain 001. This introduces the concept of quantization error. In the example above, our discrete output could be $\pm$0.5 different from our input voltage. Quantization error is unavoidable and is present in all analog to digital converters. Because of the quantization error, the discrete result of an analog signal can have a range of values. For example, if your analog input voltage is one, with a quantization error of 1 (+ 0.5 and -0.5) your output discrete signal could be 000, 001 or 010. Either of the three outputs would be considered correct.

2.2. Measuring Performance and Error Sources

Before diving into the different types of Analog to digital converters, it is important to understand the various performance characteristics of Analog to digital converters. The various types of continuous signals means there are many different types of ADCs, whose specific performance characteristics fit certain applications. Therefore, there is no one ADC that is suitable for all applications.

The authors in paper [16], discuss performance parameters and sources of error present in analog to digital converter architectures. It is essential to understand these performance characteristics and error sources so one can evaluate the different architectures and see what architecture meets their needs. As per the author, the specific performance parameters to evaluate performance are as follows: Resolution, Linearity, Temperature Sensitivity, Accuracy, SNR - Signal to Noise Ratio, Sampling Rate. Resolution is measured in bits and gives the number of input voltage levels. For example, if the resolution of a converter is 4 bits, there are 16 input voltage levels. SNR, also known as Signal to Noise Ratio, is the relation between the largest value of rms input signal and rms noise value. The SNR is ultimately a ratio that compares the signal to background noise and ideally a higher SNR value is wanted. The formula for SNR can be seen below.

$$SNR = 20\log\left(\frac{Vin(\max)}{Vnoise}\right),$$

$$Vin(max) = \frac{2^N}{2\sqrt{2}}(Vlsb)$$

$$V_{noise} = QE_{RMS} = \frac{Vlsb}{\sqrt{12}}$$

$$SNR = 20Nlog(2) + 20log\sqrt{12} - 20\log(2\sqrt{2})$$

Following performance parameters, the Authors of [16] also presented error sources that are present in ADCS. These error sources are crucial when designing ADC architectures and choosing the right architecture for your application. Authors in [16] presented and discussed the following sources of errors: Quantization Error, DNL or Differential Non - Linearity, INL or Integral Non - Linearity, Offset Error, Gain or scale factor error, Aliasing. It is important to note that the Aliasing discussed here is different from the one being estimated and presented in this paper. DNL - Differential Non-Linearity is the difference between the actual step width and ideal step width. Ideally the step should be 1LSB but because of Differential Non-Linearity, the actual step width could be more or less. For example, if the step width is 1.2, the DNL $= 1.2 - 1 = 0.2$

or if the step width is 0.8, the DNL $= 0.8 - 1 = -0.2$. If the step width is zero, your DNL is 1, which means the digital code for that analog input is missing. The ideal step is $\frac{Vref}{8}$. INL - integral nonlinearity is the integral of all DNL's (differential non linearities). The INL tells you how much the actual transfer function deviates from the ideal transfer function. At any given point the INL is the accumulation of all the DNL's up to that point. Another way to calculate the INL at a given point is to measure the difference between the actual and perfect transfer function.

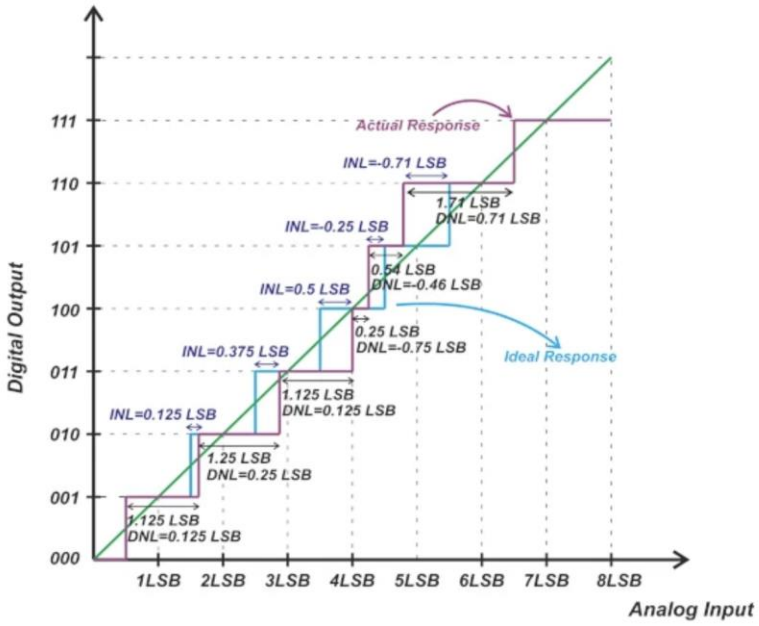


Figure 2. *Example of INL and DNL error [4]*

We can use the figure above to see an example of how to calculate INL and DNL. If we look at the Digital Output 101, the actual transfer functions step width is 0.54. To calculate DNL, we take the difference between the actual and ideal step width (1 LSB), resulting in -0.46.

Furthermore, if you look at the figure above, you can see that if we look at the output code 101, the ideal transfer function is at 4.5LSB, while the actual response is at 4.25LSB. The difference between the actual and ideal transfer function results in an INL error of - 0.25. Using the other method, of adding all previous DNL's, you would get the following expression for INL at 101: $0.125 + 0.25 + 0.125 - 0.75 = -0.25$. The offset and gain (scale factor error) can make

the transfer function deviate from the ideal transfer function. Therefore, before selecting an ADC architecture and resolution, you have to check the offset and gain error to make sure you are making the correct selection. Offset error can be measured when the ideal transfer function shifts left or right on the horizontal axis. To find the offset error, apply a zero voltage to the ADC, if the output is not zero, then an offset error is present. By finding the difference between the ideal and actual value, you can calculate the offset value. Offset error is typically measured in LSB or least significant bits. On the other hand, gain or scale factor error is the difference between the slope of the ideal transfer function vs the actual transfer function. If the slope of the actual transfer function is above the ideal slope, there is a positive gain error, otherwise if the actual transfer function slope is below the ideal transfer function you have a negative gain error.

2.3. Analog to Digital Convertor Architectures

There are numerous different types of ADC architectures, each which have different performance characteristics and suited better for specific applications. As mentioned by authors in [38] the characteristics of different ADCs can be divided into two categories: static and dynamic. Static characteristics are parameters that do not change with time, i.e. transfer function, quantization noise, gain, offset, etc. On the other hand, Dynamic Characteristics are parameters that change with time, i.e. setting time, conversion time. In addition to having static or dynamic characteristics, Analog to Digital converters can be implemented as either serial or parallel. Analog signals are usually first passed through a sample and hold circuit then the sample is compared to different quantization levels and converted to discrete time. If completed in a single step you have a parallel converter. If the comparison between the sample and hold output and quantization levels are done in serial fashion you have a serial converter. Meaning if the different quantization levels are compared in different parts, then it is serial. Besides serial and parallel converters there is a third category called High Performance Architectures which have a high sampling rate and resolution above 16 bits. The section below will discuss the different architectures in detail.

2.3.1. Flash ADC

The Flash ADC is a parallel architecture that is known for its fast speed, simple design and well suited for applications that require a large bandwidth. The Flash ADC uses 2^n comparators to compare the fixed input signal to a varying reference signal. Each comparator has a reference voltage coming from an external source and is equally spaced by VLSB from the largest reference voltage to smallest reference voltage. The fixed input signal to the comparators is the analog input signal that needs to be converted. If your analog input is being converted to a n bit discrete output, the Flash ADC would then need 2^n comparators. Each comparator compares a reference signal to the fixed input signal. The reference voltages are equally spaced between the comparators and come from an external source. If the input voltage is greater than the reference voltage the output of the comparator is 1, otherwise it is 0. The output of the comparators, known as a thermometer code, is fed to an m: n encoder which produces the binary digital output. The limitations of the Flash ADC comes from the comparators. As the resolution increases, the number of comparators also increases which means increased hardware and power.

2.3.2. Counter Type ADC:

2.3.2.1. Counter ADC - Ramp ADC

Another type of ADC is the Counter ADC. This analog to digital converter can be either a Ramp ADC or a Tracking ADC. The Ramp ADC consists of a comparator, AND Gate, digital counter and DAC (Digital to Analog Converter. Essentially the digital counter starts counting from zero and with each clock pulse the value of the counter is increased by a step, converted back to analog form, and compared to the analog input. The AND gate is essentially a control signal which allows the digital counter to increase till it becomes equal to the sampled input. The limitation of this architecture is that it will take 2^{N-1} cycles to convert a sample of data. Furthermore, for each new sample the counter must be preset to zero. With this architecture, the risk of collisions and aliasing may also arise.

2.3.2.2. Counter ADC - Tracking Type ADC

As mentioned above the Tracking Type ADC is another type of Counter ADC. This architecture uses an up/down counter, which means there is no need to preset the counter to zero. The ADC uses previous values and either moves up or down depending on the current sample. The Tracking Type ADC is also known as a Derivative ADC, since the up/down function depends on the difference between current and previous values. The limitation of this architecture is due to the two counters used, as the outputs may oscillate between two states.

2.3.3. Integrating ADC:

2.3.3.1. Integrating ADC - Single Slope ADC

As mentioned above, an example of a Serial Analog to Digital Converter is the Single and Dual Slope ADC. The Single Slope ADC consists of an integrator, comparator, and counter, where the input signal is fed to the integrator. The time taken for the capacitor to charge is proportional to the input voltage. The counter will then count as the same time taken by the capacitor to charge giving an approximate value of the conversion. This architecture is simple, allows for better resolution but has a large conversion time.

2.3.3.2. Integrating ADC - Dual Slope ADC

On the other hand, we have the Dual Slope ADC. In this architecture the integration is performed over a fixed period of time. Integration varies proportionally with input voltage sampled. Following this, the capacitor is then allowed to discharge, and the counter runs throughout this discharging period, which converts the analog to digital signal. This is the most accurate, low power, high resolution data convertor with the least probability of error. The only limitation of the Dual Slope ADC is its slow and has a conversion time of $2 * 2n$ cycles.

2.3.4. High Performance ADC:

2.3.4 .1. High Performance ADC - Delta Sigma ADC

The Delta Sigma ADC consists of two main blocks. One block contains the sigma delta modulator, comparator and a single bit digital to analog converter. The other block contains a digital filter that changes the sigma delta output to binary code with filtering. With the Delta sigma ADC, the input signal is fed to an integrator which gets the rate of change of voltage and feeds this result to a comparator. The comparator will output 0 or 1 based on whether or not a positive or negative slope was present. The result from the comparator is fed to a DAC (Digital to Analog Converter) which feeds the output signal to an op amp. The op amp is part of a feedback loop which takes the output of the Digital to Analog converter and current input signal. The number of integrators present tells you how the architecture is termed. For example, if two integrators are present, the architecture is called 2nd Order Delta Sigma ADC. Due to oversampling this architecture has moderate speed and for a higher order system a larger area is used.

2.3.4 .2. High Performance ADC - Pipelined ADC

The Pipelined ADC is another example of a high-performance analog to digital converter that converts the input signals in a number of stages. A pipelined analog to digital converter is made up of n stages that are used to convert input signals. Each stage has a sample hold circuit, two or more low resolution flash analog to digital converters, k bit DAC, adder/subtractor and a residue amplifier. In the first stage the input signal is passed to a sample hold circuit which takes the output and feeds it to a Flash ADC which in turn generates a binary output. This binary output is converted to an analog signal using a DAC and subtracted from the input. The result is amplified and sent to the next stage. Furthermore, once the next stage receives an input, the previous stage can start processing the next sample. The Pipelined ADC is suited for high resolution, speed applications and provides a high throughput. The limitation is that the power consumption is increased as the stages grow.

2.4. SAR ADC

The Successive Approximation Register Analog to Digital Convertor is one of the many types of analog to digital convertors. The successive approximation analog to digital converter (SAR ADC) compares all possible quantization levels with the input to provide the most accurate output. The SAR ADC uses a successive approximation register in order to provide all the quantization levels needed for comparison. The figure below shows a diagram of all the components needed to construct the SAR ADC.

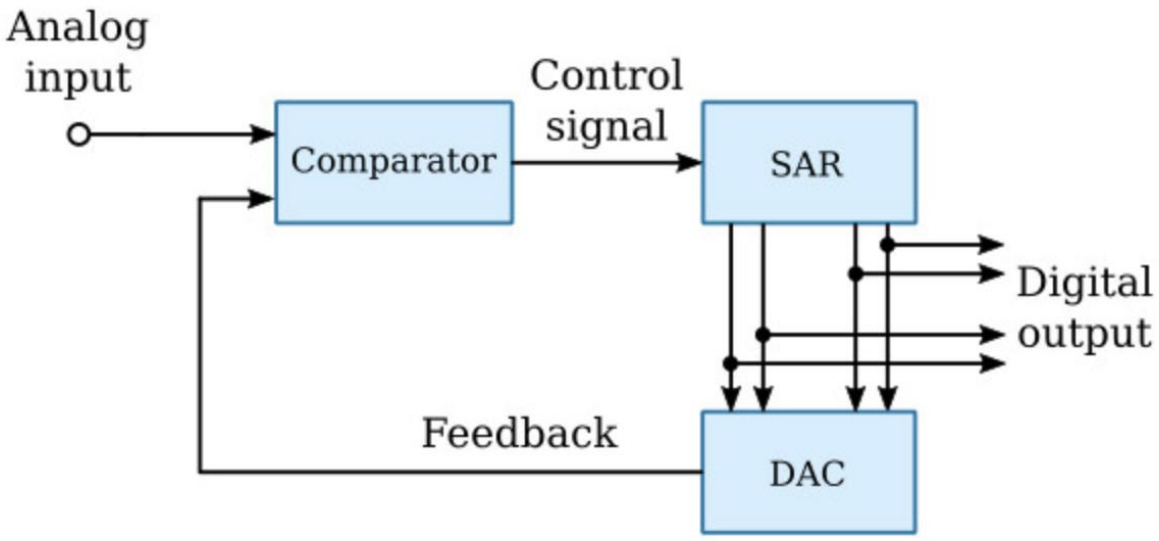


Figure 3*.Component Diagram of Successive Approximation Analog to Digital Convertor [40]*

As seen in the figure above, the SAR ADC is made up of a comparator, Successive Approximation Register and a Digital to Analog Convertor. In addition to the mentioned components, you may also use a sample and hold circuit, which would take in the analog input and output a signal to the comparator. This would then be fed to the Successive Approximation Register. The sample and hold circuit is used to sample our analog input. As the name describes,

the sample and hold circuit will create samples of the voltage it is fed and hold them for a finite time.

As seen in the figure below, the sample and hold circuit is constructed using a transistor, operational amplifier, switch, and capacitor. The switch in between allows the analog input to be sampled, while the capacitor is responsible for holding the analog input. When the switch is closed, the input is sampled while when open the signal is held. When the sample is being held, a constant signal is generated. This constant signal can then be converted to a digital signal using an analog to digital converter.

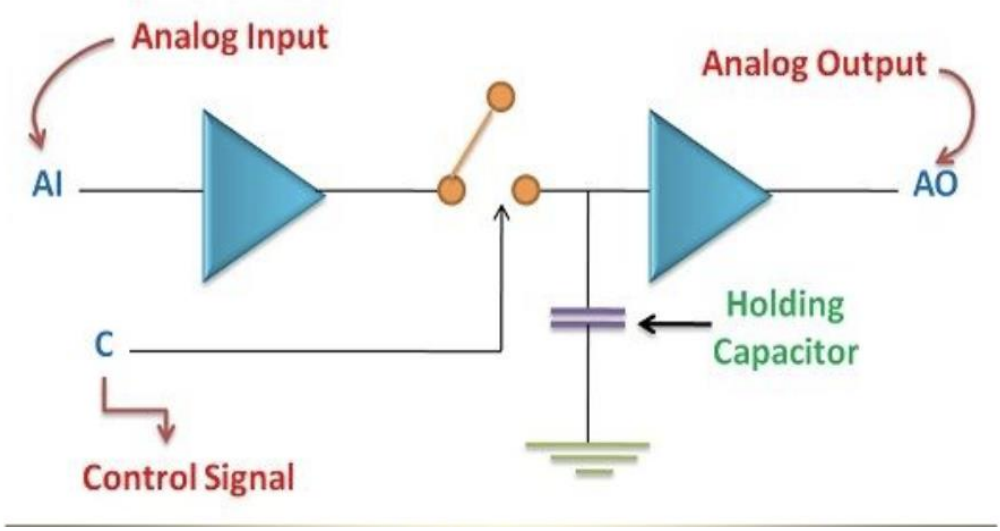


Figure 4*.Sample and Hold Circuit [36]*

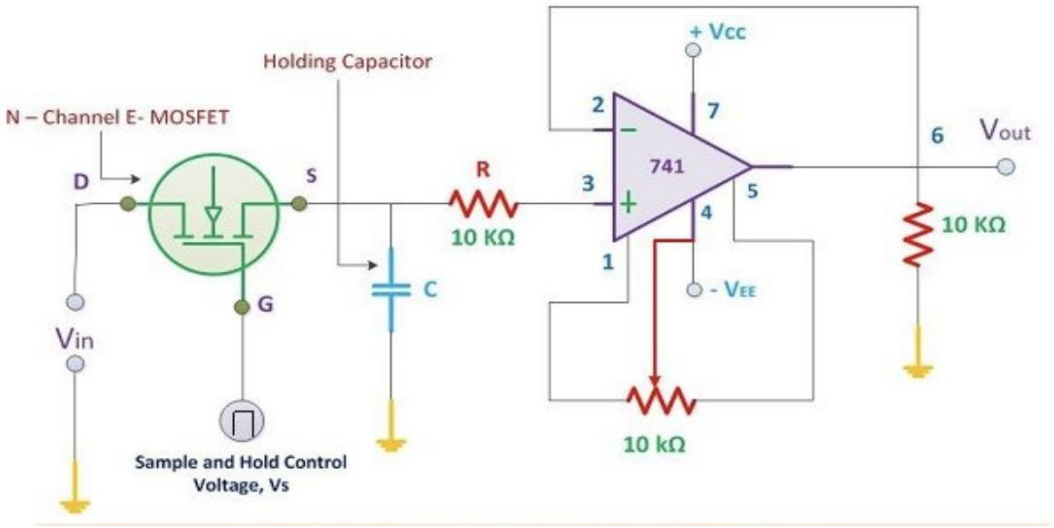


Figure 5*.Sample and Hold Circuit Component Diagram [36]*

The figure above displays the components used to create the sample and hold circuit. The sample and hold circuit features a N-channel Enhancement MOSFET, capacitor and an operational amplifier. The N-channel Enhancement MOSFET will be used as a switch to sample the signal, while the capacitor is responsible for holding the signal.

An analog signal is fed to the drain of the MOSFET, while a control voltage is applied to the gate terminal of the MOSFET. The control voltage fed to the gate, acts like a switch which allows the analog signal to be sampled. When the control voltage is high, the MOSFET is switched ON and acts as a closed switch so the signal can be sampled. When the control voltage is low, the MOSFET is switched OFF and acts as an open switch so the signal can propagate towards the "holding" capacitor. When the MOSFET is acting like a closed switch (i.e. Control voltage high) the "holding capacitor" receives the input analog signal through the drain terminal and charges. When the MOSFET is acting like an open switch (i.e. Control Voltage low), with the help of the operational amplifier the capacitor cannot discharge, so the signal is held for a finite time.

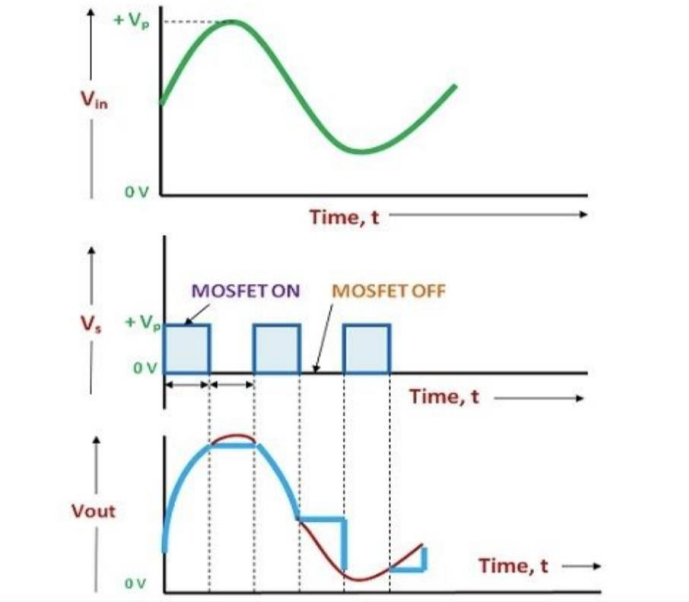


Figure 6.*Voltage vs Time graph of Sample Hold Circuit [36]*

In the figure above, we can see the original analog signal that is fed into the sample and hold circuit (Vin) and its corresponding output. Vs represents the control voltage that is applied to the N-channel Enhancement MOSFET and Vout represents the final analog signal outputted by the sample and hold circuit. When the Vp of our control voltage is high, the MOSFET is ON and acts like a closed switch so the signal can be sampled. As soon as the control voltage is low, the MOSFET is OFF and acts like an open switch, so the signal is held. When the sample is being held, the signal is constant.

2.4.1. SAR Register

The most important component of the Successive Approximation Analog to Digital Convertor is the Successive Approximation Register. As mentioned above the SAR ADC compares all quantization levels with the input to provide the most accurate result. The different quantization levels are provided by the Successive Approximation Register (SAR). Depending on the output of the comparator, the SAR will output a different quantization level.

The comparator compares the input analog signal and the analog output of the SAR. Although the SAR outputs a discrete binary number, it is simultaneously fed to a Digital to Analog Converter, so that it can be used by the comparator. If the input analog signal is greater than the signal received from the SAR, the most significant bit of our quantization level is kept as 1 and the next bit is set to 1. On the other hand, if the input analog signal is less than the analog signal of the SAR, the most significant bit is set to zero and the next bit is set to 1. To understand the working of the SAR, it is necessary to use an example. A sample conversion of an analog to digital signal using the SAR ADC is shown below. In the example below we are using a four-bit SAR, meaning our input signal is in the range of 0 to 15 volts.

1. Input voltage of 11 volts is applied to the SAR ADC and the SAR outputs a default initial value of 1000 in binary.
2. 1000 is converted to 8 volts using the DAC and sent to the comparator. The input voltage is greater than the SAR voltage, so the quantization level changes to 1100 in binary.
3. 1100 is converted to 12 volts using the DAC and sent to the comparator, The input voltage is less than the SAR Voltage, so the quantization level changes to 1010 in binary.
4. 1010 is converted to 10 volts using the DAC and sent to the comparator, The input voltage is less than the SAR Voltage, so the quantization level changes to 1011 in binary.
5. 1011 is converted to 11 volts using the DAC and sent to the comparator. The input voltage is not less than the SAR voltage and we have reached the least significant bit so there are no more quantization levels available for comparison. Therefore, we have reached result 1011 in binary.

The SAR ADC is a power efficient converter and can decrease power consumption to ultra-low ranges in certain areas. This reduction in power consumption comes at the cost of resolution. Furthermore, the SAR ADC is complex, and the cost is greater than other ADC's such as the Counter type ADC.

2.5. Error Control Coding

Error control coding is a technique often seen in communication systems to increase the reliability of transmission and decrease the probability of errors. Error control coding is an important component to this paper, which is why it must be discussed in detail.

A simple form of error control coding or error detection is using parity bits. The message that is being sent is divided into groups of equal length (i.e. same number of bits) and a single parity bit is appended to each group. The parity bit can be either zero or one, ensuring that there are either an even or odd number of bits in each group. Before transmission of your signal, the parity bit is generated. The number of ones or zeros is counted and depending on the result an extra bit is added to your data. Therefore, the bits transmitted to the receiver have increased by k, where k is the number of parity bits. The two types of parity bits are even and odd parity. If even parity is implemented to detect errors and the data has an even number of one's, the parity bit is zero, but if the number of one's is odd, the parity bit is one. On the other hand, if odd parity is being implemented to detect errors and an even number of ones is present the parity bit is one while if there is an odd number of bits the parity bit is zero.

The basic idea of error control coding is to add extra redundancy(bits) to your intended message to detect errors more robustly. As discussed later, residue mathematics are particularly advantageous as they can speed up operations by using remainders. Therefore, the concept of remainders can be introduced into error control coding by using remainders in place of parity bits. This method is known as the cyclic redundancy check or CRC. The steps to implement error detection using CRC are as following:

1. A k number of zeros are appended to your m data bits. The number of zeros is the polynomial equation minus 1. The polynomial equation is also equivalent to your divisor.
2. The $(n + k)$ bit message is divided by the divisor, leaving a remainder.
3. The remainder is appended to your data, creating a message. The $(n + k)$ message bits are now transmitted.
4. On the receiver side, the message (data + remainder) is divided by a divisor and a remainder is found.
5. If the remainder is zero, the message was transmitted error free, but if there is a non-zero remainder, an error was detected.

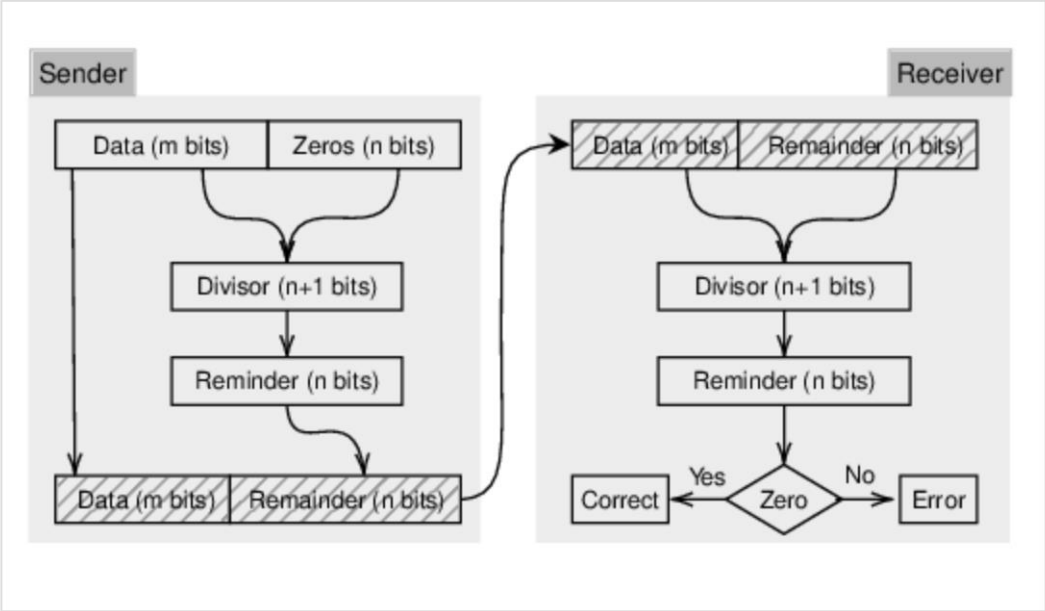


Figure 7*.Cyclic Redundancy Check [25]*

2.6. ADC Testing Methods:

As previously mentioned, Analog to Digital Converters are an integral part of any mixed signal system. Furthermore, the functionality and performance of an analog to digital converter can have a significant impact on the entire system. With the ever-increasing evolution and development of analog to digital converters, cost-effective, high-performance ADCs are being developed and have been the main focus of research. Hence testing to support these convertors has become a problem. The development of testing methods allows you to guarantee that the ADC is functioning properly and meets the requirements of the application.

There are two common Analog to Digital testing strategies, ATE (Automatic Test Equipment) or BIST (Built in Self-Test). When testing Analog to Digital Converters with ATE, the ADC is given arbitrary waveforms, and the test response is collected. This test response can be compared to the ideal output, to verify accuracy. With Automatic Test Equipment, you need connections to

access the inputs and outputs of the ADC to enable communication with the computer software that completes the testing. This computer software has the inputs, test process, stimulus and timing needed to complete testing. Yet there are many challenges that arise with this form of testing.

In paper [17], authors discussed challenges that are faced in Embedded ADC's. Automatic Testing Equipment is a common method to test functionality of Embedded ADCs. The need for the computer software being used in ATE to store and transfer large amounts of data from the ADC represents a significant disadvantage. Furthermore, the external noise and need for high quality test equipment represent reasons why ATE may not be the best choice for testing Embedded ADC systems. As the system speed and complexity increases, the price and complexity of Automatic Test Equipment also increases, therefore research has been increasing in finding BIST Strategies that try to reduce test cost. BIST or Built in Self-Test is another testing method that can be used to verify the functionality of the Embedded Analog to Digital Converters. BIST testing involves moving all or some of the ATE functions onto the chip itself. Built in Self Testing can be used during production or field testing which reduces time and testing cost.

Built-in Self testing methods can be characterized as either Online or Offline. Online testing involves sending input test patterns to the device under test and generating parity bits. Once the device under test produces a result a parity bit is once again generated and compared to the parity bit initially generated. On the other hand, offline testing involves passing input test patterns to the device under test and generating signatures. Signatures are a compressed version of the output from the device under test and are compared to the stored correct responses (ideal signatures). If the difference does not exceed a set limit, the output can be considered correct. Online testing can be used in any process so they can detect early faults, while with offline testing first occurrence of errors cannot be detected. Furthermore, online testing takes its inputs from sources such as on-board sensors while offline testing uses a test pattern generator on chip and needs to be activated during test time to send analog signals to the device under test. Generating test patterns that cover all possible errors is very difficult and with reasons mentioned above, online testing may be the better option depending on the situation.

The authors in [17], describe how in both the Automatic Test Equipment and Built in Self-Test Methods the challenges that arise in designing these methods mainly deal with Test Access Mechanisms, Source/Sink and Test Procedure. Test Access Mechanisms deal with the physical access to the ADC. Gaining access to the physical inputs and outputs can be a challenge as they are usually deep within the System on Chip. Therefore, a test access mechanism which involves using additional logic to connect ADC terminals to test sources and sinks must be present. Essentially the test access mechanism provides a wrapper around the Analog to Digital Converter which implements the functionality mentioned above and can switch between test and normal mode. As previously mentioned, the test mechanism allows for connection to the source and sink. The Source generates the test stimuli given to the input of the Analog to Digital Converter. The Sink on the other hand compares the result taken from the output of the ADC with the ideal output. Sources and Sinks can be implemented on chip with BIST, externally with ATE or through a combination of both test methods. Finally, you have the test procedure. The test procedure essentially provides the test that is being applied to the system. Depending on the test procedure chosen you can face issues such as unconfident test results, inefficient hardware usage and other overheads not expected.

For Built in Self-Test strategies to be effective, there must be some preexisting circuitry present. If the required circuitry already exists it can be shared with other digital blocks. If a Digital to Analog Converter is on the same chip as the ADC testing, you can give a digital input which will be converted and fed to the analog to digital converter, and the digital output can be verified for accuracy. On the other hand, when the DAC is not present the generation of an analog input test signal and digital response analyzer for the ADC output is needed. Most BIST strategies are application limited to specific ADC architectures.

Analog to Digital converters can be tested through either Dynamic or Static Testing. Static testing involves applying an input signal and slowly varying it to create static transfer characteristics such as gain, offset, integral nonlinearity and differential nonlinearity. Dynamic tests involve measuring ADC performance with signals that change with time. Dynamic tests measure parameters such as the total harmonic distortion, spurious free dynamic range, signal to noise and distortion ratio and signal to noise ratio. To complete ADC Static Testing, there is a

need to provide a linear and accurate input test signal along with the ability to collect many samples. The large number of samples are required to ensure an accurate measurement. Furthermore, the static test method chosen should maintain low test cost and hardware area. As discussed by authors in paper [13], there is various research going into cutting down static linearity test time by using reduced code techniques or applying spectral testing methods. To complete an accurate dynamic test, there is an increase in the cost of the test setup, therefore there is a need for research that finds new solutions to reducing cost while keeping an accurate result. A common dynamic testing method is the ADC spectral test but one of the challenges is the need of high-cost equipment (high accuracy signal generators, frequency synthesizers). This problem becomes more relevant with BIST as part of the BIST requirement is to add minimal hardware area. As mentioned above, one of the parameters that need to be tested is the Integral Non-Linearity error and the Differential Non-Linearity Error. A common method of testing and ADC for these errors is by applying a ramp signal. This method has a small overhead which makes it perfect for implementation through a BIST. Due to the output of ADC being digital, it is possible to implement a Built in Self-Test method with digital circuits. Researchers from Yonsei University [27], proposed a simple BIST method which was implemented solely through digital circuitry. The proposed system is shown below.

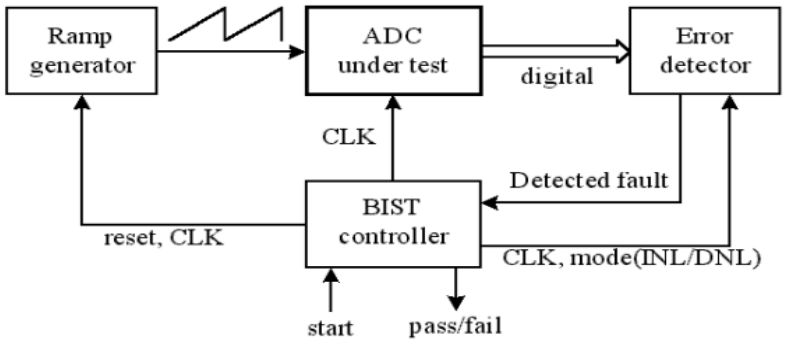


Figure 8.*Proposed Digital Circuit BIST [27]*

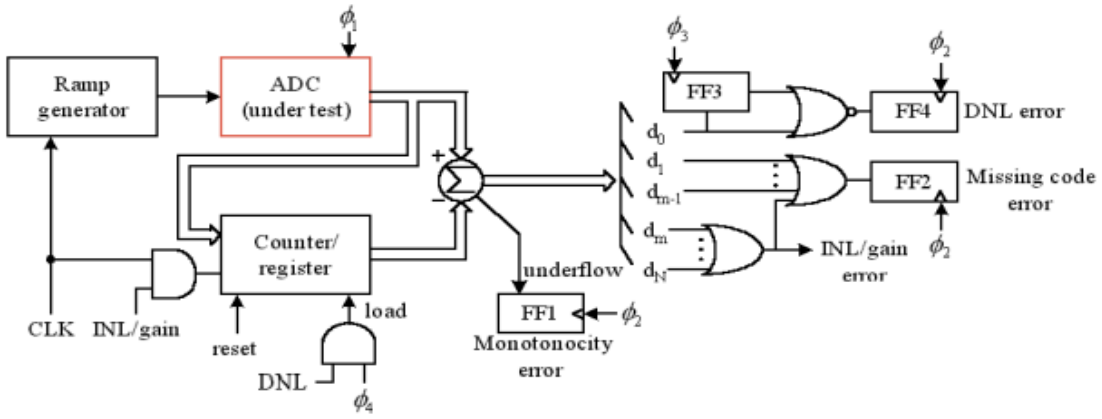


Figure 9. *Proposed Error Detection Circuit [27]*

The design shown in the figure above can handle four errors: Missing Code Error, Monotonicity Error, DNL error and INL/gain error. Each of these errors are explained below. In each of the equations below, the $x(n)$ represents the output of the current sample.

1. **Missing Code Error:** If the difference between two consecutive samples is greater than 1 LSB, this means that the output increased by more than 1 LSB while the input has increased by 1 LSB. The missing code error can be represented by the following formula:

$$x(n) - x(n-1) \geq 2LSB$$

2. **Monotonicity Error:** If the difference is negative then the ADC output decreases while the input is monotonically increased. This can be represented by the equation below.

$$x(n) - x(n-1) < 0$$

3. **DNL Error:** If the ADC generates the same output for 3 samples, then the output is the same while input changes(increased by 3 LSB).This can be represented by the equation below.

$$x(n) = x(n-1) = x(n+1)$$

4. **INL/gain error:** If the ideal (output from ramp signal generator) and actual sample difference is greater than some boundary you have an INL error. This can be expressed by the following equation:

$$ideal(n) - x(n) > boundary$$

In [22], researchers from the University of California Santa Barbara, presented a BIST strategy that performed linear histogram testing and measured INL/DNL errors. Furthermore, this paper also showed how a linear ramp test stimulus can achieve desired test accuracy. The proposed BIST for ADC and DAC testing is shown in the figure below.

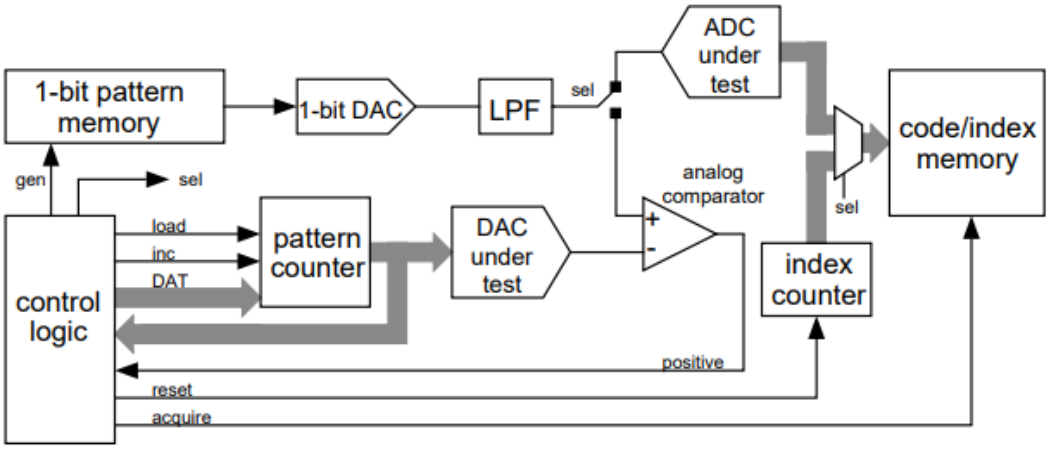


Figure 10*.Proposed BIST for ADC and DAC testing from researchers at University of California Santa Barbara [22]*

To test an ADC using the above architecture the following process is applied. The 1-bit pattern memory stores one period of waveform. The pattern memory is then loaded to the DAC and processed through the low pass filter, where the output is fed to the ADC under test. The result of the ADC under test is then stored in the code/index memory. The collected data is then analyzed to make a pass or fail decision.

2.7. Residue Number System Theory and Modular Arithmetic

2.7.1 Remainder/modulus

When dividing two numbers, you often have a remainder. When the divisor is not a factor of your dividend, the result of division has a “leftover” or remainder. For example, if you are calculating the result of twenty divided by four, the quotient or “result” is zero with remainder of zero. As you can see in the example above the divisor (four) is a factor of your dividend(twenty) which results in a remainder of zero. On the other hand, if you are calculating the result of twenty divided by three, you notice that the divisor(three) is not a factor of your dividend(twenty) which means you will have a remainder. When completing the above example, the quotient is 6, remainder 2 as three can go into the number twenty, six times leaving a remainder of two. Residues are another name for remainders and can be used interchangeably.

2.7.2 Residue number system:

We often work with number systems such as decimal or binary. These number systems rely on weighted positions. For example, the decimal number 15 can be broken down into the weighted positions shown below.

$$15_{10} = 1*10^1 + 5*10^0$$

The subscript of a number defines the base of a number. In the case of the example above, the subscript ten, tell us the number fifteen is base ten or a decimal number. To break down numbers into their weighted positions, we use positional notation. To break down a number into weighted positions, you take the digit and multiply it by the base raised to its position. If you look at the example above, the number fifteen (base ten) is made up of digits one and five. The first digit five is multiple by its position raised to the base. When working with residue number systems, numbers are no longer broken into a weighted number system but rather with their remainder. The residue number system essentially allows you to take a weighted number and convert it into

a residue system, where each position represents a modulus. The moduli of different positions are chosen so that they are relatively prime to one another.

2.7.3 Modular Arithmetic

Modular arithmetic is a system in which you complete operations using residues. Modular arithmetic can be better understood through an example.

$$2403 + 791 + 688 + 4339 = 8221$$

To complete the operation above, you would need to have hardware that would support an input and output of fourteen bits. On the other hand, if we were to calculate the remainder or modulus of each number and use them for arithmetic, we would have a much faster operation. If we were to calculate the modulus ten of each operand, the operation would now look like:

$$3 + 1 + 8 + 9 = 21 = 1$$

Because the operands are converted to a modulus ten, the answer must also be a modulus ten. Once we add all the operands, we get 21, which when converted to modular ten is one. With the above operation in residue number systems, you can complete the operation with hardware that uses five bits. The fewer bits result in better performance and lowest cost/area. Needing fewer bits to perform operations, means you need less hardware to implement circuits, which in turn means a smaller surface area being used.

Depending on the number system or base given, we can use the positions of the digits to see the different moduli of the number. These moduli include the base itself and exponentials of the base. To better understand this, we can look at the following example. Given a four-bit binary number we can see modulus two, four, eight and sixteen. As you can see these moduli are just the base to the power of the digits position.

$1011_2 = 11_{10}$

Mod 2 (2^1): Take every digit up to first position, convert to decimal and you will get the modulus 2 result.

$1_2=1_{10}$, therefore 11mod2 is 1.

Mod 4 (2^2): Take every digit up to second position, convert to decimal and you will get the modulus 4 result.

$11_2=3_{10}$, therefore 11mod4 is 3.

Mod 8 (2^3): Take every digit up to third position, convert to decimal and you will get the modulus 8 result.

$011_2=3_{10}$, therefore 11mod8 is 3.

Mod 16 (2^4): Take every digit up to fourth position, convert to decimal and you will get the modulus 16 result.

$1011_2=11_{10}$, therefore 11 mod 16 is 11.

The steps above can be applied to any number system or base and be used as a fast way to find the remainders or moduli of any base.

3

Theory

3.1. Quantization Error in Analog to Digital Convertors

In order to transform an input signal, the analog to digital converter must assign discrete amplitude values, a process which is known as quantization. This process results in an error that is seen in every type of analog to digital converter architecture and is known as the Unavoidable Quantization Error. The Quantization Error is essentially the difference between the actual analog input value and the nearest quantization level and can typically be due to rounding.

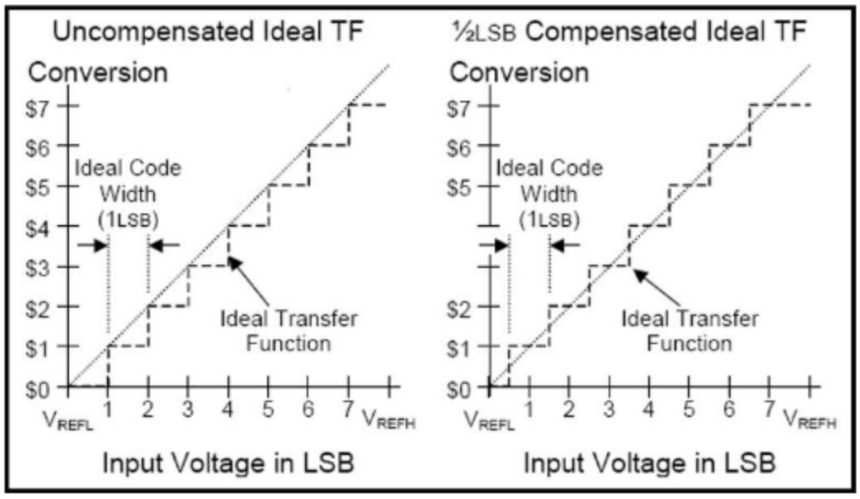


***Figure 11**.Ideal Transfer function with and without 0.5 LSB Shift*

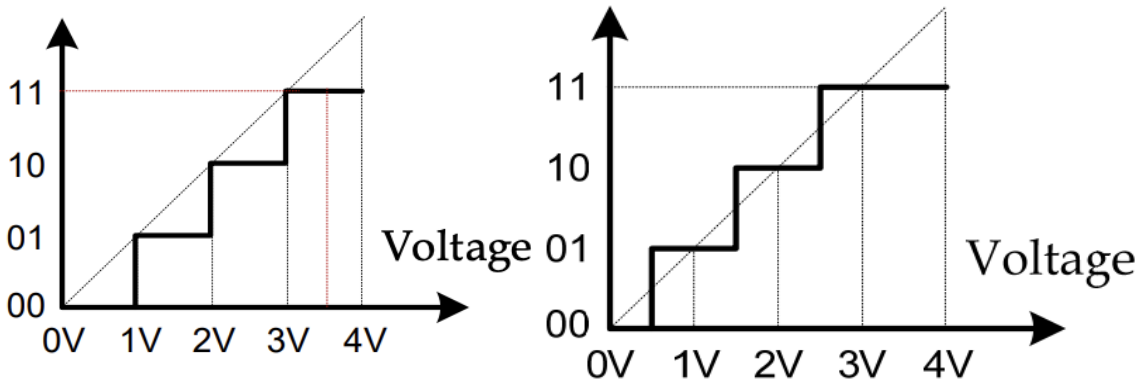
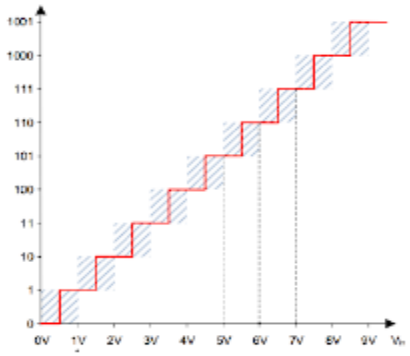


Figure 12*.Discrete values of Input Voltages with and without shift.*

The left figure above is the uncompensated or ideal result of analog to digital conversion. In this graph it can be seen, any input from 0.0 to 0.9 will be converted to 0, any input from 1.0 to 1.9 will be converted to 1, etc. The quantization error for an input of 1.9 volts would therefore be 0.9 (1.9-1.0) . A simple shift 0.5 volts left shown in the figure above to the right, shows how the quantization error can be decreased. Analog to Digital Convertors in the real world, often have a shift in their output to reduce the quantization error. In the compensated result, shown in the right figure, you can see any input from 0.0 to 0.5 would produce a result of zero, while any input from 1.5 to 2 will produce a result of 2. For an input of 1.9, the quantization error would now be 0.1(2-1.9) which is a significantly smaller error.

3.2. Error Detection in Analog to Digital Convertors

A method to detect errors when converting inputs is by using more than one analog to digital converter. Let's now investigate error detection using two analog to digital converters by looking at the following example.



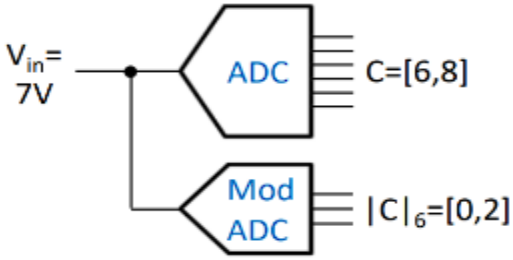


Figure 13 (Left). *Analog to Digital Conversion results for 0 to 9 V,* ***Figure 13 (Right).*** *ADC Error Detecting Method*

In the above example, we are taking an input of 7 volts and using a binary weighted analog to digital converter to calculate the result. The result is shown in the output of the top analog to digital converter, denoted by C. The image on the left shows the results that can be obtained by an input of 0 to 9 volts. If you take into consideration the quantization error of 0.5 volts, we can see from the graph the acceptable results are 6,7 or 8 volts. This is because if you look at the lower 0.5 quantization error your error free output is either 6 or 7, while if you look at the higher 0.5 quantization your error free output is either 7 or 8 volts. This work chose to implement cyclic error control coding, which means the second analog to digital converter is one that outputs the modulus of our input. For this example, the modulus chosen is six. Therefore, modulus six applied to our input, would result in one volt. From the graph shown on the left, we can see that the acceptable results of a one-volt output would be 0,1 or 2 due to the quantization error.

3.3. Syndrome

Now the syndrome can be calculated. To calculate the syndrome, the following equation is used, $Syndrome = |C - C\ modX|_{modX}$. If we use the system mentioned above, the syndrome for an input of 7 volts is 0,1,2,4,5. This set of numbers represents the error free syndrome for an input of seven volts. This result is shown in the wheel below.

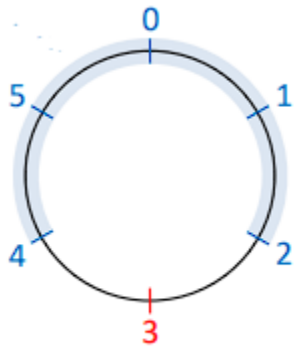


Figure 14.*Possible Syndromes for an input of 7 volts and Modulus 6*

If you look at the wheel above, you see the blue shaded area represents the error free syndrome for an input of seven volts and modulus six. In the same figure the three is unshaded as it is the erroneous syndrome. It is important to note that taking the modulus six of an input and getting a value of 5 is equivalent to getting a value of -1. Similarly, taking the modulus six of an input and getting a value of 4 is equivalent to a value of -2. The error free syndrome can therefore be summarized using the form [-2,2].

3.4. Undetectable errors and Calculating Syndrome

From here, the idea of undetectable errors can be introduced. Undetectable or undetected errors are simply errors that are not detected. To further understand this, we build on to the example above, by applying an error to the output, C. Let's say that an error was applied to the C, where for an input of seven volts you produce an output, C of ten volts. If we keep the modulus result error free and choose it to be 1 which is from the error free remainder range of [0,2], your syndrome results in 3 which means an error was detected. On the other hand, if we introduce an error at the output where with an input of seven volts, your output C is thirteen volts, if the remainder result is kept at 1, the resulting syndrome is 0. As calculated above, a zero syndrome is an error free case for the input value of seven volts, yet our output in this case was thirteen volts. This is where the idea of unpredictable errors is introduced.

Now carrying on with the above example and changing the input to be five volts instead of seven volts, with a modular analog to digital converter that operates on mod 6, you notice that the

syndrome is the same. The error free syndrome for an input of five volts is also 0,1,2,4,5. Furthermore, if you change the input to be six volts, you will notice that the error free syndrome remains the same at 0,1,2,4,5. Noticing above that if the modulus is kept the same, no matter the input voltage you obtain the same error free syndrome. This leads us to developing a generalization. The generalization can be developed from one of the examples above, to develop the generalization we will use an operation with modulus 6 and an arbitrary input voltage of C.

$$S = |C - |C|_6|_6$$

$$S = |C - 1, C, C + 1 - |C - 1, C, C + 1|_6|_6$$

$$S = |C - 1, C, C + 1 - |C - 1|_6, |C|_6, |C + 1|_6|_6 =$$

$$|C - 1 - |C - 1|_6|_6 = \mathbf{0},$$

$$|C - |C - 1|_6|_6 = \mathbf{1},$$

$$|C + 1 - |C - 1|_6|_6 = \mathbf{2},$$

$$|C - 1 - |C|_6|_6 = -\mathbf{1} = \mathbf{5},$$

$$|C - |C|_6|_6 = \mathbf{0},$$

$$|C + 1 - |C|_6|_6 = \mathbf{1},$$

$$|C - 1 - |C + 1|_6|_6 = -\mathbf{2} = \mathbf{4},$$

$$|C - |C + 1|_6|_6 = -\mathbf{1} = \mathbf{5},$$

$$|C + 1 - |C + 1|_6|_6 = \mathbf{0},$$

$$= |[-2,2]|_6 = \ |-2,-1,0,1,2|_6$$

$$= \mathbf{4}, \mathbf{5}, \mathbf{0}, \mathbf{1}, \mathbf{2} - \boldsymbol{no\ errors}$$

From the derivation above, we can conclude that the error free syndrome for the general case (arbitrary input voltage) is:

General Case (mod r): $|[-2,2]|_r = r-2,r-2,0,1,2$ where r is the chosen modulus

For the above generalized formula, three propositions need to be stated:

1. The minimum value r or your modulus needs to be six. If a number less than six is chosen for your modulus, there will be no numbers in the given range to represent errors. The example below, shows a proof of the proposition stated above.

   Ex. **(mod 5):** $|[-2,2]|_r = 3,4,0,1,2$

   For modulus five the result will be any number from zero to four. As seen in the example above, the error free syndrome produced will cover the entire range of modulus five (zero to four). Therefore, there are no numbers left to represent the erroneous case. The same result will be seen for any number less than six.

2. If the shaded areas in the graph below (the transient graph of the analog digital converter) do not overlap, meaning that we are only looking at the intended value and one higher or the intended value and one lower, the general formula for calculating the error free syndrome changes. If the scenario matches the above description, then the general error free syndrome is:

General Case for non-overlapping regions (mod r): $|[-1,1\ ]|_r = r-1,0,1$ where r is the chosen modulus

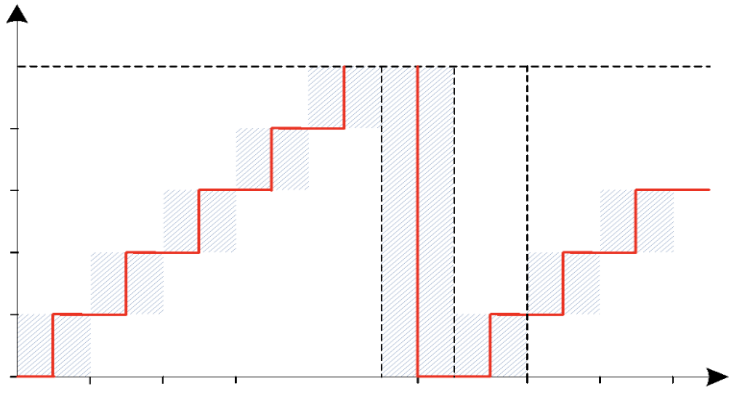


Figure 15.*Non overlapping regions.*

The proof of the above-mentioned formula is shown below:

1. Negative Tolerance (-1), i.e. the value is below:

$$S = |C - |C|_6|_6$$

$$S = |C, C - 1 - |C, C - 1|_6|_6$$

$$S = |C, C - 1 - |C|_6, |C - 1|_6|_6 =$$

$$|C - |C|_6|_6 = 0,$$

$$|C - |C - 1|_6|_6 = 1,$$

$$|C - 1 - |C|_6|_6 = -1 = 5,$$

$$|C - 1 - |C - 1|_6|_6 = 0$$

$$= |[-1,1]|_6 = |-1,0,1|_6$$

$$= \mathbf{5}, \mathbf{0}, \mathbf{1} - \boldsymbol{no\ errors}$$

2. Positive Tolerance (+1), i.e. the value is above:

$$S = |C - |C|_6|_6$$

$$S = |C, C + 1 - |C, C + 1|_6|_6$$

$$S = |C, C + 1 - |C|_6, |C + 1|_6|_6 =$$

$$|C - |C|_6|_6 = 0,$$

$$|C - |C + 1|_6|_6 = -1 = 5,$$

$$|C + 1 - |C|_6|_6 = 1,$$

$$|C + 1 - |C + 1|_6|_6 = 0$$

$$= |[-1,1]|_6 = |-1,0,1|_6$$

$$= \mathbf{5}, \mathbf{0}, \mathbf{1} - \boldsymbol{no\ errors}$$

For the formula stated above, the minimum modulus value changes from six to four. Taking a modulus less than four will mean that there will be no numbers left to represent the erroneous syndrome. To better understand, take an example where the modulus taken was three and the transient featured non overlapping regions.

Ex. **(mod 3) :** $|[-1,1\ ]|_r = 2,0,1$

As you can see in the above example the error free syndrome is the entire number range for modulus three, leaving no numbers within the modulus three range to detect erroneous syndromes.

3.5. Estimating Undetectable Errors using Probability Inequality

As analog inputs are converted to digital or discrete outputs through analog to digital convertors, the problem of undetected errors arises. In the case when the erroneous output produces an error free syndrome, we have undetected errors.

This paper presents the idea that the aliasing rate for converting signals using analog to digital converters, where the output transient is overlapping, the

Aliasing Rate or probability of undetected errors is $\boldsymbol{P_{nd} < 5/r}$

provided that all the error patterns are equally likely or independent.

3.6. Probability Inequality for Estimating Aliasing Rate Proof

The aliasing rate is defined as a ratio between the number of errors that are not detected by the error free syndrome to the total number of possible errors in the pair of m-bit input integers and the integer that is produced by the modulus r. The m bit integer C can have $0 \leq C \leq 2^m - 1$ possible errors while the integer produced by the modulus $|C|_r$ can have $0 \leq |C|_r \leq r-1$ possible errors. We are looking at the total number of possible errors in the pair above, as the denominator of the ratio.

The total number of possible errors in the pair C, $|C|_r$ is $r2^m - 9$. This is due to the fact that the total number of possible combinations is $r2^m$. Since we are looking at the case in which the transient is overlapping (ie. producing a tolerance of ± 1), the m bit integer and modulus r can each have three acceptable values, meaning there are nine total error free combinations. Therefore, the total number of possible errors in the pair C, $|C|_r$ is $r2^m - 9$. In the previous section, we saw that for a tolerance of ± 1 there are five possible error free syndromes (r-2, r-1,0,1,2 where r is your modulus). If any of these error free syndromes is placed in a combination with one of the $r2^m - 9$ possible errors, you will have an undetected error. Therefore, the aliasing rate is $\frac{5*2^m-9}{r*2^m-9}$. Knowing that the $r > 5$ and $m > 0$, you can bound the aliasing rate. The calculations are shown below.

$$-r < -5 \; or \; -r * 9 < -5 * 9$$

$$5r2^m - 9r < 5r2^m - 5 * 9 \; or \; r(5 * 2^m - 9) < 5(r2^m - 9)$$

$$\frac{5*2^m-9}{r2^m-9} < \frac{5}{r}$$

4

Design

4.1. System Design

The complete system was implemented using the idea of error control coding, where extra redundancy bits are added to represent the specific modulus of the input. The complete system featured two Successive Approximation Register Analog to Digital Convertors, where the first SAR ADC converts the input into a digital output. The second converter takes in the input signal and uses extra logic to produce a residue of the input. Each instance of this complete system is unique to a specific modulus. For this specific implementation, a six-bit input was chosen giving an input range of 0 to 63 volts or a total of 64 possible inputs. Given this range, three systems were implemented to represent the following modulus set: {3,5,7}. Figure 16 below shows a high-level diagram of the complete system and its components.

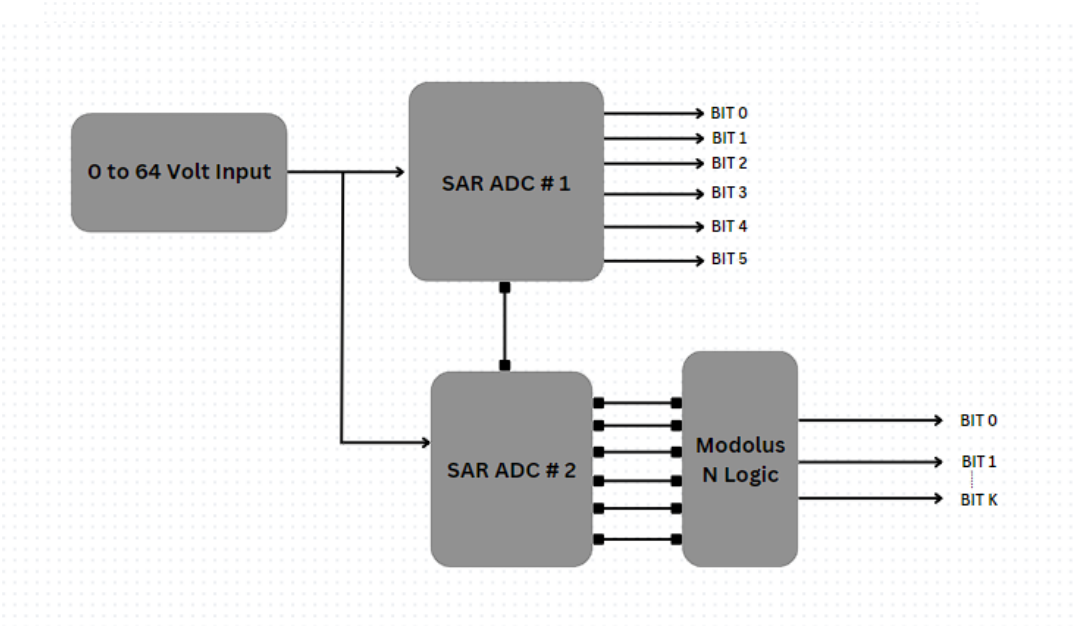


Figure 16. *Proposed ADC Testing Method*

In the figure above, the initial input voltage is fed to both SAR ADC's. SAR ADC #1 takes in the initial input voltage and produces a six-bit digital output. SAR ADC #2 takes in the initial input voltage and produces a residue of the initial input voltage which is shown through the bits produced by the Modulus N Logic. It is important to note that the residue output is a binary representation of the chosen number system and that the number of residue bits varies depending on the modulus chosen. For the Modulus 3 system, the number of residue bits is two. This is due to the fact that the modulus three of any integer can be 0,1 or 2. On the other hand, for the Modulus 5 system, the number of residue bits is three. Finally for the Modulus 7 system, the number of residue bits is also three. In order to test the aliasing rate inequality presented in this paper, the system above was extended to calculate the syndrome.

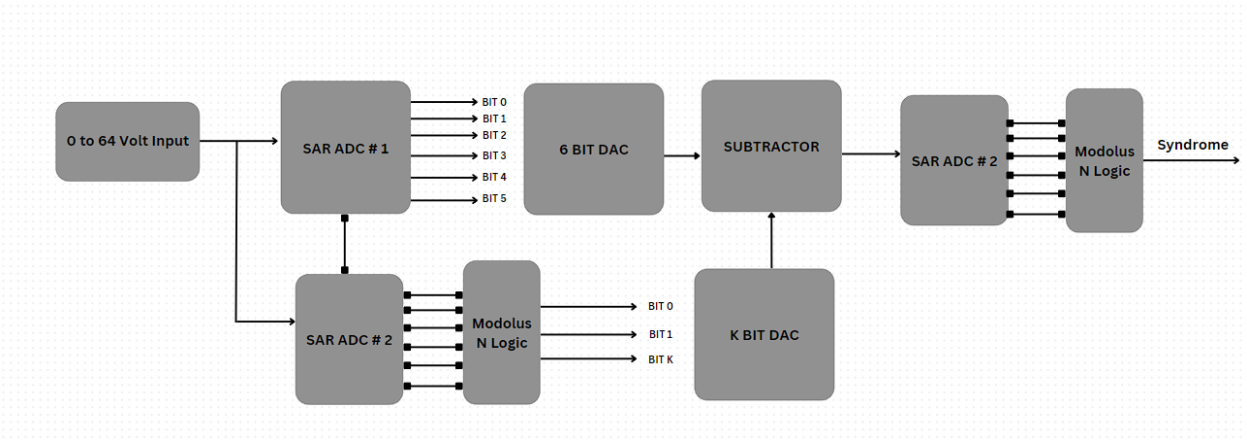


Figure 17. *Proposed ADC Testing Method with Syndrome Calculation*

In the extended structure, the modulus and digital output of the SAR ADCs are converted to analog signals using a binary weighted digital to analog converter. Following the conversion, the signals are subtracted, and the modulus is found using the second SAR ADC and some logic. This output represents the syndrome.

4.2. System Component Breakdown

4.2.1. 6 Bit Successive Approximation ADC

The Successive Approximation Register implemented in this paper is a simplified version. The design used in this approach features the SAR register, DAC and comparator but no sample hold circuit. In addition, the input to the design is a Direct Current Voltage. The full model used in this paper is shown in the figure below (figure 19) including a simplified block diagram (figure 18) of the interactions of each individual component. The comparator used in the construction of the SAR ADC was implemented using a 741 Operational Amplifier. The six-bit SAR ADC implemented can convert numbers in the range of 0 to 63. To operate the digital converter, the circuit is first reset and then using the digital clock, the converter will iterate through the quantization levels till the desired result is found. All 64 quantization levels were tested, and it was confirmed the design was fully functional.

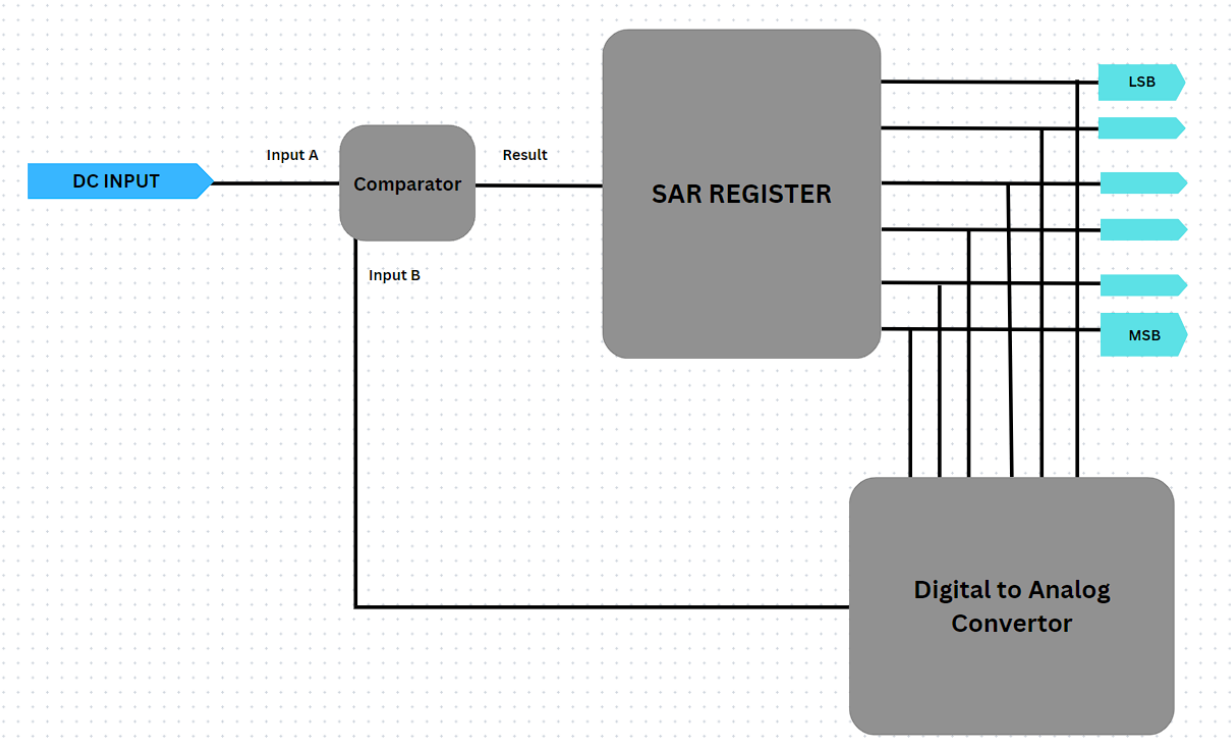


Figure 18.*Successive Approximation Register ADC*

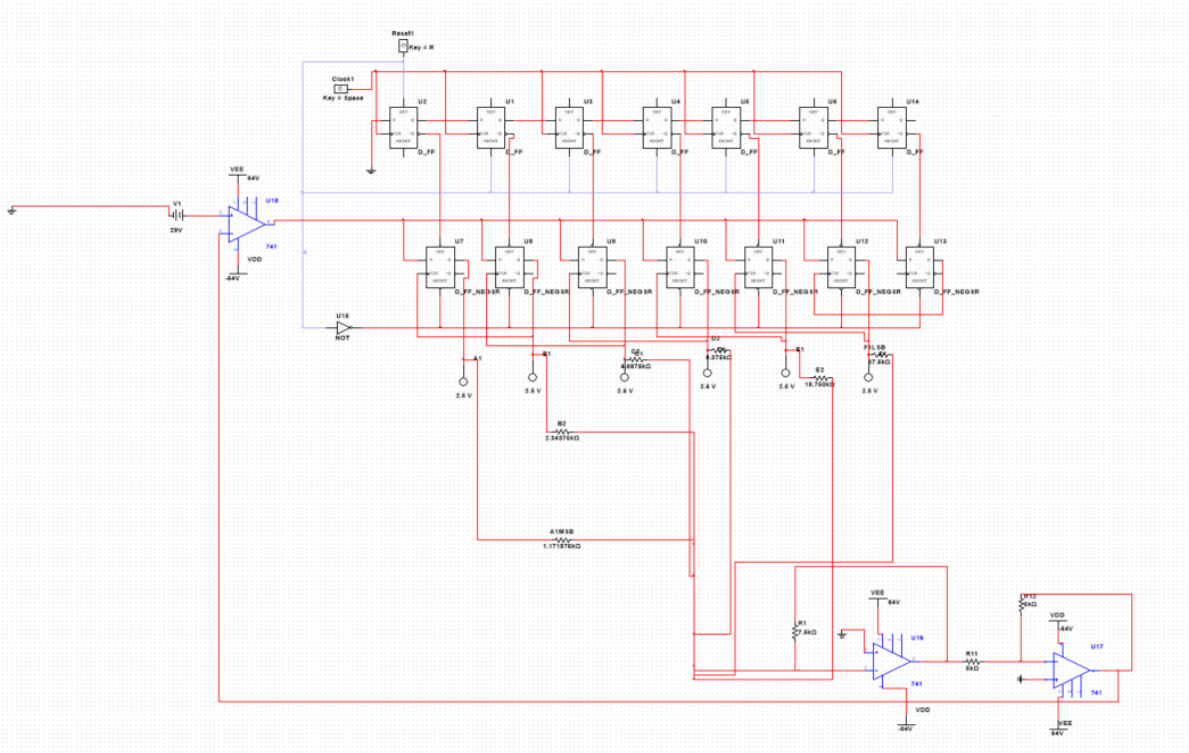


Figure 19. *SAR ADC Circuit Implementation*

4.2.2. Successive Approximation Register (SAR)

The most distinguishing feature of the SAR ADC is the successive approximation register. The Successive approximation register is responsible for providing the output of the SAR ADC. The SAR outputs a certain quantization level depending on the value of the comparator. Depending on the value of the comparator the quantization level will increment the next bit and either change the present bit to one or zero. This quantization level is the result of our analog to digital conversion.

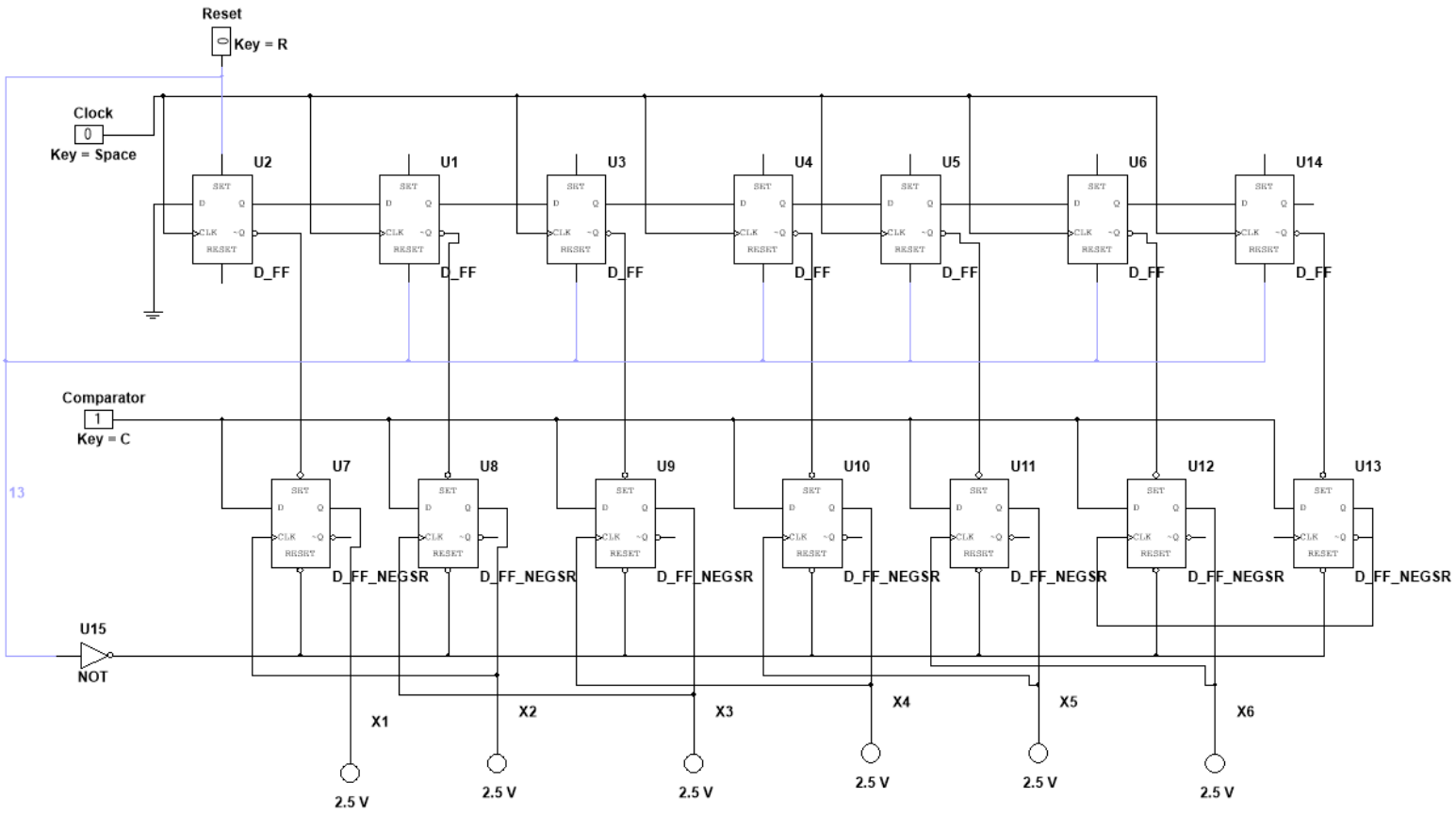


Figure 20. *Successive Approximation Register (SAR)*

The figure above is the complete successive approximation register. The SAR is constructed using a series of D Flip Flops and lightbulbs to represent the binary output (OFF= 0 or ON=1). As mentioned above, the output of the SAR is highly dependent on the value of the comparator. To simulate the comparator, a digital switch was used. Furthermore, a digital switch was also used to simulate the value of the clock and reset switch. If the value of the comparator is one, then the most significant bit is set to 1 and the next bit is set to 1. On the other hand, if the value is zero, then the most significant bit is set to zero and the next bit is set to one. The first step in the process of the Successive Approximation Register is to reset the circuit. Once the circuit is

reset, the first bulb should come on as this is the initial state. This is shown in the image below. It is important to note that in the image below, the comparator is set to one.

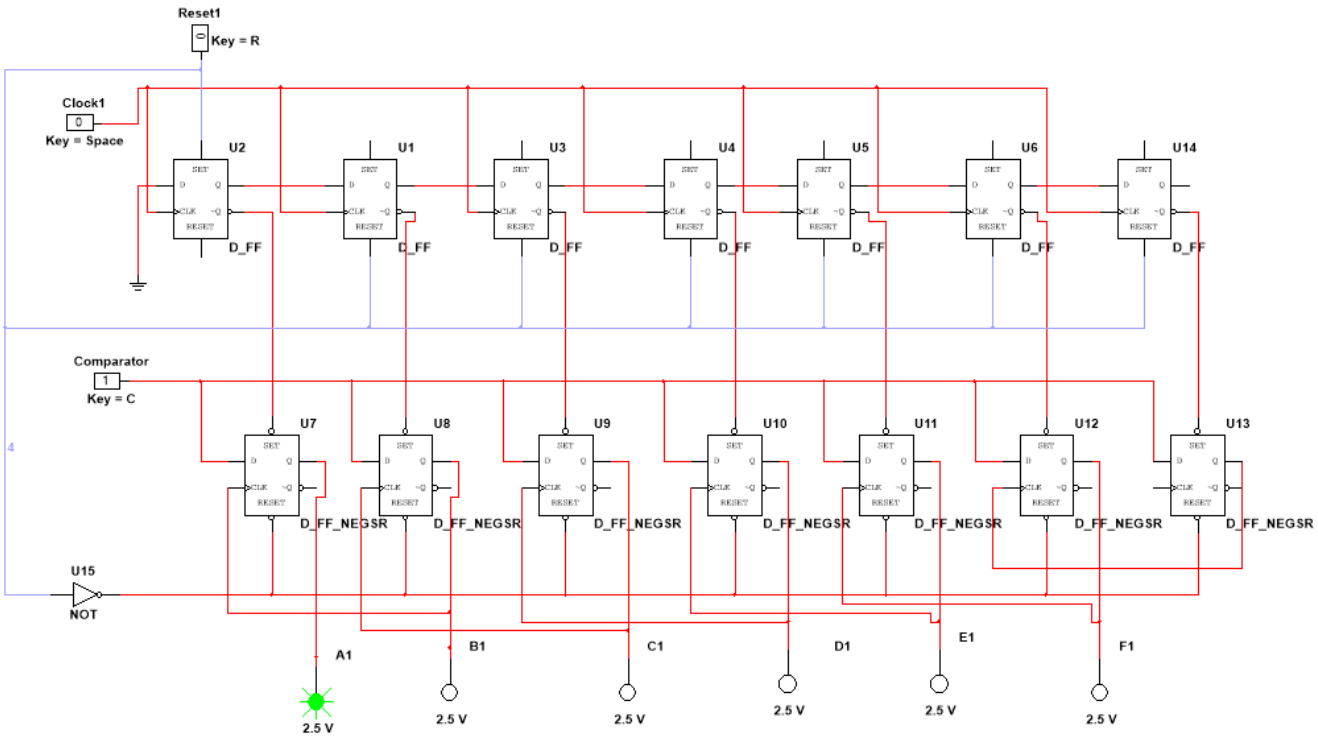


Figure 21. *SAR output when circuit reset, and comparator set to one.*

If the comparator is set to zero, the circuit has been reset and the clock is cycled/simulated, we are presented with a different pattern. Initially after the circuit is reset, the output is the same initial state as shown above whether the comparator is set to one or zero, where only the first bulb is lit. As the clock progresses, the output pattern will be different depending on whether the comparator value was one or zero. Once the clock goes through a cycle, the next output is displayed below. It is important to note that in the image below, the comparator value is zero.

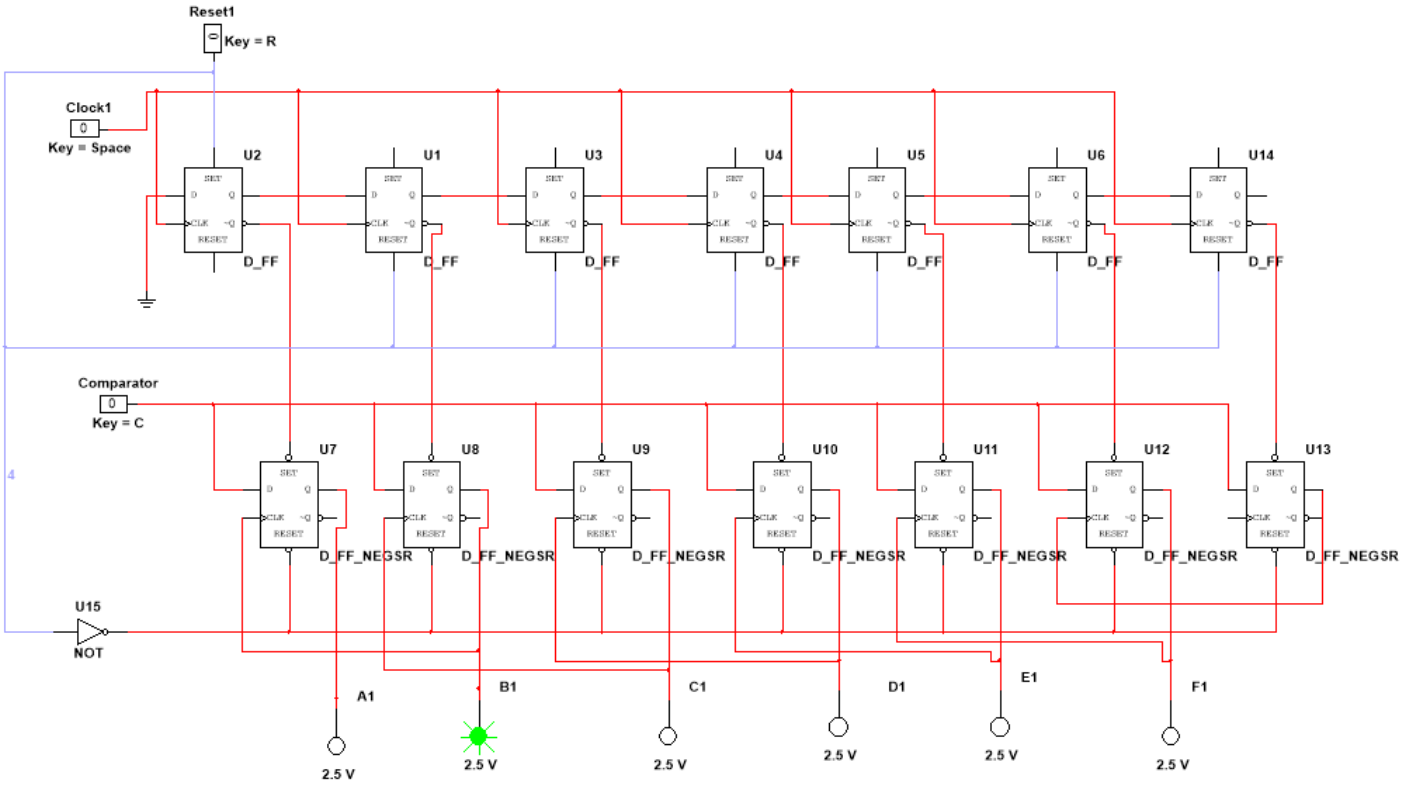


Figure 22.*SAR output when clock simulated from previous state, comparator set to zero.*

In the above, the most significant bit was set to zero (i.e. The bulb turned off) and the next bit was set to one. If we continue cycling/simulating the clock, each MSB will be set to zero while the next bit will be set to one. This will continue until the last bit is set to one (i.e. the bulb is ON), this is shown in the figure below.

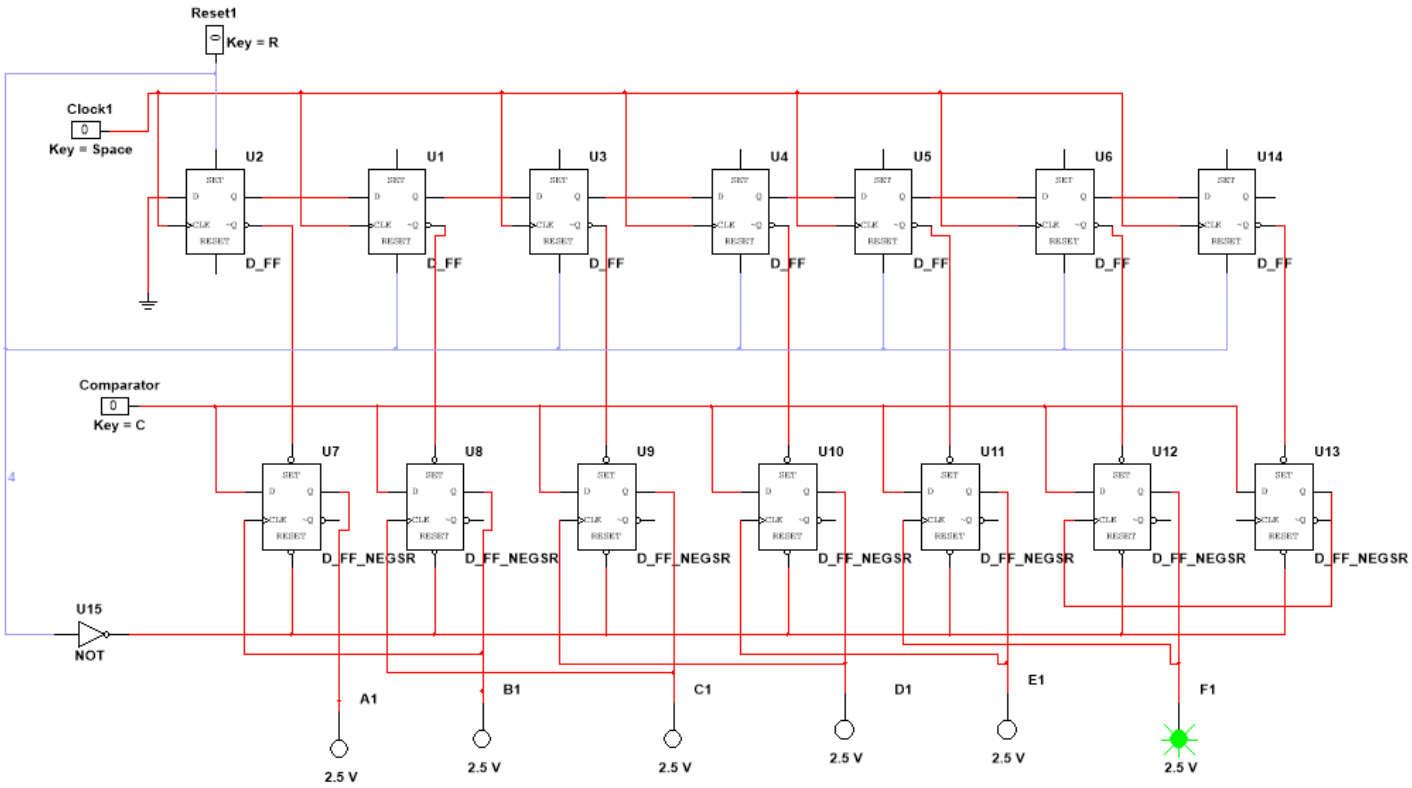


Figure 23*.SAR output when clock simulated to last state with comparator set to zero.*

As mentioned above, if the comparator value is set to one, the Successive Approximation Register will produce a different output pattern. Following the initial state, if the comparator is kept at a value of one and the clock continues cycling, the output will look like the image below.

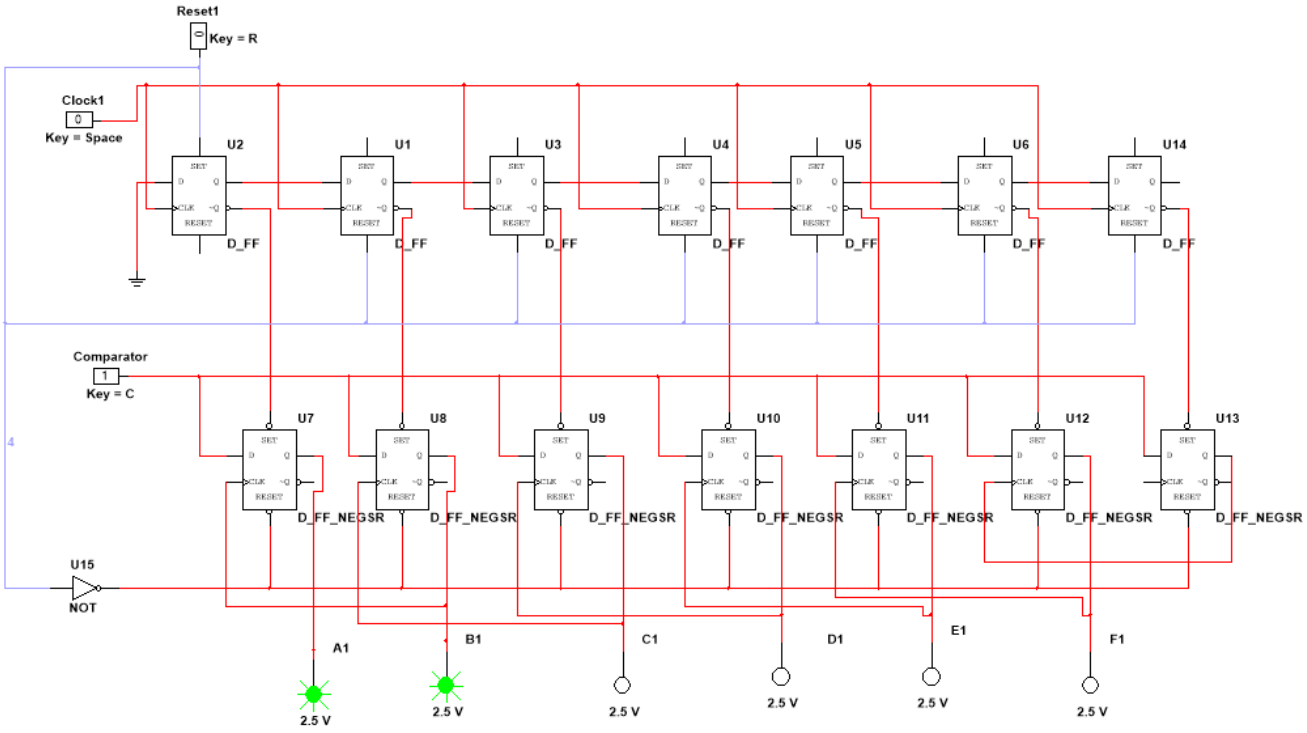


Figure 24*.SAR output when clock simulated from initial state with comparator set to one*

As you can see in the image above, the most significant bit was kept one and the next bit was also then set to one. If we were to keep cycling the clock, the most significant bit along with the next bit would be set to one. This would continue until the last bit is set to one. The final stage can be seen in the image below.

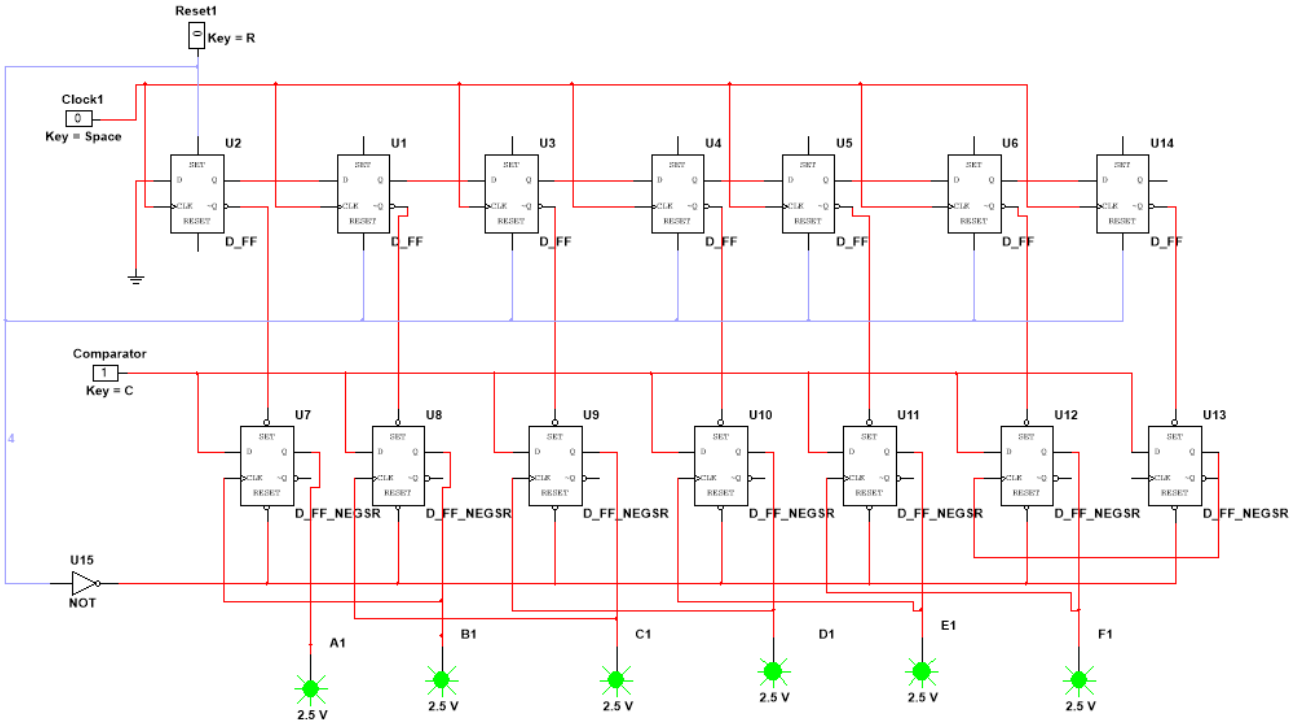


Figure 25.*SAR output when clock simulated to last state with comparator set to one*

4.2.3. Digital to Analog Convertor (DAC):

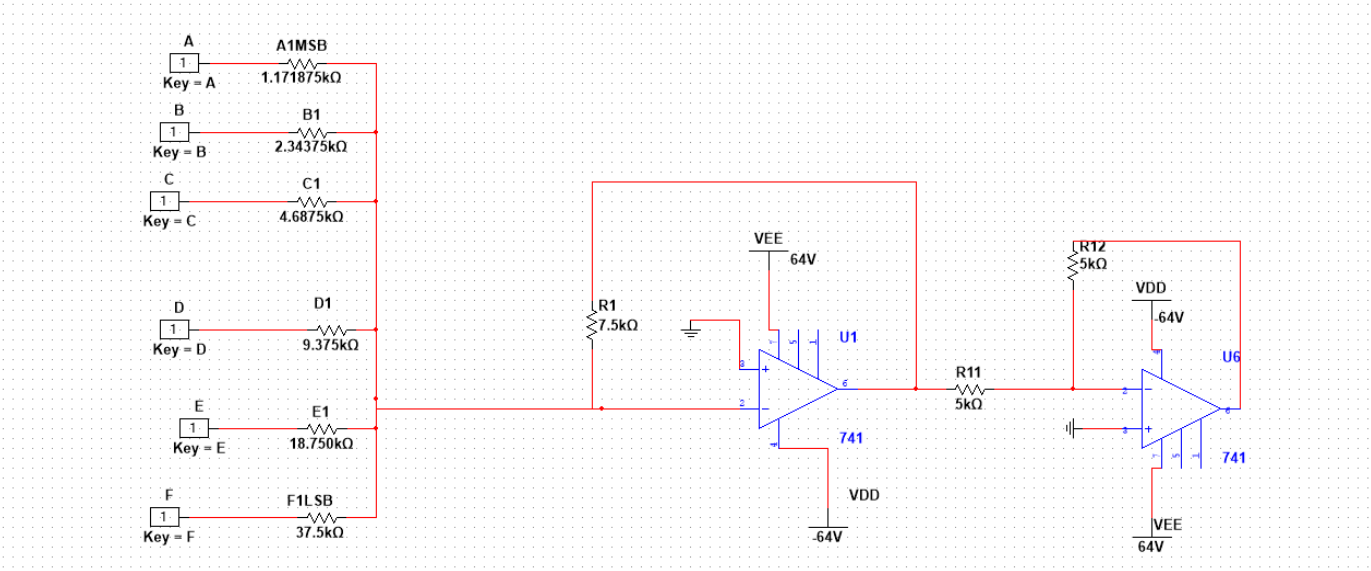


Figure 26.*Ideal Circuit DAC Implementation*

The above is an ideal digital to analog convertor. The Digital to Analog Convertor is made up of a series of resistors and two operational amplifiers. The resistor at the top is the value of the most significant bit (sixth bit) and as you follow the resistors down you reach the least significant bit(zeroth bit). The second operational amplifier is used to invert the signal as the result of the first operational amplifier is a negative voltage.

4.2.3.1. Calculating the values of the resistors

To calculate the values of the resistors, we will start with the top resister (i.e. Most significant bit) and work our way down to the bottom resistor (i.e. Least significant bit). The digital to analog convertor used in this design was specifically a six-bit binary weighted digital to analog convertor. The binary weighted DAC consists of a parallel set of resistors and a feedback resistor. The value of the feedback resistor was chosen arbitrarily. The input to the binary weighted DAC is simulated using the digital interactive constant.

The equation used to calculate the value of the resistors is:

$$Vout = [\frac{R_f}{R_5} * V_5 + \frac{R_f}{R_4} * V_4 + \frac{R_f}{R_3} * V_3 + \frac{R_f}{R_2} * V_2 + \frac{R_f}{R_1} * V_1 + \frac{R_f}{R_0} * V_0]$$

One thing to note is that in the formula above, V_D is 5 volts. In National Instruments Multisim, the value of the Digital Interactive Constant, which provides the logic zero or logic one input has a voltage of five volts. In the equation above R_f represents the feedback resistor while values R_5 to R_0 represent the six input resistors. The digital interactive constants provide a logic one or zero to the input resistors, to determine which resistors will be used in the calculation of our analog signal. Furthermore values V_5 to V_0 represent the input voltages provided to the resistors. Since the interactive digital constants have a set voltage of 5 volts, if they are set to one, they will provide the resistor with a voltage of 5 volts, while if they are set to zero, the resistor will receive an input voltage of zero volts.

Calculation 1:

To understand how the analog signal is calculated, we will look at the first case where the most significant bit is set high, while the remaining are set low. This gives us an input of '100000' which when converted should output approximately 32 volts. As explained above, since the first resistor will receive an input of logic one it will have a voltage of 5 volts, while the other resistors being logic zero, will have an input voltage of zero. Therefore $V_5 = 5\ volts$ while $V_4 \ldots V_0 = 0\ volts$. If we plug this information into the equation above, you get:

$$32 = [\frac{7.5k}{R_5} * 5]$$

Now if you rearrange the equation above to solve for R_5 you obtain 1.1718 KΩ. To calculate the ideal values of all other resistors, repeat the process mentioned above, by feeding the desired resistor an input logic value of one, while keeping the remaining as zero. Using these ideal values for the resistors, we can get an output voltage that is very close to the ideal analog value of our digital input. An example of the output produced by the digital to analog convertor is shown below.

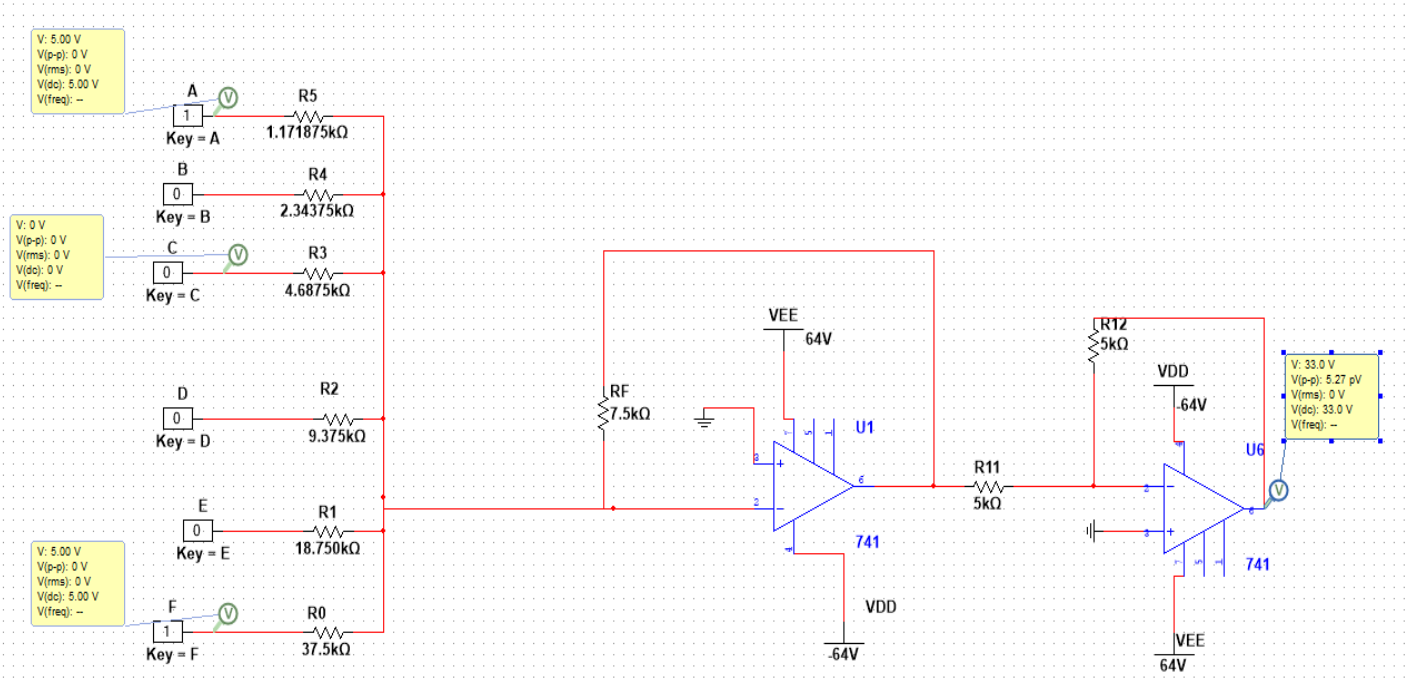


Figure 27.*Example Output of DAC*

In the figure above, we are providing an input value of '100001'. This means that R_5 and R_0 are receiving an input voltage of five volts, while the other resistors are receiving an input of zero volts. This can be seen in the figure above, where the voltage probe indicates a value of 5 volts for resistors R_5 and R_0 while the other resistors as seen by the voltage probe on R_3 have a zero voltage.

5 Results

This paper aimed to introduce a new signature testing method for the Successive Approximation Analog to Digital Convertor and discuss how the aliasing rate of an Analog to Digital Convertor testing system implemented using SAR ADCs can be estimated before transmission using a probability inequality. The proof of the inequality used in estimating and any other calculations required can be found in the previous sections.

5.1. Signature Testing Method using Successive Approximation Analog to Digital Convertors

As previously mentioned, Online BIST testing involves using parity bits to detect errors. The parity bit is generated given the input to the device under test and then compared to the parity bit generated from the device under test, where a mismatch would represent an error. Instead of parity bits, an error detection method known as Cyclic Redundancy Check (CRC) can be used. Essentially with the Cyclic Redundancy Check a remainder is found by dividing the intended message by the polynomial equation. Furthermore, to test the Analog to Digital Convertor, some remainder of the initial input analog signal is used. A detailed explanation of CRC can be found in the Background section above. The foundation of the system introduced in this paper is through the combination of online BIST testing and cyclic redundancy check in place of parity bits. One thing to note is that the cyclic redundancy check is compressing the input signal. The result after compression is the signature and will be used to detect errors. Dual Analog to Digital Convertors can be used to implement a similar method, where the first analog to digital convertor is generating the message while the second ADC convertor is a copy of the first and produces compressed version of the input to be used for error detection or signature analysis. The

compressed “signature” of the input is not ideal as it is a loss of information, and it may not be possible to reproduce the input from the compressed result.

This paper presented the design seen in Figure 17. This design uses two Successive Approximation Analog to Digital Convertors where the second analog to digital converter is not a copy of the first. Instead, the secondary Successive Approximation Register Analog to Digital Convertor produces a remainder of our m bit input and produces a much smaller output. It is important to note that the modulus Analog to Digital Convertor is producing a binary represented modulus. Furthermore, every remainder that is produced for a specific modulus is represented in binary bits. For example, for a m bit input and secondary analog to digital converter that used seven as the modulus, the modular adc will represent the m bit output with three binary bits. As compared to the previous method of using copied analog to digital converter, the method presented in this thesis sees a hardware reduction. This is simply since the modular adc is producing a much smaller output. Furthermore, this method introduces compression or signatures since the input is being converted to a smaller modulus output. With compression and the erroneous operation of analog to digital converters, the issue of undetected error arises.

This is simply because there is a smaller range of values that can be used for error detection. Since the modular analog to digital converter produces a smaller output, there is a lesser range of values that can be used to detect errors in the syndrome. The following section of this work will present a method for estimating the probability of undetected errors and it will be made evident that this probability can be made negligible by introducing a larger modulus. If the resolution of the analog to digital converters increased, the probability of undetected errors will remain the same. As mentioned above, the smaller range leads to undetected errors and therefore the resolution will not increase the undetected errors. The design presented in this thesis can be used for verification of the analog to digital converter itself or for transmission of the signal.

The system in this thesis also introduces a different method to calculate Differential and Integral Non-Linearity. Differential non-linearity (DNL) is essentially calculated by taking the difference of the actual and ideal values. Following this the Integral Non-Linearity (INL) can be taken by calculating the sum of every DNL up to the specific point. As more values are obtained, the value of the INL increases as it is a sum of all the DNL’s up to that specific point, this would

mean a large accumulator is needed. Therefore, this in turn can result in increased hardware and cost. With the method presented in this thesis, it is shown that DNL and INL errors can be calculated in a cheaper and more hardware efficient manner.

Instead of using the ideal and actual outputs of the primary analog to digital converter, the signatures can be used to calculate the INL and DNL errors. The modulus of the discrete value obtained from the primary analog to digital converter is calculated and compared to the signature of the modular ADC. This method is completed using the syndrome, $|\ C\ -\ C\ modX\ |_{modX}$. The syndrome essentially calculates the difference between the modulus of primary ADC output and the modular ADC. The signature of the primary output can be considered the "actual output". On the other hand, the output from the modular adc can be considered the "ideal" value as it is being used to test the primary ADC. Since the signatures are being used to obtain the INL error, a hardware reduction is seen as the storage needed now is $INLmodR$ where R is the modulus being used to test the system.

With current testing methods, the following process is followed: Apply input to circuit under test, compress result and obtain signature, feed signature to output response analyzer and determine if the result matches the expected signature. With this method, it is hard to determine where the error has occurred. It is difficult to identify if the error occurred within the analog to digital converter itself or at the output before being sent to the output response analyzer. The proposed architecture also provides the ability for testing to be done every clock cycle. Through this it can be seen at which point of the quantization process an error occurs, allowing one to know where the error may have arisen once detected at the output. The architecture presented in this paper calculates the syndrome at every stage of the quantization process or at every clock cycle. From this it can be determined if any stage in the conversion process has an error by checking the calculated syndrome.

5.1. Estimating Aliasing rate using the presented architecture

To summarize this paper aims to state the following:

1. Given a tolerance of ± 1 the probability of undetected errors or aliasing rate when testing a m bit integer by a mod r syndrome is $Pnd < 5/r$. The mod r must be greater than 5.
2. Given a tolerance of 0 you obtain the discrete adder case the probability of undetected errors or aliasing rate when testing with an m bit integer and mod r syndrome is $Pnd < 1/r$. The mod r must be greater than 1.
3. The error free syndromes for a mod r and m bit integer are r-2, r-1,0,1,2.

The extended architecture mentioned in the above section was used to test the above statements. A six-bit input signal (0 to 63 volts) and a modulus seven was chosen for testing. The input signal of choice for this specific test was 37. Given a tolerance of ± 1 we are expected to have an aliasing rate that is less than $5/7$. To test these errors were thrown into the system and the output syndromes were observed. Every possible combination of erroneous signals in both the input and modulus were introduced and the corresponding syndromes were recorded in a table, which can be seen in the Appendix.

From the statements presented in this paper the following information can be obtained using the extended architecture and an input signal of 37 Volts.

1. $P_{nd} < 5/7 = 0.71$
2. Given a modulus of 7, the error free syndromes are:
    a. $r - 2 = 5$
    b. $r - 1 = 6$
    c. 0
    d. 1
    e. 2
3. Given the tolerance of ± 1, there will be 9 error free combinations which will produce syndromes from the list mentioned above. Any combination that does not produce one of the syndromes above can be considered a detected erroneous combination. For this specific case, a syndrome of three or four will be considered erroneous.
4. The total number of erroneous combinations present are $r * 2^m - 9$, $(7 * 2^6) - 9 = 439$

5. The total number of undetected error combinations present are $5 * 2^m - 9$, $(5 * 2^6) - 9 = 311$

Using Table A, the results above can be confirmed. Given the table the total total erroneous combinations were counted. Given an input signal of 37 and a modulus of 7, the result of the modulus would be 2. Due to the $\pm$ 1 our input can be any of the following: 36,37,38. On the other hand due to tolerance the correct modulus can be: 1,2,3. Therefore the nine possible error free combinations are:

1. (36,1), Syndrome = 0
2. (36,2), Syndrome = 6
3. (36,3), Syndrome = 5
4. (37,1), Syndrome = 1
5. (37,2), Syndrome = 0
6. (37,3), Syndrome = 6
7. (38,1), Syndrome = 2
8. (38,2), Syndrome = 1
9. (38,3), Syndrome = 0

Knowing the above combinations, the total number of erroneous combinations can be counted. These are any remaining combinations that produce a syndrome of 0,1,2,3,4,5,6. This resulted in a total of 439 possible erroneous combinations. From these 439 combinations the undetected combinations were ones whose syndrome was 0,1,2,5,6. These were combinations whose input or modulus signals had errors but produced an error free syndrome. A total of 311 combinations produced undetectable errors. Using this information the aliasing rate or undetected error was calculated as following : $aliasing\ rate\ = number\ of\ undetected\ errors\ /\ total\ number\ of\ errors$

$$aliasing\ rate\ = 311/439$$

$$aliasing\ rate\ \simeq\ 0.71$$

Therefore, the probability inequality presented in this paper matched that of the results calculated using the extended architecture.

As mentioned before the modulus (mod r) must be greater than 5. A modulus less than five will result in no error free syndromes. This means no combination will be identified as an error based on the syndrome, giving a 100% aliasing rate. To confirm the statements presented in this paper, an input of 37 and a modulus of 5 is used. Using the extended architecture and all possible erroneous combinations, a table was constructed which contains the input, modulus 7 and syndrome. Table B can be found in the tables section below. Once again due to the tolerances, an input of 37 can be any of the following values: 36,37,38. Furthermore a modulus 5 can be any of the following: 1,2,3.

1. $r - 2 = 3$
2. $r - 1 = 4$
3. 0
4. 1
5. 2

Given the tolerances, there are nine possible error free combinations, these are listed below. The error free combinations were passed through the extended architecture and their syndrome was found and placed in the table. The list below contains the error free combinations along with their syndromes.

1. (36,1), Syndrome = 0
2. (36,2), Syndrome = 4
3. (36,3), Syndrome = 3
4. (37,1), Syndrome = 1
5. (37,2), Syndrome = 0
6. (37,3), Syndrome = 4
7. (38,1), Syndrome = 2

8. (38,2), Syndrome = 1
9. (38,3), Syndrome = 0

Once the error free combinations were passed, every possible erroneous combination was passed through the architecture and once again the syndrome was found and recorded in Table B. From this table, it was determined that there were 311 possible erroneous combinations, of which all 311 combinations were undetected errors, ie. resulted in error free syndromes.

Therefore, the aliasing rate was calculated as follows:

$$aliasing\ rate\ = number\ of\ undetected\ errors\ /\ total\ number\ of\ errors$$

$$aliasing\ rate\ = 311/311$$

$$aliasing\ rate\ = 1.0$$

Using the statements presented in this paper the following information was derived:

1. $P_{nd}\ < 5/5\ = 1.00$
2. Zero error free syndromes.
3. The total number of erroneous combinations present are $r * 2^m - 9$, $(5 * 2^6) - 9\ = 311$
4. The total number of undetected error combinations present are $5 * 2^m - 9$, $(5 * 2^6) - 9\ = 311$

As you can see using the method presented in this paper, the aliasing rate was seen to be approximately less than 1.0 which is the same value as calculated using the extended architecture. It can be seen from either method, that using a modulus of less than 5 would result

in 311 possible erroneous combinations of which all 311 are seen to be undetected errors. Every possible erroneous combination would result in an error free syndrome.

Therefore, the inequality presented in this paper is a useful tool to estimate the aliasing rate that may occur given the modulus chosen. It is important the modulus be greater than 5 to ensure that errors can be properly detected and in order to minimize the aliasing rate, a large modulus should be used.

6 Conclusion

6.1. Summary

Analog to digital converters have played an integral role in the evolution of integrated circuits. ADCs are responsible for the conversion of different continuous time signals and therefore it is crucial that an error free discrete signal is produced . There are many different analogs to digital convertor architectures, each with different characteristics and use cases. To ensure an accurate, error free output, Analog to Digital Convertors are often paired with error detection techniques to identify potential errors that may occur. Cyclic Redundancy Check is a common error control coding technique which essentially uses the modulus of the intended message to detect errors. Built-in self-testing is a common ADC testing method which essentially moves testing functions to on chip and may use on chip hardware to implement testing. An important factor of Built in Self testing is that there is no need to place the system into a test mode, all testing can be completed during normal operation.

This thesis presented the following major ideas:

1. Generalizing Syndromes for a c bit input and modulus r.
2. Signature Testing method using modulus and two Successive Approximation Analog to Digital Convertors.
3. Using syndrome to calculate INL and DNL error.
4. Estimating the Probability of Undetected Errors or Aliasing.

The first idea presented in this paper was the generalization of the syndrome. The syndrome can be calculated using the following expression, $Syndrome = | C - C \, modX |_{modX}$. It was seen

that as the input C is varied and the modulus is kept constant, the syndrome or vector containing error free values did not change. This led to the following generalization of the syndrome:

General Case (mod r): $|[-2,2]|_r = r-2,r-2,0,1,2$ where r is the chosen modulus

It is important to note that for the above generalization to be true, two propositions must be stated. Firstly, the modulus r being used must be greater than five. If the modulus being used is less than six, there will be no value that can be used for error detection. Secondly, the transients must be overlapping so the tolerance is present.

Secondly this paper presented a new Signature testing method using two Successive Approximation Register (SAR) Analog to Digital Convertors. This method uses Online BIST Testing and Cyclic Redundancy Check as its foundation. By replacing the parity check method featured in online testing with an error control coding technique featuring modulus, a new method was presented. The first SAR converts the input signal to discrete form, while the second SAR calculates the remainder of our input signal. Furthermore, this system was extended to calculate the syndrome of the remainder and discrete output, allowing all stages of conversion to be tested for errors.

Thirdly this paper introduced a modified version of calculating the Differential and Integral Non-Linearity Error. Using signatures, the INL and DNL can be calculated. Differential non-linearity is the difference between the actual and ideal output, while the Integral Non-Linearity is essentially a sum of these values. If the primary analog to digital convertor is considered the "actual output", the modolus can be calculated and compared to the output of the modular adc which can be considered the "ideal output". This signature method of calculating the INL and DNL error can result in a hardware reduction as INL is being calculated using modolus instead of its full value, so smaller values are being summed.

Finally, since the modular analog to digital convertor is compressing the output, a lesser range of values is being used to detect errors hence undetected errors arise. Aliasing or the probability of undetected error is a ratio between the number of undetected errors and the total number of errors. To estimate Aliasing, the following inequality can be used:

Aliasing or probability of undetected errors is $P_{nd} < 5/r$

This inequality was proven using the dual Successive Approximation Register ADC system mentioned above. By selecting an input and calculating the corresponding outputs along with all possible errors, a table was created. The aliasing rate was calculated by counting the results found in the table and it was seen to match the inequality presented in this paper.

6.2. Future Work

A benefit of using residue mathematics and remainders is the ability to reconstruct the original signal. This approach can be extended to the thesis. It was seen that by using a secondary modular ADC to complete testing a hardware reduction can be obtained since the converter is not a copy but instead obtains the modolus and produces a smaller output. Instead of having a high-resolution primary analog to digital converter, parallel modolus analog to digital converters can be used. The modolus parallel analog to digital converters can be used to reconstruct the signal and therefore the need for a larger more expensive analog to digital converter is no longer needed. In addition to these ADC's an extra modolus ADC would be needed to implement error control coding like the methodology presented in this paper. The work in this thesis presented a method for error detection, however error correction must be considered. With the possibility of errors arising, the need for correction is significant. The system presented in this paper would need to be analyzed and an ideal method for error correction would need to be introduced. A method of attaching redundant data such as error correction code would need to be further studied and a suitable approach to introduce the redundant data to this system would need to be found.

TABLES

A Table A - 7 Bit Input with Mod 7

Table A is a record of every possible input of 7 bits and mod 7 combination that can be applied to the presented system. Furthermore, the syndrome for each combination is calculated using the design and recorded in the table.

Combination	Input	Mod7	Syndrome
0	0	0	0
1	0	1	6
2	0	2	5
3	0	3	4
4	0	4	3
5	0	5	2
6	0	6	1
7	1	0	1
8	1	1	0
9	1	2	6
10	1	3	5
11	1	4	4
12	1	5	3
13	1	6	2
14	2	0	2
15	2	1	1
16	2	2	0
17	2	3	6
18	2	4	5
19	2	5	4
20	2	6	3
21	3	0	3
22	3	1	2
23	3	2	1
24	3	3	0
25	3	4	6
26	3	5	5
27	3	6	4
28	4	0	4
29	4	1	3

Combination	Input	Mod7	Syndrome
30	4	2	2
31	4	3	1
32	4	4	0
33	4	5	6
34	4	6	5
35	5	0	5
36	5	1	4
37	5	2	3
38	5	3	2
39	5	4	1
40	5	5	0
41	5	6	6
42	6	0	6
43	6	1	5
44	6	2	4
45	6	3	3
46	6	4	2
47	6	5	1
48	6	6	0
49	7	0	0
50	7	1	6
51	7	2	5
52	7	3	4
53	7	4	3
54	7	5	2
55	7	6	1
56	8	0	1
57	8	1	0
58	8	2	6
59	8	3	5
60	8	4	4
61	8	5	3
62	8	6	2
63	9	0	2
64	9	1	1
65	9	2	0

Combination	Input	Mod7	Syndrome
66	9	3	6
67	9	4	5
68	9	5	4
69	9	6	3
70	10	0	3
71	10	1	2
72	10	2	1
73	10	3	0
74	10	4	6
75	10	5	5
76	10	6	4
77	11	0	4
78	11	1	3
79	11	2	2
80	11	3	1
81	11	4	0
82	11	5	6
83	11	6	5
84	12	0	5
85	12	1	4
86	12	2	3
87	12	3	2
88	12	4	1
89	12	5	0
90	12	6	6
91	13	0	6
92	13	1	5
93	13	2	4
94	13	3	3
95	13	4	2
96	13	5	1
97	13	6	0
98	14	0	0
99	14	1	6
100	14	2	5
101	14	3	4

Combination	Input	Mod7	Syndrome
102	14	4	3
103	14	5	2
104	14	6	1
105	15	0	1
106	15	1	0
107	15	2	6
108	15	3	5
109	15	4	4
110	15	5	3
111	15	6	2
112	16	0	2
113	16	1	1
114	16	2	0
115	16	3	6
116	16	4	5
117	16	5	4
118	16	6	3
119	17	0	3
120	17	1	2
121	17	2	1
122	17	3	0
123	17	4	6
124	17	5	5
125	17	6	4
126	18	0	4
127	18	1	3
128	18	2	2
129	18	3	1
130	18	4	0
131	18	5	6
132	18	6	5
133	19	0	5
134	19	1	4
135	19	2	3
136	19	3	2
137	19	4	1

Combination	Input	Mod7	Syndrome
138	19	5	0
139	19	6	6
140	20	0	6
141	20	1	5
142	20	2	4
143	20	3	3
144	20	4	2
145	20	5	1
146	20	6	0
147	21	0	0
148	21	1	6
149	21	2	5
150	21	3	4
151	21	4	3
152	21	5	2
153	21	6	1
154	22	0	1
155	22	1	0
156	22	2	6
157	22	3	5
158	22	4	4
159	22	5	3
160	22	6	2
161	23	0	2
162	23	1	1
163	23	2	0
164	23	3	6
165	23	4	5
166	23	5	4
167	23	6	3
168	24	0	3
169	24	1	2
170	24	2	1
171	24	3	0
172	24	4	6
173	24	5	5

Combination	Input	Mod7	Syndrome
174	24	6	4
175	25	0	4
176	25	1	3
177	25	2	2
178	25	3	1
179	25	4	0
180	25	5	6
181	25	6	5
182	26	0	5
183	26	1	4
184	26	2	3
185	26	3	2
186	26	4	1
187	26	5	0
188	26	6	6
189	27	0	6
190	27	1	5
191	27	2	4
192	27	3	3
193	27	4	2
194	27	5	1
195	27	6	0
196	28	0	0
197	28	1	6
198	28	2	5
199	28	3	4
200	28	4	3
201	28	5	2
202	28	6	1
203	29	0	1
204	29	1	0
205	29	2	6
206	29	3	5
207	29	4	4
208	29	5	3
209	29	6	2

Combination	Input	Mod7	Syndrome
210	30	0	2
211	30	1	1
212	30	2	0
213	30	3	6
214	30	4	5
215	30	5	4
216	30	6	3
217	31	0	3
218	31	1	2
219	31	2	1
220	31	3	0
221	31	4	6
222	31	5	5
223	31	6	4
224	32	0	4
225	32	1	3
226	32	2	2
227	32	3	1
228	32	4	0
229	32	5	6
230	32	6	5
231	33	0	5
232	33	1	4
233	33	2	3
234	33	3	2
235	33	4	1
236	33	5	0
237	33	6	6
238	34	0	6
239	34	1	5
240	34	2	4
241	34	3	3
242	34	4	2
243	34	5	1
244	34	6	0
245	35	0	0

Combination	Input	Mod7	Syndrome
246	35	1	6
247	35	2	5
248	35	3	4
249	35	4	3
250	35	5	2
251	35	6	1
252	36	0	1
253	36	1	0
254	36	2	6
255	36	3	5
256	36	4	4
257	36	5	3
258	36	6	2
259	37	0	2
260	37	1	1
261	37	2	0
262	37	3	6
263	37	4	5
264	37	5	4
265	37	6	3
266	38	0	3
267	38	1	2
268	38	2	1
269	38	3	0
270	38	4	6
271	38	5	5
272	38	6	4
273	39	0	4
274	39	1	3
275	39	2	2
276	39	3	1
277	39	4	0
278	39	5	6
279	39	6	5
280	40	0	5
281	40	1	4

Combination	Input	Mod7	Syndrome
282	40	2	3
283	40	3	2
284	40	4	1
285	40	5	0
286	40	6	6
287	41	0	6
288	41	1	5
289	41	2	4
290	41	3	3
291	41	4	2
292	41	5	1
293	41	6	0
294	42	0	0
295	42	1	6
296	42	2	5
297	42	3	4
298	42	4	3
299	42	5	2
300	42	6	1
301	43	0	1
302	43	1	0
303	43	2	6
304	43	3	5
305	43	4	4
306	43	5	3
307	43	6	2
308	44	0	2
309	44	1	1
310	44	2	0
311	44	3	6
312	44	4	5
313	44	5	4
314	44	6	3
315	45	0	3
316	45	1	2
317	45	2	1

Combination	Input	Mod7	Syndrome
318	45	3	0
319	45	4	6
320	45	5	5
321	45	6	4
322	46	0	4
323	46	1	3
324	46	2	2
325	46	3	1
326	46	4	0
327	46	5	6
328	46	6	5
329	47	0	5
330	47	1	4
331	47	2	3
332	47	3	2
333	47	4	1
334	47	5	0
335	47	6	6
336	48	0	6
337	48	1	5
338	48	2	4
339	48	3	3
340	48	4	2
341	48	5	1
342	48	6	0
343	49	0	0
344	49	1	6
345	49	2	5
346	49	3	4
347	49	4	3
348	49	5	2
349	49	6	1
350	50	0	1
351	50	1	0
352	50	2	6
353	50	3	5

Combination	Input	Mod7	Syndrome
354	50	4	4
355	50	5	3
356	50	6	2
357	51	0	2
358	51	1	1
359	51	2	0
360	51	3	6
361	51	4	5
362	51	5	4
363	51	6	3
364	52	0	3
365	52	1	2
366	52	2	1
367	52	3	0
368	52	4	6
369	52	5	5
370	52	6	4
371	53	0	4
372	53	1	3
373	53	2	2
374	53	3	1
375	53	4	0
376	53	5	6
377	53	6	5
378	54	0	5
379	54	1	4
380	54	2	3
381	54	3	2
382	54	4	1
383	54	5	0
384	54	6	6
385	55	0	6
386	55	1	5
387	55	2	4
388	55	3	3
389	55	4	2

Combination	Input	Mod7	Syndrome
390	55	5	1
391	55	6	0
392	56	0	0
393	56	1	6
394	56	2	5
395	56	3	4
396	56	4	3
397	56	5	2
398	56	6	1
399	57	0	1
400	57	1	0
401	57	2	6
402	57	3	5
403	57	4	4
404	57	5	3
405	57	6	2
406	58	0	2
407	58	1	1
408	58	2	0
409	58	3	6
410	58	4	5
411	58	5	4
412	58	6	3
413	59	0	3
414	59	1	2
415	59	2	1
416	59	3	0
417	59	4	6
418	59	5	5
419	59	6	4
420	60	0	4
421	60	1	3
422	60	2	2
423	60	3	1
424	60	4	0
425	60	5	6

Combination	Input	Mod7	Syndrome
426	60	6	5
427	61	0	5
428	61	1	4
429	61	2	3
430	61	3	2
431	61	4	1
432	61	5	0
433	61	6	6
434	62	0	6
435	62	1	5
436	62	2	4
437	62	3	3
438	62	4	2
439	62	5	1
440	62	6	0
441	63	0	0
442	63	1	6
443	63	2	5
444	63	3	4
445	63	4	3
446	63	5	2
447	63	6	1

B

Table B - 7 Bit Input with Mod 5

Combination	Input	Mod5	Syndrome
0	0	0	0
1	0	1	4
2	0	2	3
3	0	3	2
4	0	4	1
5	1	0	1
6	1	1	0
7	1	2	4
8	1	3	3
9	1	4	2
10	2	0	2
11	2	1	1
12	2	2	0
13	2	3	4
14	2	4	3
15	3	0	3
16	3	1	2
17	3	2	1
18	3	3	0
19	3	4	4

Combination	Input	Mod5	Syndrome
20	4	0	4
21	4	1	3
22	4	2	2
23	4	3	1
24	4	4	0
25	5	0	0
26	5	1	4
27	5	2	3
28	5	3	2
29	5	4	1
30	6	0	1
31	6	1	0
32	6	2	4
33	6	3	3
34	6	4	2
35	7	0	2
36	7	1	1
37	7	2	0
38	7	3	4
39	7	4	3

Combination	Input	Mod5	Syndrome
40	8	0	3
41	8	1	2
42	8	2	1
43	8	3	0
44	8	4	4
45	9	0	4
46	9	1	3
47	9	2	2
48	9	3	1
49	9	4	0
50	10	0	0
51	10	1	4
52	10	2	3
53	10	3	2
54	10	4	1
55	11	0	1
56	11	1	0
57	11	2	4
58	11	3	3
59	11	4	2
60	12	0	2
61	12	1	1
62	12	2	0
63	12	3	4
64	12	4	3
65	13	0	3
66	13	1	2
67	13	2	1
68	13	3	0
69	13	4	4
70	14	0	4
71	14	1	3
72	14	2	2
73	14	3	1
74	14	4	0
75	15	0	0
76	15	1	4
77	15	2	3
78	15	3	2
79	15	4	1

Combination	Input	Mod5	Syndrome
80	16	0	1
81	16	1	0
82	16	2	4
83	16	3	3
84	16	4	2
85	17	0	2
86	17	1	1
87	17	2	0
88	17	3	4
89	17	4	3
90	18	0	3
91	18	1	2
92	18	2	1
93	18	3	0
94	18	4	4
95	19	0	4
96	19	1	3
97	19	2	2
98	19	3	1
99	19	4	0
100	20	0	0
101	20	1	4
102	20	2	3
103	20	3	2
104	20	4	1
105	21	0	1
106	21	1	0
107	21	2	4
108	21	3	3
109	21	4	2
110	22	0	2
111	22	1	1
112	22	2	0
113	22	3	4
114	22	4	3
115	23	0	3
116	23	1	2
117	23	2	1
118	23	3	0
119	23	4	4

Combination	Input	Mod5	Syndrome
120	24	0	4
121	24	1	3
122	24	2	2
123	24	3	1
124	24	4	0
125	25	0	0
126	25	1	4
127	25	2	3
128	25	3	2
129	25	4	1
130	26	0	1
131	26	1	0
132	26	2	4
133	26	3	3
134	26	4	2
135	27	0	2
136	27	1	1
137	27	2	0
138	27	3	4
139	27	4	3
140	28	0	3
141	28	1	2
142	28	2	1
143	28	3	0
144	28	4	4
145	29	0	4
146	29	1	3
147	29	2	2
148	29	3	1
149	29	4	0
150	30	0	0
151	30	1	4
152	30	2	3
153	30	3	2
154	30	4	1
155	31	0	1
156	31	1	0
157	31	2	4
158	31	3	3
159	31	4	2

Combination	Input	Mod5	Syndrome
160	32	0	2
161	32	1	1
162	32	2	0
163	32	3	4
164	32	4	3
165	33	0	3
166	33	1	2
167	33	2	1
168	33	3	0
169	33	4	4
170	34	0	4
171	34	1	3
172	34	2	2
173	34	3	1
174	34	4	0
175	35	0	0
176	35	1	4
177	35	2	3
178	35	3	2
179	35	4	1
180	36	0	1
181	36	1	0
182	36	2	4
183	36	3	3
184	36	4	2
185	37	0	2
186	37	1	1
187	37	2	0
188	37	3	4
189	37	4	3
190	38	0	3
191	38	1	2
192	38	2	1
193	38	3	0
194	38	4	4
195	39	0	4
196	39	1	3
197	39	2	2
198	39	3	1
199	39	4	0

Combination	Input	Mod5	Syndrome
200	40	0	0
201	40	1	4
202	40	2	3
203	40	3	2
204	40	4	1
205	41	0	1
206	41	1	0
207	41	2	4
208	41	3	3
209	41	4	2
210	42	0	2
211	42	1	1
212	42	2	0
213	42	3	4
214	42	4	3
215	43	0	3
216	43	1	2
217	43	2	1
218	43	3	0
219	43	4	4
220	44	0	4
221	44	1	3
222	44	2	2
223	44	3	1
224	44	4	0
225	45	0	0
226	45	1	4
227	45	2	3
228	45	3	2
229	45	4	1
230	46	0	1
231	46	1	0
232	46	2	4
233	46	3	3
234	46	4	2
235	47	0	2
236	47	1	1
237	47	2	0
238	47	3	4
239	47	4	3

Combination	Input	Mod5	Syndrome
240	48	0	3
241	48	1	2
242	48	2	1
243	48	3	0
244	48	4	4
245	49	0	4
246	49	1	3
247	49	2	2
248	49	3	1
249	49	4	0
250	50	0	0
251	50	1	4
252	50	2	3
253	50	3	2
254	50	4	1
255	51	0	1
256	51	1	0
257	51	2	4
258	51	3	3
259	51	4	2
260	52	0	2
261	52	1	1
262	52	2	0
263	52	3	4
264	52	4	3
265	53	0	3
266	53	1	2
267	53	2	1
268	53	3	0
269	53	4	4
270	54	0	4
271	54	1	3
272	54	2	2
273	54	3	1
274	54	4	0
275	55	0	0
276	55	1	4
277	55	2	3
278	55	3	2
279	55	4	1

Combination	Input	Mod5	Syndrome
280	56	0	1
281	56	1	0
282	56	2	4
283	56	3	3
284	56	4	2
285	57	0	2
286	57	1	1
287	57	2	0
288	57	3	4
289	57	4	3
290	58	0	3
291	58	1	2
292	58	2	1
293	58	3	0
294	58	4	4
295	59	0	4
296	59	1	3
297	59	2	2
298	59	3	1
299	59	4	0
300	60	0	0
301	60	1	4
302	60	2	3
303	60	3	2
304	60	4	1
305	61	0	1
306	61	1	0
307	61	2	4
308	61	3	3
309	61	4	2
310	62	0	2
311	62	1	1
312	62	2	0
313	62	3	4
314	62	4	3
315	63	0	3
316	63	1	2
317	63	2	1
318	63	3	0
319	63	4	4

Table B is a record of every possible input of 7 bits and mod 5 combination that can be applied to the presented system. Furthermore, the syndrome for each combination is calculated using the design and recorded in the table.

References

[1] A. Benso, S. Chinsano, G. Di Natale, P. Prinetto and M. L. Bodoni, "Online and offline BIST in IP-core design," in IEEE Design & Test of Computers, vol. 18, no. 5, pp. 92-99, Sept.-Oct. 2001, doi: 10.1109/54.953276.

[2] “ADC : Offset Error and Gain Error Explained,” *www.youtube.com*. https://www.youtube.com/watch?v=G7jkCyGipc8&ab_channel=ALLABOUTELECTRONICS (accessed Dec. 31, 2023).

[3]A. J. Atchaya, D. S. Shylu Sam, A. J. Herinsha and V. T. Mahendra, "Design of High Speed Time – Interleaved SAR Analog to Digital Converter," 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), Coimbatore, India, 2022, pp. 460-463, doi: 10.1109/ICDCS54290.2022.9780703.

[4] *Allaboutcircuits.com*. [Online]. Available: https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converter-integral-nonlinearity-inl-error/. [Accessed: 17-Jan-2024].

[5] “Art of Problem Solving,” *artofproblemsolving.com*. https://artofproblemsolving.com/wiki/index.php/Modular_arithmetic/Introduction

[6] A. Timoshenko and K. Lomovskaya, "A survey on effective techniques of designing high-performance 1-GHz bandwidth ADC," 2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers, Belgrade, Serbia, 2011, pp. 1609-1611, doi: 10.1109/TELFOR.2011.6143869.

[7]B. Chen, "A scheme for wide input range precision SAR ADC," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1029-1032, doi: 10.1109/MWSCAS.2017.8053102.

[8]B. E. Jonsson, "Area efficiency of ADC architectures," 2011 20th European Conference on Circuit Theory and Design (ECCTD), Linköping, Sweden, 2011, pp. 560-563, doi: 10.1109/ECCTD.2011.6043595.

[9]B. Le, T. W. Rondeau, J. H. Reed and C. W. Bostian, "Analog-to-digital converters [A review of the past present and future]", IEEE Signal Processing Magazine, pp. 69-77, Nov. 2005.

[10] “Built-in self-test (BiST) - Semiconductor Engineering,” *Semiconductor Engineering*, Sep. 13, 2019. https://semiengineering.com/knowledge_centers/test/built-in-self-test-bist/ (accessed Jan. 16, 2024).

[11] C. Coulston, “EENG 383 - lecture notes,” *Mines.edu*. [Online]. Available: https://inside.mines.edu/~coulston/courses/EENG383/lecture/lecture19.html. [Accessed: 17-Jan-2024].

[12] C. H. Vun and A. B. Premkumar, "RNS encoding based folding ADC," 2012 IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Korea (South), 2012, pp. 814-817, doi: 10.1109/ISCAS.2012.6272165.

[13] Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, India, 2015, pp. 0512-0516, doi: 10.1109/ICCSP.2015.7322537.

[14] Conference and Exhibition 2000 (Cat. No. PR00537), Paris, France, 2000, pp. 216-220, doi: 10.1109/DATE.2000.840041.

[15]D. Bundalo, F. Softie, D. Pašalić, Z. Bundalo and M. Kostadinović, "Design of analog to digital converters for multiple-valued systems," 2015 4th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2015, pp. 228-231, doi: 10.1109/MECO.2015.7181910.

[16] “Different Analog to Digital Converters Architectures,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 4, pp. 1256–1263, Feb. 2020, doi: https://doi.org/10.35940/ijitee.d1641.029420.

[17] E. Balestrieri, P. Daponte, L. De Vito, F. Picariello, S. Rapuano and I. Tudosa, "Embedded ADC testing challenges: proposals from research," 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace), Turin, Italy, 2019, pp. 567-572, doi: 10.1109/MetroAeroSpace.2019.8869608.

[18] EDN, “BIST schemes for ADCs,” *EDN*, Aug. 29, 2014. https://www.edn.com/bist-schemes- for-adcs/#:~:text=The%20most%20simplified%20BIST%20test (accessed Dec. 31, 2023).

[19] H. -S. Lee and C. G. Sodini, "Analog-to-Digital Converters: Digitizing the Analog World," in Proceedings of the IEEE, vol. 96, no. 2, pp. 323-334, Feb. 2008, doi: 10.1109/JPROC.2007.911069.

[20] "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," in IEEE Std 1241-2010 (Revision of IEEE Std 1241-2000) , vol., no., pp.1-139, 14 Jan. 2011, doi: 10.1109/IEEESTD.2011.5692956.

[21] J. A. Wepman, "Analog-to-digital converters and their applications in radio receivers," in IEEE Communications Magazine, vol. 33, no. 5, pp. 39-45, May 1995, doi: 10.1109/35.393000.

[22] Jiun-Lang Huang, Chee-Kian Ong and Kwang-Ting Cheng, "A BIST scheme for on-chip ADC and DAC testing," Proceedings Design, Automation and Test in Europe Conference and Exhibition 2000 (Cat. No. PR00537), Paris, France, 2000, pp. 216-220, doi: 10.1109/DATE.2000.840041.

[23] J. Patel and H. Bhatt, "Performance Evaluation Of Different Types Of Analog To Digital Converter Architecture." Available: https://www.ijert.org/research/performance-evaluation-of-different-types-of-analog-to-digital-converter-architecture-IJERTV1IS10129.pdf

[24] J. Schat, "ADC test methods using an impure stimulus: A survey," 2018 IEEE 23rd European Test Symposium (ETS), Bremen, Germany, 2018, pp. 1-5, doi: 10.1109/ETS.2018.8400687.

[25] J. Sun, "The 1090MHz Riddle," *mode-s.org*. https://mode-s.org/decode/content/ads-b/8-error-control.html (accessed Dec. 31, 2023).

[26]K. G. Merkel and A. L. Wilson, "A survey of high performance analog-to-digital converters for defense space applications", *Proc. IEEE Aerospace Conf.*, vol. 5, pp. 2415-2427, Mar. 2003.

[27] K. Yoo *et al.*, "Built-in Self-Test Implementation for an Analog-to-Digital Converter," *Journal of the Korean Physical Society*, vol. 41, no. 6, pp. 963–966, 2002, Accessed: Dec. 31, 2023. [Online]. Available: http://soc.yonsei.ac.kr/Abstract/International_journal/pdf/No26_Built-in%20Self-Test%20Implementation%20for%20an%20Analog-to-Digital%20Converter.pdf

[28] M. E. R. Romero, E. M. Martins, R. R. Santos and M. E. D. Gonzalez, "Analog to digital converter for binary and multiple-valued logic," 2011 IEEE Second Latin American Symposium on Circuits and Systems (LASCAS), Bogota, Colombia, 2011, pp. 1-4, doi: 10.1109/LASCAS.2011.5750283.

[29] M. George, “A comprehensive overview of BIST in VLSI,” *ChipEdge VLSI Training Company*, Dec. 07, 2023. https://chipedge.com/a-comprehensive-overview-of-bist-in-vlsi/ (accessed Jan. 16, 2024).

[30] M. Li, S. Kesa, and C. Williams, “Modular Arithmetic | Brilliant Math & Science Wiki,” *brilliant.org*. https://brilliant.org/wiki/modular-arithmetic/

[31]R. Nitnaware, A. D. Tijare and M. M. Matey, "Design of binary architecture for Successive Approximation Analog-to-Digital Converter," 2015 International

[32] P. A. Ramamoorthy and B. Potu, "High-speed ADC using residue number system," International Conference on Acoustics, Speech, and Signal Processing,, Glasgow, UK, 1989, pp. 1063-1066 vol.2, doi: 10.1109/ICASSP.1989.266615.

[33] R. C. Hicks, "A survey of analog-to-digital converters for radar applications," 92 International Conference on Radar, Brighton, UK, 1992, pp. 534-537.

[34] R. H. Walden, "Analog-to-digital converter technology comparison," Proceedings of 1994 IEEE GaAs IC Symposium, Philadelphia, PA, USA, 1994, pp. 217-219, doi: 10.1109/GAAS.1994.636970.

[35] R. H. Walden, "Analog-to-digital converter survey and analysis," in IEEE Journal on Selected Areas in Communications, vol. 17, no. 4, pp. 539-550, April 1999, doi: 10.1109/49.761034.

[36]“Sample and Hold circuit,” *Electronics Coach*, 19-Dec-2017. [Online]. Available: https://electronicscoach.com/sample-and-hold-circuit.html. [Accessed: 17-Jan-2024].

[37] S. Arar, “Understanding ADC Integral Nonlinearity (INL) error,” *Technical Articles*, Feb. 22, 2023. https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converter-integral-nonlinearity-inl-error/

[38] S. Bashir, S. Ali, S. Ahmed and V. Kakkar, "Analog-to-digital converters: A comparative study and performance analysis," 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2016, pp. 999-1001, doi: 10.1109/CCAA.2016.7813861.

[39]S. Shrivastava, S. Malviya, and N. Gupta, “Built in self test architecture using logic module,” *International Journal of VLSI Design and Communication Systems*, vol. 8, no. 4, pp. 25–34, Aug. 2017, doi: 10.5121/vlsic.2017.8403.

[40]“Successive-approximation ADC,” *Ecstudiosystems.com*. [Online]. Available: https://ecstudiosystems.com/discover/textbooks/basic-electronics/ad-and-da-converters/successive-approximation-adc/. [Accessed: 17-Jan-2024].

[41]S. Weaver, B. Hershberg, P. Kurahashi, D. Knierim and U. -K. Moon, "Stochastic Flash Analog-to-Digital Conversion," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 11, pp. 2825-2833, Nov. 2010, doi: 10.1109/TCSI.2010.2050225.

[42]T. E. Linnenbrink et al., "ADC testing - Part 7 in a series of tutorials in instrumentation and measurements", *IEEE Instrum. and Meas. Magazine*, vol. 9, no. 2, pp. 39-49, 2006.

[43]T. Waho, "Non-binary Successive Approximation Analog-to-Digital Converters: A Survey," 2014 IEEE 44th International Symposium on Multiple-Valued Logic, Bremen, Germany, 2014, pp. 73-78, doi: 10.1109/ISMVL.2014.21.

[44] X. Tang et al., "Low-Power SAR ADC Design: Overview and Survey of State-of-the-Art Techniques," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 6, pp. 2249-2262, June 2022, doi: 10.1109/TCSI.2022.3166792.

[45] Zhu Hongyu and H. Li, "A BIST scheme to test static parameters of ADCs," 2012 IEEE Symposium on Electrical & Electronics Engineering (EEESYM), Kuala Lumpur, Malaysia, 2012, pp. 109-112, doi: 10.1109/EEESym.2012.6258600.