Tri-Level Comparators in High Speed SAR ADC

Ole-Georg Berg

Thesis submitted for the degree of
Master in Electronic, Information and Technology,
Microelectronics
60 credits

Department of Physics
Faculty of Mathematics and Natural Sciences

UNIVERSITY OF OSLO

Autumn 2022

Tri-Level Comparators in High Speed SAR ADC

Ole-Georg Berg

Tri-Level Comparators in High Speed SAR ADC

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

Contents

List of Figures

List of Tables

Preface

This master thesis is submitted as the final project in Electrical Engineering, Informatics and Technology, Master's Programme at the University of Oslo. The work described herein was conducted under the supervision of Associate Professor Kristian Gjertsen Kjelgård, Associate Professor Jørgen Andreas Michaelsen, and Professor Dag Trygve Eckhoff Wisland. This thesis is performed in collaboration with Novelda which is a semiconductor company located in Oslo, Norway.

I would like to extend our gratitude to the department of informatics at the University of Oslo for constructing an outstanding Master's Programme and recruiting exceptional lecturers that eagerly have shared their knowledge with me.

Finally, I will especially honor my family which have motivated and supported me through both the bachelor thesis and master thesis.

Oslo
September 14, 2022
Ole-Georg Berg

1 Introduction

In modern electronics systems, interaction between the analog domain and the digital domain is inevitable in order to perform signal processing in the digital domain. Signal conversion is necessary to interact between these domains as the real world is analogue. Data converters are named according to the conversion direction, accordingly, Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC).

ADCs convert an analog time domain signal with continuous amplitude to discrete time with discrete amplitude as illustrated in Figure 1. A digitalization process may be split up into two processes, respectively, time discretization as referred to sampling of the input signal and amplitude discretization referred to quantization [14]. The sampling process is to ensure the input signal remains steady throughout the conversion. It can be realized by either a track-and-hold (T/H) or a sample-and-hold (S/H). A T/H architecture continuously tracks the input signal during the "track" phase and keeps it steady during the "hold" phase. The quantization process is to find the digital output code that best represents an analog sampled value [14]. The closest quantized representation of the sampled input voltage is outputted as the convert's finite digital conversion code.

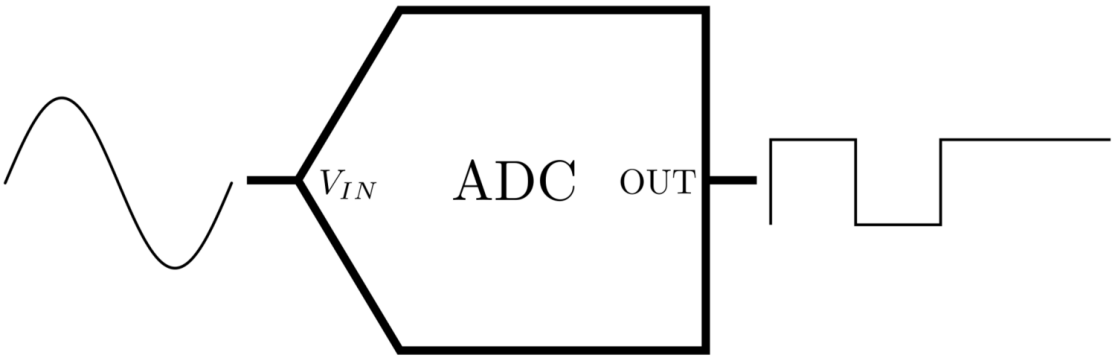


Figure 1: ADC Symbol

There is a finite number of quantization steps within the converter's full scale range and it is two to the power of N, were N is the resolution of the ADC in bits. The quantization process is actually a rounding to the closest quantized representation of the sampled input voltage, therefore it adds error to the input signal. The quantization error is commonly referred to as quantization noise. It does not appear as physical noise, but it is a direct result of the voltage error caused by rounding to the nearest quantization level. Converters are limited by the quantization noise and its restricted to $\pm$ ½ LSB, were the Least Significant Bit (LSB) is the step size between two adjacent quantization levels as depicted in Figure 2.

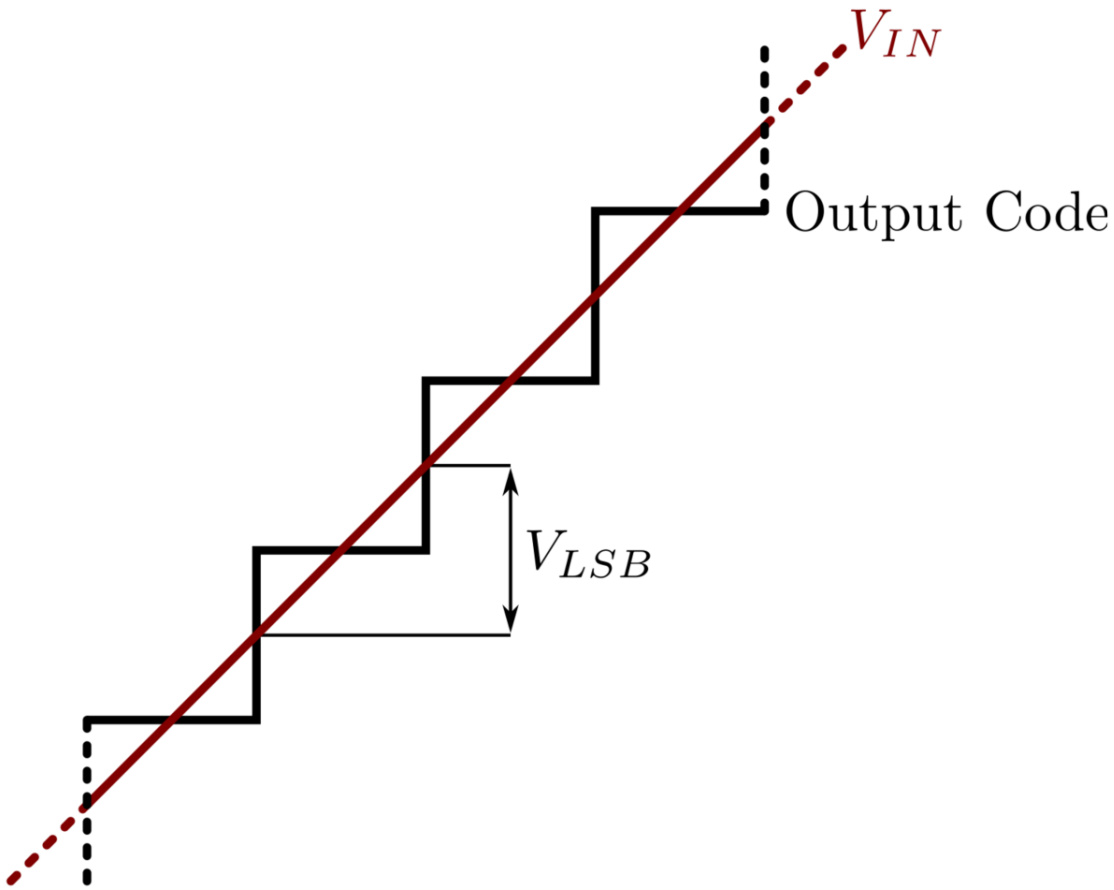


Figure 2: Quantization Step Size

1.1 ADC Architectures

Converters are categorized into two categories, respectively parallel and serial converters. A parallel converter outputs the digital code in a single step by simultaneously comparing different quantization levels. Serial converters compare discrete signal in different quantization levels in a serial fashion. The Successive Approximation Register (SAR) ADC is a serial converter. Parallel converters are intrinsically faster and serial converters are slower. Techniques to improve the converter's overall conversion speed exists and is investigated.

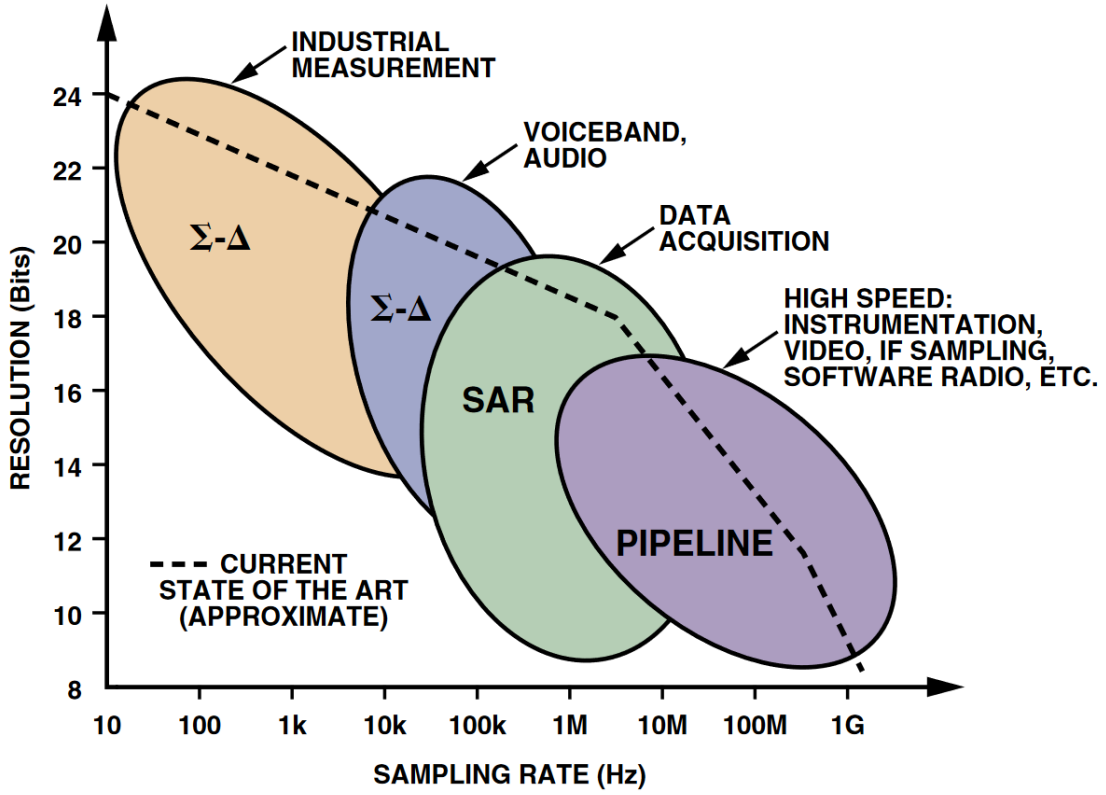


Figure 3: ADC Architecture Comparison [8]

Converters are applicable for different application were it has strengths and weaknesses. Figure 3 displays in general different ADC architectures with their applicable segments of applications, sampling rate, and the achievable number of bits resolution in the converters. The pipeline architecture is more like a method of realizing a relatively fast and power efficient converter. The SAR ADC is suitable for the purpose of this work considering its power efficiency [13]. SAR ADCs are not intrinsically fast, but their speed is improvable by running multiple ADC slices in parallel and achieving an outstanding high speed low power converter suitable in high speed converters for RF applications.

1.2 SAR ADC

The SAR ADC architecture is ubiquitous in data converters due to its power efficiency [13] and compatibility in modern CMOS processes. SAR ADC operates in a serial fashion were the unknown digital representation of the sampled input signal is gradually solved throughout the successive approximation cycles. The process of finding the closest quantization level (digital code) of the sampled input signal act as a binary search algorithm. A SAR ADC is composed by utilizing mainly three components, respectively a comparator, DAC and SAR logic. The key component is the comparator which

performs the binary search for the closest quantized representation of the sampled input voltage. The comparator has a limited amount of time making each comparison and output a well defined decision to ensure an operational binary search algorithm. The SAR cycles are complete without unacceptably large error if the comparator resolve input voltage ± ½ LSB fast enough. The comparator decision time behaves in an exponential characteristic given by its time constant, respectively (τ). A fast comparator is accomplished by designing it with a small enough time constant. To achieve a long Mean Time Between Failures (MTBF) is directly translated to design the comparator with a small enough time constant. The comparator's available decision time decreases with increasing converter speed. To accomplish the SAR cycle to complete without unacceptably large errors in the result, it is necessary to design the comparator fast enough. The converter's MTBF rate is proportional to the comparator's available decision time. As a consequence, reducing the available decision time results in a shorter MTBF, and directly limits the attainable converter's speed. This translates to designing the comparator fast enough with the emphasis on guaranteeing the SAR cycle to complete. A longer MTBF requirement translates to a limited attainable converter's speed and higher power consumption.

1.3 Task for This Work

The kind of investigated ADC in this thesis is a SAR ADC. A conventional method to accomplish the SAR cycle to complete without unacceptably large error is to design the comparator with a small enough time constant. This work grasps the issue with the comparator's inevitability to complete fast enough by introducing a timeout scheme on its decision time, and hence design comparator with a modest low time constant. The central purpose of this work is to investigate the viability of imposing a constraint on the comparator decision time and guaranteed the SAR cycle to complete. Further, gain an additional bit of resolution in the converter's output by introducing a timeout scheme on the comparator decision time corresponding to ± ¼ LSB. An unresolved decision detector is employed to detect when the comparator does not output a valid logic level within a bounded time. An unresolved comparison could potentially cause errors in the digital output code. This thesis investigates the viability to detect an unresolved decision at the successive approximation steps by introducing a time constraint on the comparator's decision time.

The authors of [6, 17] outline a method of imposing a constraint on the comparator decision time and timeout the comparator when a decision is not reached within the allocated time. It is possible to resolve an additional bit of resolution in the converter's output by setting the time constraint threshold correctly. The comparator's exponential time to voltage characteristic could cause large errors in high speed SAR ADC because of the comparator's inability to complete a decision fast enough ahead of the next comparison. The ± ¼ LSB detection region is hereafter referred to as the middle-level detec-

tion region. The strict requirement for a fast comparator to achieve long MTBF reduces by imposing a time constraint on the comparator decision scheme.

It is possible to expand the comparator determination detection regions from two-level to tri-level by imposing a time constraint on the comparator decision time. A tri-level comparator uses an unresolved decision to realize a third detection region by only one comparator. An unresolved decision is related to the comparator decision time. In fact, the comparator may be able to provide a decision, but not within the defined time constraint, denoted as an unresolved decision.

Conceptually, it is possible to expand the number of time constraint segments and achieve even more detection regions to increase the converter's resolution. This thesis consists of examining if it is practical to set a time constraint corresponding to ± ¼ LSB to detect a middle-level region, hence realizing an additional bit of resolution in the converter's output and consequently double the number of quantization steps.

The comparator decision time is highly dependent on its time constant, were the time constant is very sensitive to process, voltage, and temperature (PVT) variations. To detect a middle-level decision region it is necessary to employ trimming on the comparator's time constraint. The aim of this thesis is to explore whether trimming the timeout is feasible and quantifies the overhead required compared to designing a comparator with an inherently low time constant, focusing on high sample rate ADCs (2 GSa/s) for RF applications.

1.4 Limitations

This work takes first place at a system level and then downwards at each individual component. Circuit architectures are compared at a system level and then the most suitable architecture of each component is implemented and applied in the final design. Therefore, emphasis on their implementation issues is not encountered at a system level, but reported for the circuit which is actually implemented. However, if potential implementation issues are predictable, then it is encountered at a system level perspective.

The main emphasis of this work is to investigate the viability of applying a time constraint on the comparator decision time is suitable to reduce the metastability rate of the comparator's decision and gain an additional bit resolution. Then, if time allows, take the second step and implement a fully differential SAR ADC with the implemented unresolved detection circuit.

This thesis does not encounter a design of a calibration circuit that performs foreground calibration and background calibration of the time constraint. However, the target calibration goal is presented.

1.5 Scope

This thesis concerns theoretical analysis of high speed tri-level comparators, a prototype tape-out of a PVT trimmable high speed tri-level comparator, and a tape-out of SAR ADC design using the PVT trimmable tri-level comparator. The two aforementioned designs are located at the same chip, but isolated from each other. For the remaining part of this report, the two designs are treated in relation due to their commonalities.

1.6 Prerequisite

The tape-outed circuit was assumed to be in-house one month before the submission of this thesis. There has been a risk of not being able to manufacture as planned due to global semiconductor chip shortages. Tape-out was initially assumed to be achievable at the end of September 2021, but it was not possible before after Easter this year and my work has therefore been exposed for some months. Measurements have been impossible to perform as the chip is not yet received. Planned test procedure for the measurements are presented with intended input signals.

1.7 Thesis Outline

Chapter 2 takes a brief look at the core of a SAR ADC, including its basic functionally, principle of operation, and building blocks. Additionally, a brief view of the principle of the adopted PVT trimmable tri-level comparator. It is assumed the reader is well versed in the fundamental building blocks within the field of analog electronics and mixed-signal CMOS integrated circuit design, many potential topics have been omitted from this chapter due to preserving the central part of this thesis. A conceptual overview of the thesis aim and fundamental concepts are provided in Chapter 3, which must be apparent to fully appreciate the rest of the report. Chapter 4 describes the implementation portion of the project. It is separated into two parts: the design of the PVT trimmable tri-level comparator and the integration of the PVT trimmable tri-level comparator in a fully differential SAR ADC. Nuances and considerations related to the implementation are presented along the way. Chapter 5 presents the testing and validation of the implementation, were the core part of this project is first presented, then the second goal takes place. First, conceptualizing of simulation methods is described, followed by actual simulations, and lastly concluded with the test results. In Chapter 6, reflections on the conducted work are analyzed and reflected upon. In the end, Chapter 7 concludes the report with an outlook for future work.

2 Background

This chapter concerns the most pertinent topics related to the remaining of this report. First, the SAR ADC fundamental principle is explained, followed by a detailed view of the SAR ADC building blocks. Then a view of the comparator as a building block in a SAR ADC and its metastability behaviour. Thereafter, a schematic view of the selected clocked comparator architecture is depicted and explained in detail. Then a method of establishing a mathematical equation to describe the exponential time to voltage relationship in the comparator is founded. Followed by a view of DAC architectures and a state-of-the-art switching scheme. Then a view of the SAR logic realization. Consequently, a study of different time constraint architectures. Finally, a description of the comparator's PVT relationship and the applicability of trimming is presented.

2.1 SAR ADC

SAR ADC is quite commonly used because of its scaling friendly architecture [14] that has been implemented in a wide range of electronic components. It is referred to as a binary search algorithm that searches across multiple cycles to find a successive binary approximation of the sampled input signal. Figure 5 views the binary search for the closest quantized representation of the sampled input voltage, were the error voltage between V_{in} and V_{ref} approaches zero by adjusting V_{ref} most commonly in a binary fashion. One bit is resolved at each SAR cycle and it allows the next cycle to halve the residue search region. Therefore, the search area is narrowed in a binary fashion throughout the SAR cycle were the binary output word representation approaches the unknown sampled input signal V_{in}. Figure 4 shows a basic outline of a SAR ADC and its fundamental building blocks.

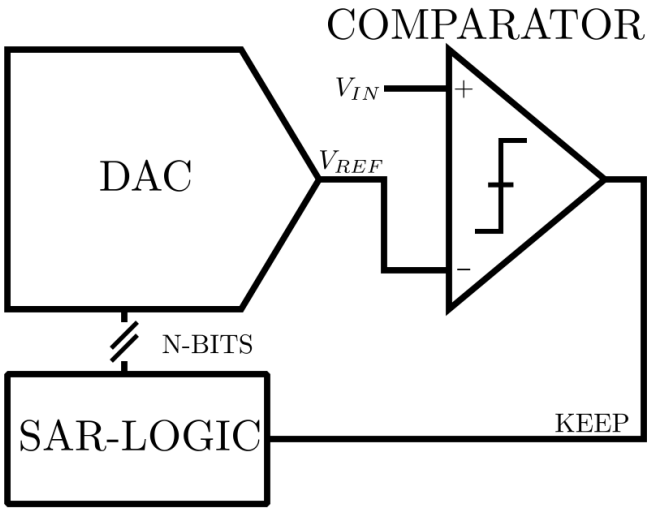


Figure 4: Conventional SAR ADC Block View

Figure 5 assume V_{in} is sampled and held constant during the whole quantization period. From a generic view; a SAR conversion starts with the SAR logic testing the MSB bit, and reevaluates the current bit determination based on the comparator's decision, and clears the bit if it was too high. Then 2^{nd} MSB is set high and tested accordingly. This process continues until all bits are tested. A digital approximation of the input signal V_{in} is quantized by ± ½ LSB at the end of the successive approximation cycle. Figure 5 shows the SAR ADC's binary search principle searching for a binary representation of an unknown value by N-step. The number of step and full scale input signal range determines how close the guessing value V_{ref} reaches V_{in} for a given LSB.

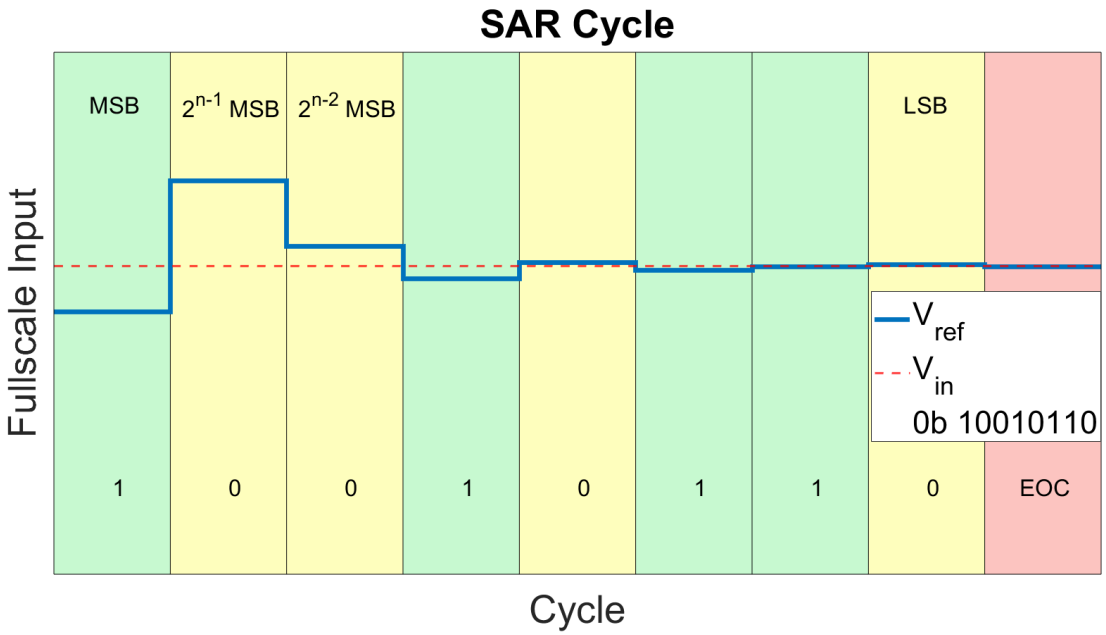


Figure 5: SAR Cycle

From a general outline, SAR ADCs does not require a special type of DAC or sample circuit, it is applicable to use any type of S/H or T/H and DAC. An energy efficient method of designing a DAC is with a charge redistribution method and it further investigated when the DAC is to be realized.

2.2 SAR ADC Overview

A fully differential configuration of a conventional SAR ADC is depicted in Figure 6. The differential input signal is sampled and stored by the DAC. The power and area needed for an S/H or T/H are to a certain extent eliminated by merging it into the differential DAC. One bit is solved at each cycle were the SAR logic controls digitally the binary weight in the differential DAC. A differential structure is preferred due to common mode rejection which cancels common-mode signals at the differential pairs.

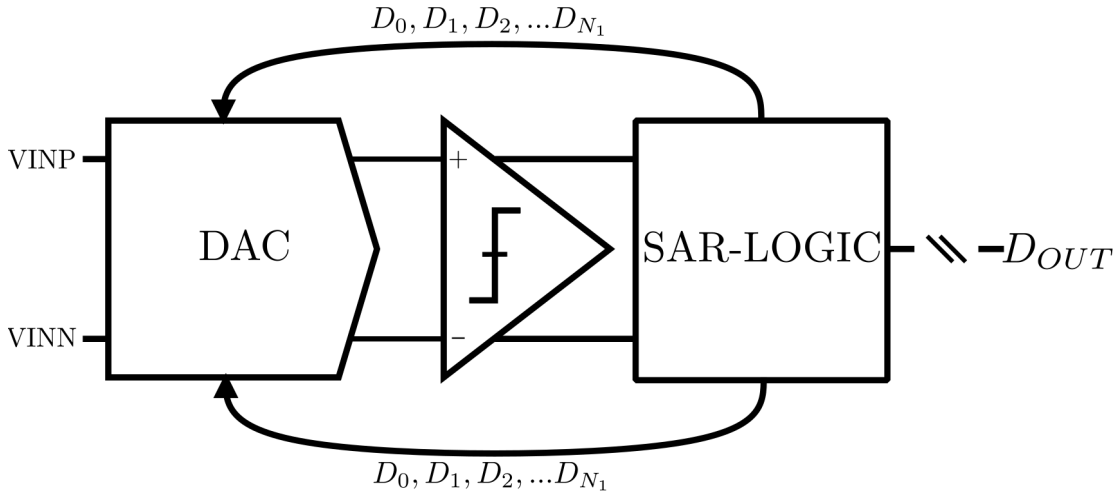


Figure 6: Block View of a Fully Differential SAR ADC

2.3 Comparator

Comparators are very important building blocks in ADCs and are considered as a one-bit quantizers. An ideal comparator outputs a logic level in the response to the polarity difference between the differential inputs. It outputs a logic high if the input difference is positive, and outputs a logic low in the presence of a negative input difference. Its usually implemented as a fully differential circuit due to the non-ideal effects as noise coupling from digital circuit and substrate noise. A single-ended comparator can ideally be considered to output a logic '1' if V_{in}>V_{ref} and logic '0' if V_{in}<V_{ref}. Actually, it turns out to be more complicated due to noise and offset voltage causing it to not be a one-to-one mapping as if the input voltage is higher then it always outputs a logic '1'.

The comparator offset voltage can be imagined as a built in voltage at one of the differential input pairs which could be negative or positive. A method to lower the effect of offset voltage is by increasing the device dimension of the input pair, but it has a cost on the circuit overall performance. However, offset voltage is usually adjustable by a trimming circuit and is thus less critical.

Hysteresis causes the comparator to more likely output the previous logic value in the next decision. It means that the comparator decision depends on the previous logic decision if there exists some hysteresis. Hysteresis causes the inputs to have to move further apart to flip the output.

There exist different types of comparator architectures which is more robust against hysteresis. Several comparator categories, clocked and continuous time types have been investigated. A clocked comparator has a reset phase ahead of each comparison period which reset the comparator and causes it to start from the same state each time. One way to minimize hysteresis is to reset the comparator ahead of decisions and therefore use clocked comparators.

Clocked comparators have a dedicated time for performing a decision (latching) and reset. For a first grasp, consider the latching and reset time is equally distributed among a clock period, were the comparator is latched when *clk* '1' and reset when *clk* '0'. The comparator starts a comparison when the clock goes high and it uses positive feedback, meaning it is a cross coupled latch that can be driven to either logic high or low quite fast. The cross coupled latch has an exponential relationship between decision time and input voltage. A small τ is highly emphasized to have a fast decision.

2.3.1 The StrongARM Latch- A Clocked Comparator

The StrongARM latch is a widely used clocked comparator. This topology is favorited because of its property to consume zero static power, generate rail-to-rail output, and the input referred offset is small [16]. The architecture utilizes four precharge switches, $S_1 - S_4$, a clocked bias device, M_7, a clocked differential pair, $M_1 - M2$, and two cross-coupled pairs, $M3 - M4$ and $M_5 - M6$. It is a clocked latch and has two clock states, respectively regeneration and reset. The nodes X, Y, P, and Q are precharged to V_{DD} during the reset phase when the clock is low. Consequently, those nodes will act as internal capacitor nodes through the high clock period. The reset process "clears" the previous decision and causes it to start from the same initial state at each comparison. The circuit offer amplification and decision when the clock goes high. Figure 7 displays a schematic view of a StrongARM latch implementation.

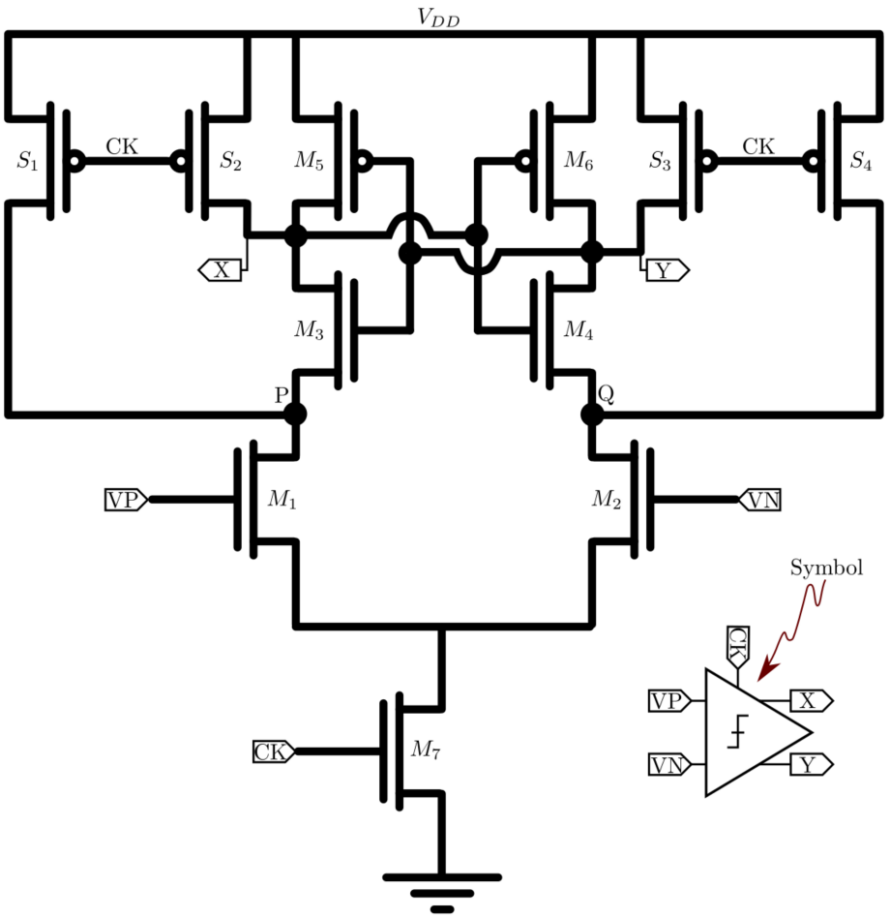


Figure 7: StrongARM Latch

The StrongARM Latch Principle of Working

Figure 8 displays the phases of a StrongARM latched comparator as it latches and resets. This illustration assumes a reset phase is asserted before the clock goes high. The first phase consists of the clock going high which turns on the tail current M_7 and switches off the precharge switches, $S_1 - S_4$. A small gate-source (V_{gs}) difference between the differential pair M_1 and M_2 causes a slight difference current flowing through the left and right branches of the differential input pairs. Consequently, nodes P and Q are discharging at unequal rates and the voltage drops at slightly different speed. The regeneration phase starts when the voltage at either node P or Q is discharged to $V_{DD} - V_{THn}$, which turns on either $M3$ or $M4$ and activates the cross-coupled pair. Any differences in the voltage at P and Q causes the back-to-back coupled inverters (cross-coupled pair) to latch the outputs to each direction (rail-to-rail) in response to the polarity of $V_P - V_N$. The speed of the latch is characterized by an exponential behavior given by comparator's time constant.

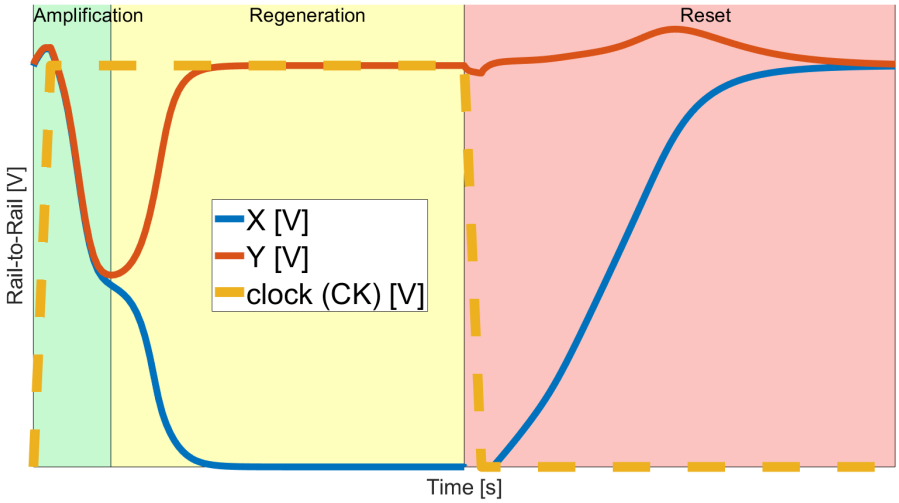
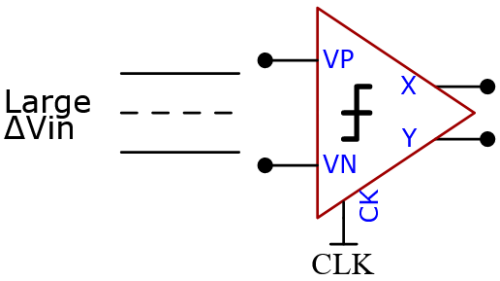


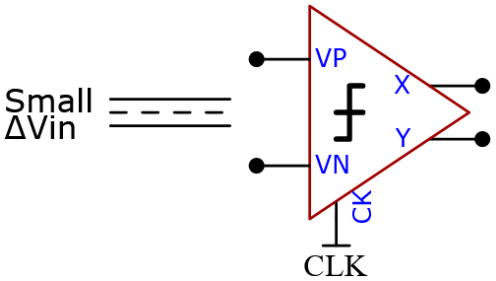
Figure 8: The StrongARM Latch Operation Phases

2.4 Comparator Metastability

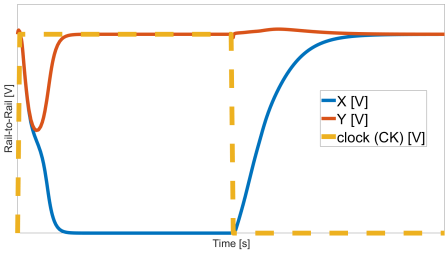
Metastability in high speed comparators is unavoidable. The prefix "meta" is a loanword from Greek that means between. A metastable state is a state existing between a defined logic level denotes as an intermediate state. The exponential time to voltage relationship in the latch causes the decision time to increase exponentially towards its maximum point until the effective differential input voltage is zero and then decays exponentially. Metastability in the latch becomes serious a problem when the effective differential inputs are close to zero. The metastability rate for the PVT trimmable tri-level comparator is reduced as the comparator decision time is constrained. The comparator's speed and input voltage relationship is illustrated in Figures 9a and 9d. In a metastable state, the comparator output could balance for a long time meaning the comparator would take too long time to actually make a decision. The comparator is stated to be in a metastable state when it is unable to make a decision during the allotted time period. There is not a clear way to eliminate metastability completely, but by making the comparator faster it is possible to minimize the probability of metastability. Metastability in comparator becomes an issue at high clock frequency due to the available comparison time persisting in an inverse proportionality to the clock frequency. Therefore, the comparator has a smaller resolve time window to actually perform a comparison by an increased clock frequency. Metastability in comparators can not be totally eliminated and it is specified by a mean time between failure (MTBF) metrics.



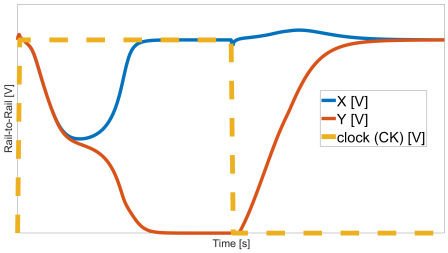
(a) Large $\Delta\ V_{in}$



(b) Small $\Delta\ V_{in}$



(c) Fast Comparator Decision



(d) More Sluggish Comparator Decision

Figure 9: Decision with Large and Small Effective Differential Input Voltages

2.4.1 Comparator Metastability Analysis

Authors of [11] presents a method of measuring the time constant (RC) in the latch. It consists of applying near zero volts across the differential inputs, latching the comparator, and measuring the time difference between v_2 and v_1 are present at the differential output. v_1 could e.g. be selected to be 1 mV and then $v_2 = e \cdot v_1$. The time constant in the latch is extracted by taking the differences between t_2-t_1.

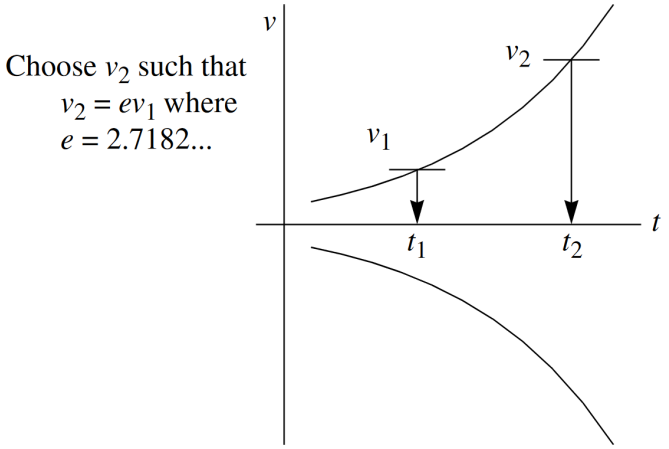


Figure 10: Exponential Relationship Between Input Voltage and Decision Time [11]

The exponential characteristics in the latch is expressed as following:

$$\begin{aligned} v_2 &= e * v_1 \\ e &= \frac{v_2}{v_1} \end{aligned} \tag{1}$$

Metastability Analysis

Comparator's metastability behavior is analyzed by employing the method presented in [11]. Figure 11 is used to analyze the comparator's latching behavior and to define a mathematical model for describing the comparison phase.

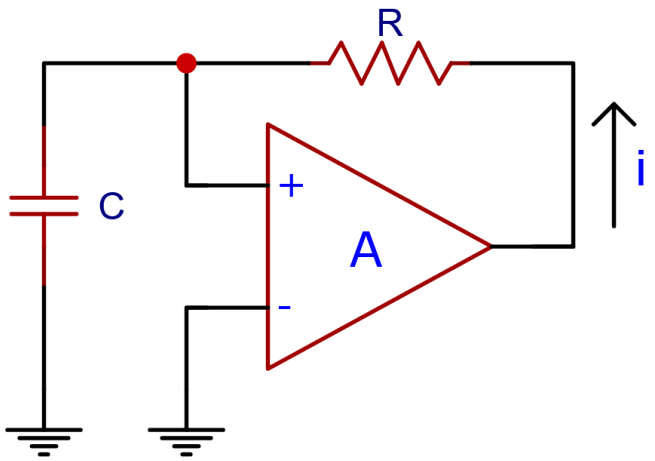


Figure 11: Circuit Model of a Latched Comparator During the Latching Process [11]

Authors of [11] formulates an equation for the voltage at the latch as in Equation 2. This equation is fundamental for further analyzing metastability in latched comparators. Equation 2 describes mathematical the behavior of latched comparators as it latches [11].

$$v = v_0 e^{\frac{(A-1)t}{RC}} \tag{2}$$

And, by inserting Equation 2 in Equation 1 are the following relationship discovered.

$$e = \frac{v_2}{v_1} = \frac{v_0 e^{\frac{(A-1)t_2}{RC}}}{v_0 e^{\frac{(A-1)t_1}{RC}}} = e^{\frac{(A-1)}{RC}(t_2-t_1)} \tag{3}$$

By solving Equation 3 concerning RC yields:

$$\begin{aligned} ln(e) &= ln(e^{\frac{(A-1)}{RC}(t_2-t_1)}) \\ RC &= (A-1)(t_2-t_1) \end{aligned} \tag{4}$$

By substitute Equation 4 into Equation 1 yields:

$$v = v_0 e^{\frac{t}{(t_2-t_1)}} \tag{5}$$

Equation 5 can be further reorganized. By assuming the comparator must switch to a logic threshold voltage V_L=1 by time T, and let ϵ be the input voltage present at the comparator's differential inputs yields:

$$\epsilon = V_L e^{\frac{-T}{(t_2 - t_1)}} \tag{6}$$

To solve Equation 6 with respect to T is natural log at both side performed.

$$\begin{aligned} ln(\epsilon) &= ln(e^{\frac{-T}{(t_2 - t_1)}}) \\ T &= -(t_2 - t_1) \cdot ln(\epsilon) = -\tau \cdot ln(\epsilon) \end{aligned} \tag{7}$$

Finally inserting the effective input voltage $\epsilon = V_{in} - V_{offset}$ and convert from natural log to logarithm is the following relationship obtained:

$$T_{DCSN}(V_{in}) = -\tau \cdot log(|V_{in} - V_{offset}|) + t_d \tag{8}$$

The factor t_d is added to the equation for modulating supplementary fixed delay. It modulates the propagation delay through the skewed output inverters at the comparator's outputs and as well the time before the comparator output flips rail-to-rail, respectively the amplification phase in Figure 8. The t_d adds a fixed delay and it does not influence the equation except adding a constant time shift and the exponential latching equation is still valid.

Equation 8 describe the comparator decision time for an arbitrary V_{in} for a given τ, V_{offset}, and t_d. The motivation of formulating an equation for the latching behavior is to have the possibility to estimate the decision time for an arbitrary input voltage V_{in}, and estimate the required time constraint to state when the input voltage lies within ± ¼ LSB. Additionally, it is employed to estimate the required time resolution for the time constraint for an introduced acceptable error contribution.

2.5 Charge Redistribution DAC

The generation of intermediately voltage levels within the converter's full scale range relies on the ratio between the binary weights in the DAC structure. The DAC's output voltage is changed by editing the capacitor ratio by connecting or disconnecting binary weighted capacitors from the array structure.

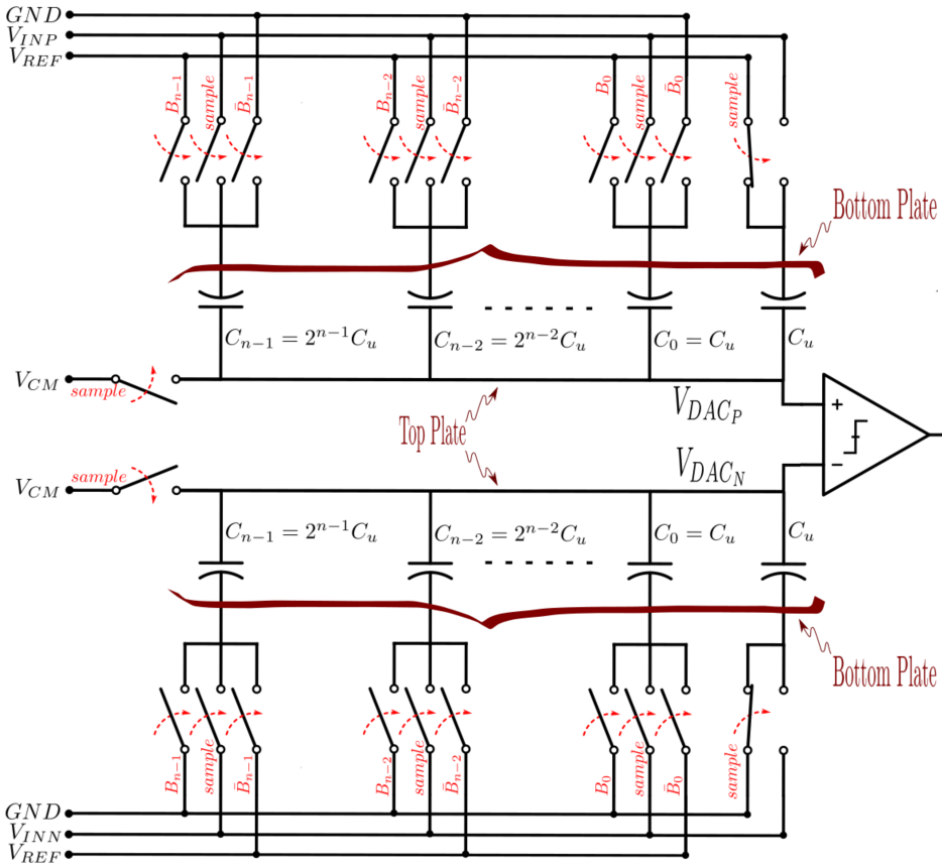


Figure 12: Conventional N-bit Differential Charge Redistribution

Figure 12 outlines a conventional realization of a fully differential charge redistribution (CR) DAC. The DAC's output voltage is adjusted by controlling the connection point of the capacitor's bottom plate by managing the state of the switches. In practical realization of CR DAC it is common to merge the sample circuit (e.g. T/H) into the capacitive array and it is achievable by employing additional sampling switches. It initially relies on sampling the differential input signal (V_{INP} and V_{INN}) at the capacitors' bottom plate and simultaneously charges the top plate to the common mode voltage V_{CM}. After the input signal is sampled, the opposite polarity of the input voltage is stored at the top plate of the captive array. The bit cycling phase takes place after the differential input signal is sampled. The convectional switching scheme relies on connecting the MSB

capacitor of the top array to V_{REF} while the other top array capacitors are connected to GND. The bottom array is configured with opposite polarity as the top array.

There have been developed many different switching schemes intended to improve the energy efficiency and reducing area consumption [14]. Capacitors are either connected or disconnected during the bit cycle to minimize the error voltage between the input signal and the DAC voltage. The switching of binary weighted capacitors results in one of two transitions, respectively "up" or "down" transition. The conventional CR DAC use charge ineffective during a down transition, it actually takes five times more energy to lower the DAC voltage during a "down" transition of the MSB bit [4, 5, 14]. This behavior occurs due to the initial charge stored at the capacitor to be lowered is discharged to ground, while the capacitor for the next bit determination is up charged from the reference power supply. In the literature, there is presented more energy efficient method of realizing a CR DAC implementation to reduce and equalize the amount of consumed energy for an "up" and "down" transition [4, 5].

2.6 Split-Capacitor Charge Redistribution DAC

As mentioned, the conventional CR DAC switching scheme is inefficient during a down transition because it lowers the C_{n-1} capacitor and simultaneously raises the C_{n-2} capacitor after the first bit determination. Inefficiently switching is related to up charging the C_{n-2} capacitor from the reference voltage and simultaneously discharging C_{n-1}. Contrary, the split CR DAC switching scheme splits the capacitive array into two subarrays were the total capacitance is unchanged. The advantage of this method is the possibility to use charge more efficiently during a down transition. Rather than upcharge another capacitor ahead of the next bit determination during a down transition, charges are reused by lowering only one of the MSB subarray capacitors. The concept of lowering capacitor rather than upcharge capacitor during a down transition yields for all successive approximation steps. Figure 13 depicts the classic realization of a single ended (SE) CR DAC and Figure 14 views the alternative split SE CR DAC implementation. A split CR DAC requires more switches, however, its advantages defend its budget compromise. Advantages as reduced settling, improved switching energy efficiency, and the number of required clock phases is unchanged, which would have been limiting high speed operation [5].

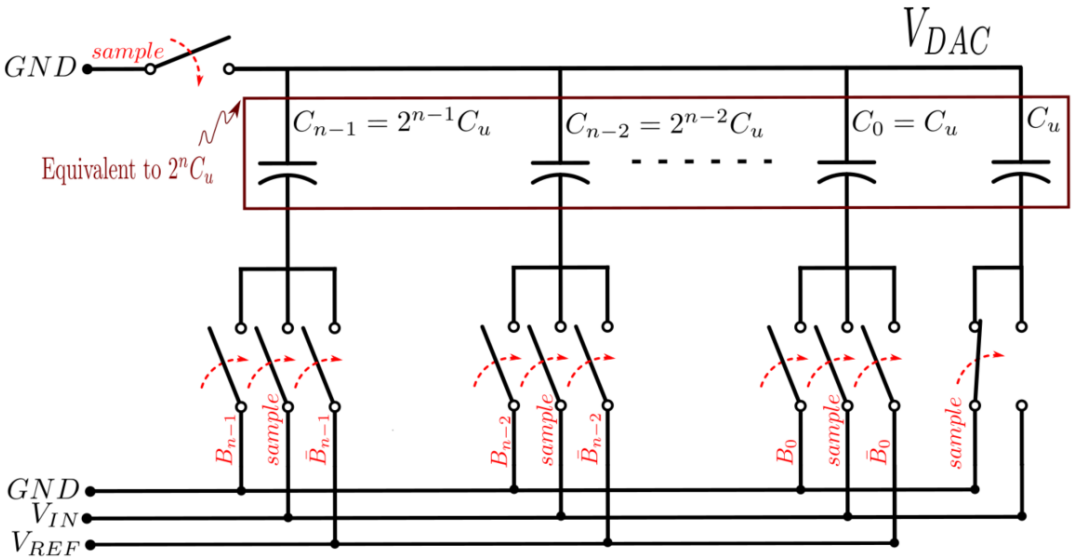


Figure 13: Conventional N-bit SE CR DAC

Consider possible transitions for a 3-bit split SE CR DAC. First, the input signal V_{IN} is sampled and stored at the capacitor array. Thereafter, the bit cycling phase takes place. Rather than connecting the C_{n-1} capacitor to the reference voltage as in a classic CR DAC, the whole MSB subarray is connected to the reference during the first bit cycle. Connecting the MSB subarray to V_{REF} causes the same amount of change at the DAC's output voltage as for the first bit determination in the classic CR DAC, respectively $V_{DAC} = -V_{IN} + \frac{1}{2}V_{REF}$. Depending on the comparator's decision, the MSB bit is set low if the decision were too high or set high if the decision were too low. By the presence of a low decision (up transition) V_{DAC} need to be increased ahead of the next bit determination. Therefore, the MSB subarray remains connected to the reference voltage, and C_{n-1} is connected additionally. Opposite, with a high decision (down transition) the MSB subarray capacitor $C_{n,n-1}$ is connected to GND and the rest of the MSB subarray remains connected. Simultaneously, the C_{n-1} capacitor in the main subarray is left unconnected from the reference voltage. This process of adding capacitance from the main subarray by the presence of up transition or disconnect capacitor from the MSB subarray during down transition applies for all bit determinations throughout the bit cycling phase.

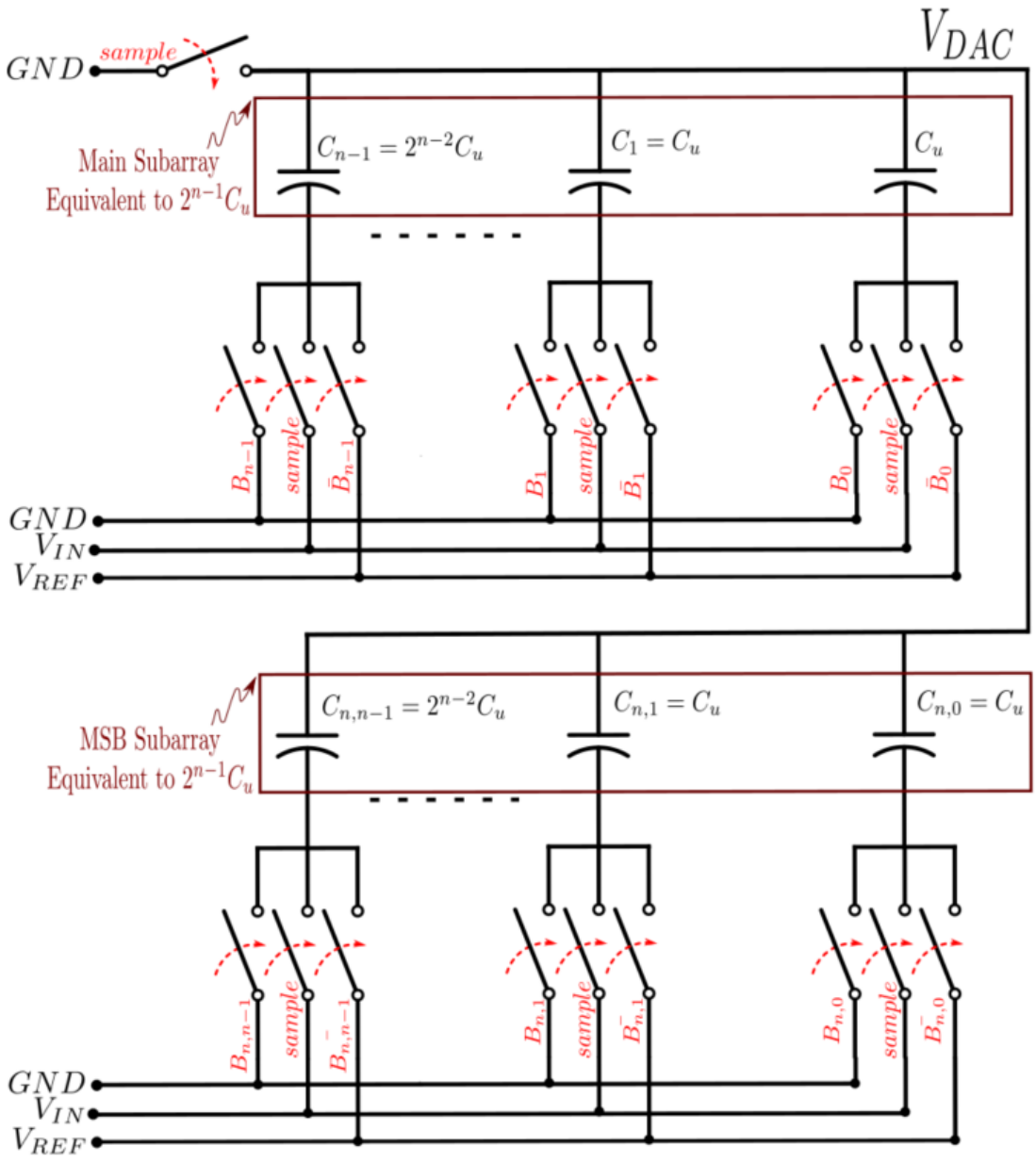


Figure 14: N-bit Split SE CR DAC

2.6.1 DAC Comparison

A conventional DAC has larger energy consumption compared to the state-of-the-art DAC as presented in [14]. It is possible to reduce the energy consumption by utilizing an alternatively switching scheme. The split-capacitor CR DAC scheme is favored from a system architecture perspective because of its energy saving compared to the convectional CR DAC. Therefore, split-capacitor CR DAC is further investigated when a suitable DAC is to be realized.

2.6.2 Capacitor Mismatch

The DAC is constructed by utilizing binary weighted capacitors to set the intermediate test voltage levels within the converter's full-scale range. The intermediate test input voltage to the comparator relies on the ratio between the capacitors. Therefore, any mismatch between the binary weighted capacitors and additionally thermal noise restricts the attainable ADC resolution. A mismatch analysis and thermal noise analysis are adopted and the dominating contribution is considered for the selected capacitor dimension.

2.6.3 Clock Bootstrapped Switch

The employed sampling switch is a clock bootstrapped switch. This architecture type resolves the issues with the switch's input-dependent on-resistance which introduces distortion [15]. The clock bootstrapped switch minimizes the switch's on-resistance within the input's full scale dynamic range [15] and it alleviates the issue with charge injection. However, it increases complexity and results in a problem of reliability. As this work is to design a prototype is not reliability problematic emphasized.

2.7 SAR Logic

The heart of the SAR ADC is the SAR logic and Figure 15 displays a conventional flip flop based implementation. It is fundamentally constructed of a chain of flip-flops were each digital bit separately is set high and reevaluated based on the comparator output level at the subsequent active clock edge, were the MSB bit is the first bit to be tested. The SAR algorithm initializes by a start of conversion (SOC) and ends by outputting an end of conversion (EOC).

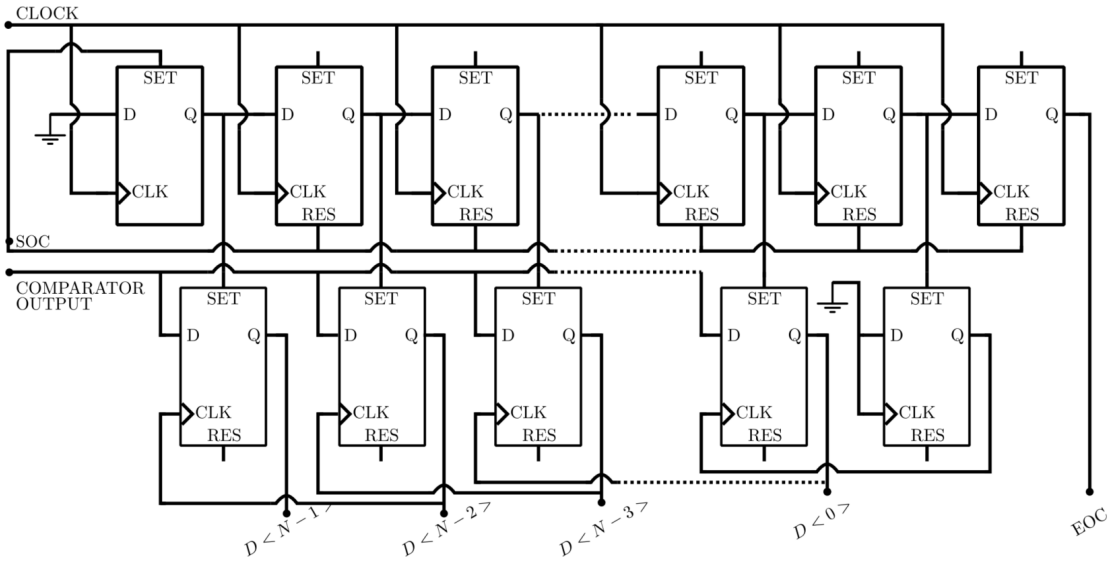


Figure 15: Conventional N-Bit Flip-Flop Based SAR Logic

2.8 Time Constraint

The purpose of this work is to investigate feasibility of constraining the comparator decision time and gain an additional bit resolution in the ADC, yields a large enough benefit to justify the complexity it incurs. This paragraph encounters the fundamentals of introducing time delay to a signal, its tunability, and potentially introduced error sources.

2.8.1 Comparator Timeout Concept

As mentioned, there is a time to voltage relationship in the comparator decision scheme. An additional bit is attained by constraining the available time for each comparison corresponding to $\pm$ ¼ LSB. Therefore, the time constraint must be trimmable across PVT variations to realize a PVT trimmable tri-level high speed comparator. The comparator comparison is time constrained by employing an unresolved decision detector at the output of the comparator. In this work, both analogue and digital tunable delay line implementations are investigated.

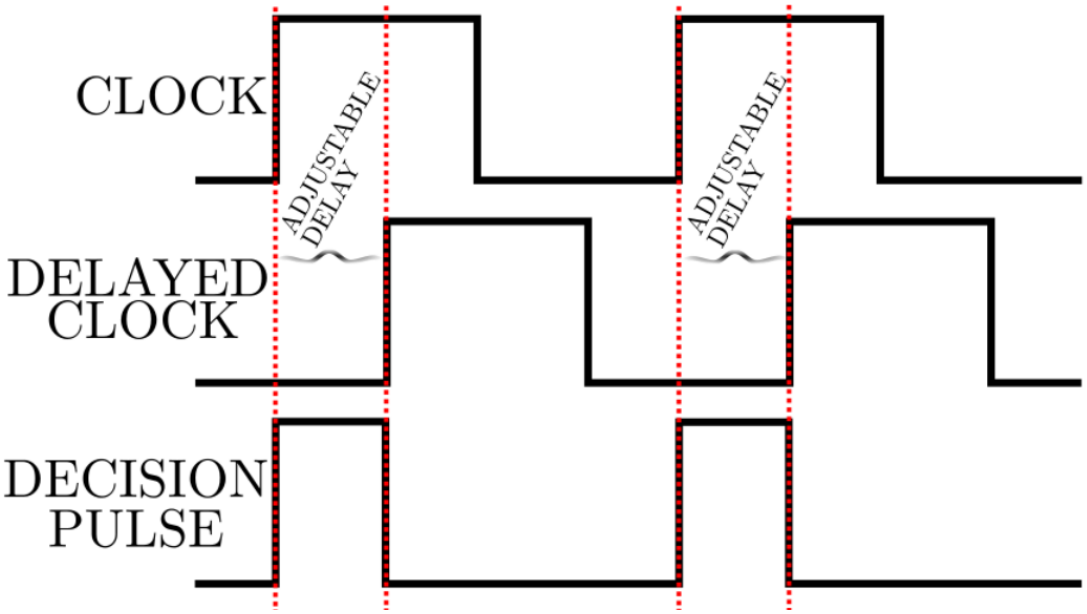
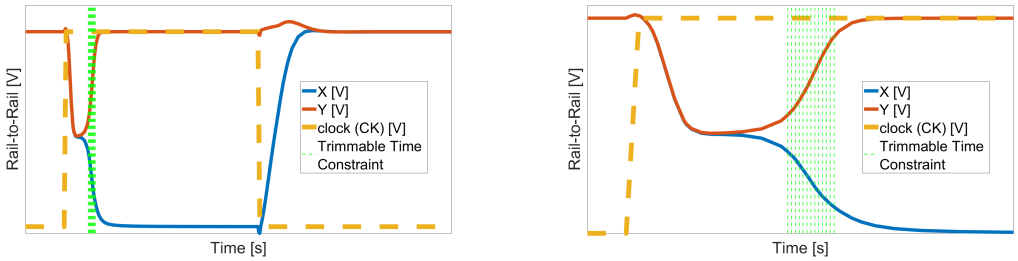


Figure 16: Illustration of a Delay Pulse

Figure 16 shows the proposed outline of a delayed pulse signal and its respective decision pulse. A delay line introduces a time delay to an input signal. The intention of constructing a decision pulse is to make an unresolved detector circuit transparent for a certain time and thereafter nontransparent. It is actually non-required to construct a decision pulse, but is more like a visual illustration to depict when the comparator outputs are captured or not. The comparator decision is only captured during a high decision pulse and consequently, the time to voltage relationship in the comparator's decision scheme is preserved. Figure 17b displays how an unresolved decision is detectable by adjusting the time constraint around the comparator's exponential decision characteristic.



(a) Large View

(b) Enlarged View

Figure 17: Trimmable Time Constraint

2.8.2 Phase Noise

Phase noise in the time constraint is directly converted to jitter in time. Furthermore, jitter is directly affecting the precision in the time to voltage mapping as viewed in Figure 18. Consequently, the adjustable delayed line needs to be robust enough against phase noise to not degrade the accuracy of the LSB bit determination.

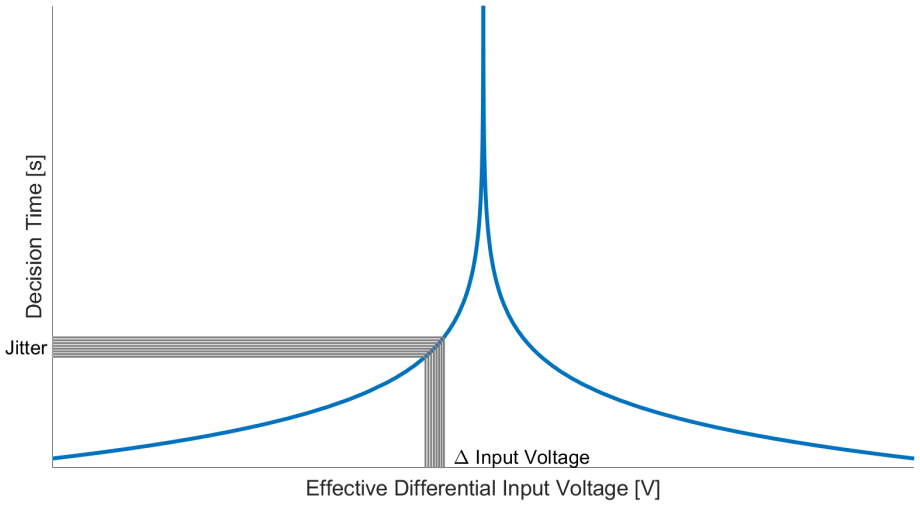


Figure 18: Time Constraint Jitter Maps Directly to Voltage Variation in the Comparator's Exponentially Decision Time Characteristic

2.8.3 Evaluating of the Impact of Phase Noise

The delay line introduces potential error sources by adding time uncertainty. It is necessary to have a variable time constraint that is robust against jitter. It is important to have low enough jitter at the delayed clock to not get unacceptable performance degradation in the LSB bit determination. Consider an ideally free running sinusoidal signal in the frequency domain, it is one tone at carrier frequency f_c. Phase noise means there are added additional frequency components to the carrier frequency f_c. The additional frequency components add time uncertainty and cause the sinusoidal period time to vary over time. Phase noise in the frequency domain corresponds to jitter in the time domain.

To evaluate the impact of phase noise in the exponential decision time behavior it is necessary to convert phase noise to jitter. Author of [7] describe a method of converting phase noise to jitter. The integrated phase noise power (A) is calculated by integrating the area underneath the phase noise power curve from the f_1 up to half of the carrier frequency f_2 as in Equation 9. The employed phase noise method samples phase noise

and therefore its is only integrated up to half of the carrier frequency. Further, by using the formula 10 presented in [7] is the RMS jitter calculated.

$$A = A_1 + A_2 + A_3 = \int_{f_1}^{f2} Lp\,(f)\,df \tag{9}$$

$$RMS_{jitter}(second) \approx \frac{\sqrt{2 \cdot 10^{A/10}}}{2\pi \cdot f_c} \tag{10}$$

Apart from the method of relating phase noise to jitter, what is acceptable jitter uncertainty? If time uncertainty caused by the phase noise of the time constraint is too large, then it degrades the accuracy of the middle-level decision detection. To estimate the effect of potential jitter in the time constraint it is necessary to employ Equation 8. As discussed, time jitter causes variation in the input voltage detection as viewed in Figure 18. Consequently, it is possible to analyze the effect of jitter in the time to voltage mapping by first estimating the potentially RMS jitter in the time constraint circuit and then relating it to the input voltage variation at a decision time around the middle-level boundary.

By first convert from logarithm to natural log and solving decision time Equation 8 regard to $|V_{in} - V_{offset}|$ yields Equation 11. Equation 12 describe the effect of input voltage variation suffered by jitter in the time constraint evolved by phase noise.

$$|V_{in} - V_{offset}| = e^{-(T_{DCSN}(in) - t_d)/\tau} \tag{11}$$

Phase noise related to the time constraint causes error around the boundary between the middle-level, and the high-level and low-level, and it is necessary to analyze its degree of influence of nonlinearity related to the LSB bit determination. Equation 12 outlines the absolute error voltage around the high-level and low-level detection caused by jitter in the time constraint. This relationship extracts the difference between the outermost edges of input voltage variation detection caused by jitter around the middle-level, and the high-level and low-level thresholds.

$$\begin{aligned} |\Delta\text{Input Voltage Error}| &= |V_{in}@(\pm V_{LSB}/4 + V_{RMS_{Jitter}}) - V_{in}@(\pm V_{LSB}/4 - V_{RMS_{Jitter}})| \\ &= |e^{-(T_{DCSN}(\pm V_{LSB}/4) + RMS_{jitter} - t_d)/\tau} - e^{-(T_{DCSN}(\pm V_{LSB}/4) - RMS_{jitter} - t_d)/\tau}| \end{aligned} \tag{12}$$

The voltage variation is further be expressed in terms of introduced error with respect to LSB as in Equation 13.

$$\text{Error Contribution in Terms of LSB} = \pm\frac{|\Delta\text{Input Voltage Error}|}{V_{LSB}} \tag{13}$$

2.9 Error Budget

To formulate an acceptable introduced error source it is essential to encounter at the ADC system as its whole. The sum of all error contributions should be less than ½ LSB as it is the final accuracy of a SAR ADC. Allowing each sub contribution to supply an error source of LSB/8 and simultaneously ensuring a total contribution is less than ½ LSB formulates the acceptable contribution of nonlinearity in an ADC.

2.10 Time Constraint Circuit Architectures

Delay lines architectures are taxonomy into tapped delay line and single output delay line [1]. Single output delay line has a single output were the tapped delay line have a single input and offer multiple outputs.

The designed delay lines must preserve a certain range and resolution, across PVT variations and noise impacts. Among those important aspects are area and power consumption important consideration which is essential to encounter when a suitable delay line shall be selected for this application. The viability of introduce a timeout scheme on the comparator decision time lapses if the delay line area must be enormous and it is important to keep the area consumption as small as possible.

2.10.1 Tapped Delay Line

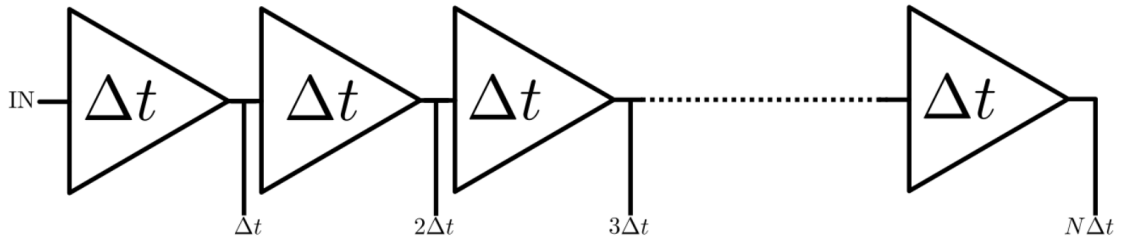


Figure 19: Tapped Delay Line

Tapped Delay Line (TDL) in its nature has a single input and multiple delayed outputs. This architecture has a fixed delay step Δt and the finest delay step is limited by the propagation delay of a single delay element (Δt). It utilizes N equally distributed delay elements were the wanted output delay is tapped out by using e.g. a multiplexer (MUX). A Δt delay element could be realized by connecting two adjacent inverters in cascade. The required time step (Δt) is accomplished by adjusting the drive strength of the inverters. It is also possible to add load between the series-coupled inverters to increase the delay even more. The delay resolution of series-coupled inverter-based TDL is fundamentally limited by the gate propagation delay through two series-coupled inverters, and it is therefore not suitable for short range delay standalone. Besides that,

the tapped method is very relevant for realizing long range delays.

It is important to be aware of the device parameters affecting the time delay (t_D) though e.g. an inverter buffer. From an equation presented in [1] $t_D \propto \frac{L}{W}$. A small Δt is realized by employ large device width (W). Opposite, the gate capacitance is $\propto$ W·L and it limits the achievable short delay step. Because the gate capacitance increases by utilizing wider transistor widths (W) it is not useful for enhance short range delay. However, this method is comparatively relevant for realizing long range delay.

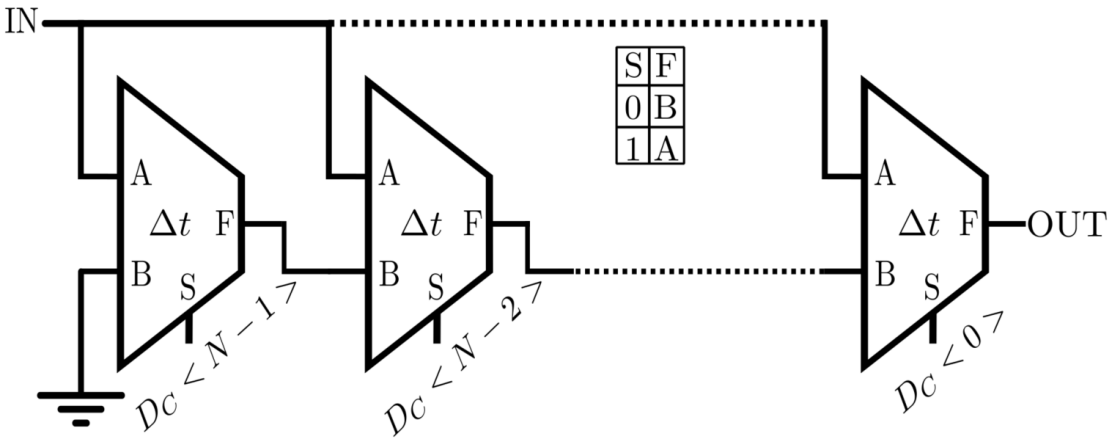


Figure 20: Tapped Delay Line Realization

Figure 20 displays a practically realization of a tapped delay line were the coarse delay is tapped out as in Table 1. It utilizes the principle of cascaded inverter buffer stages in a combination of a MUX. The MUX is either passing the previous stage or the input *IN* which is to be delayed. Only one stage is passing the input (IN) simultaneously.

D_C	Delay
0b0001	Δt
0b0010	$2\Delta t$
0b0100	$3\Delta t$
0b1000	$4\Delta t$

Table 1: Example of Digital Coded Tapped Delay Line Pattern

2.10.2 Inverters-Based Delay Line

Figure 21 displays an example of a single-output delay using the tapped delay principle. The tapped outputs are fed to a single output by using tri-state buffers. The

reader should be aware of the limitations and the possibilities of this way to implement single-output delay circuits. For a first grasp, consider the tri-state buffers introduce the same amount of delay as the inverters (Δt). Then the output is delayed by an increment of two in the following pattern, $2{\cdot}\Delta t$, $4{\cdot}\Delta t$, $6{\cdot}\Delta t$, and so on. Consequently, it is suitable for a long-range delay with a relatively large time steps. The inverted-based delay line in Figure 21 is controlled by complementary signals S0, S1, and S2 were the different delay paths are sketched by green, blue, and pink colors.

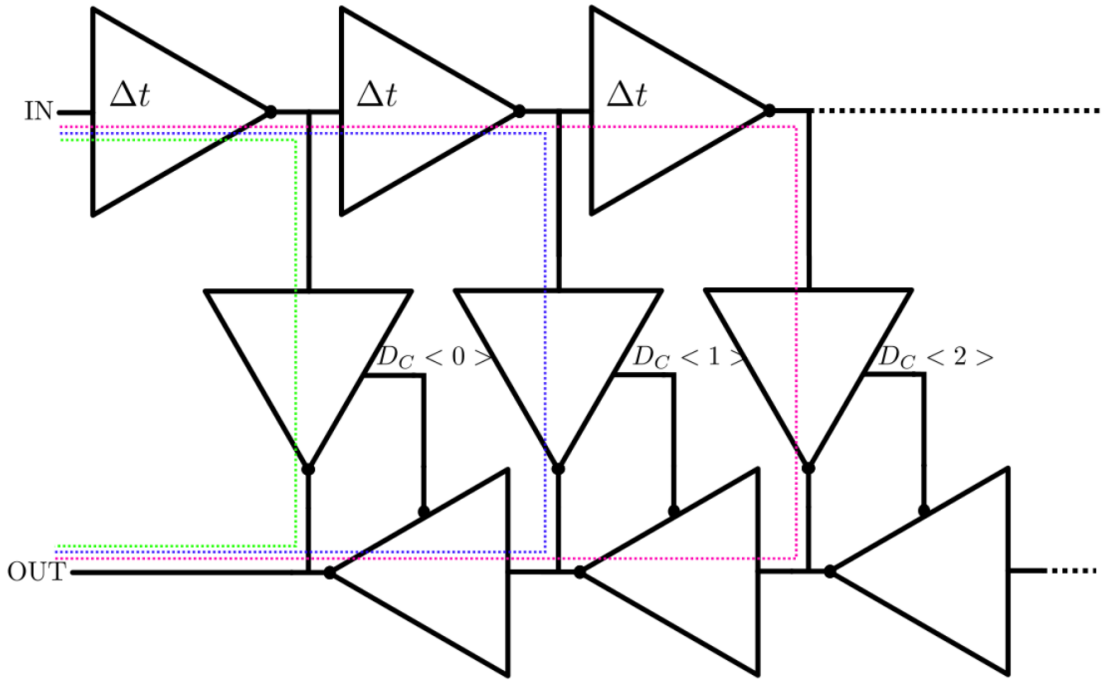


Figure 21: Inverters-Based Delay Line

2.10.3 Shunt-Capacitor Inverter

Figure 22 shows an Shunt-Capacitor Inverter (SCI) implementation. It is constructed using two inverters were the rise times and fall times of the input inverter are varied by adjusting the load with the control signals. The output inverter is added to resemble the input signal, but with an added delay. If a super-fine resolution is required then an analog tunable load element can be added in addition to the digital controlled capacitor bank. The delay step resolution for a capacitor bank standalone are limited by the transistor aspect ratio and the unit capacitor size. It is important to be aware of the practical limitations of this method of implementing a variable delay line. CMOS transistor as a switch does not to turn completely off. Therefore, there will always be present off parasitics capacitance and resistance. Consequently, it is impractical to scale the number of control bits incredibly large duo to off parasitics and resistance in the realization of small delay. This architecture is feasible to employ to achieve a relatively

fine delay line by using a reasonable number of control bits.

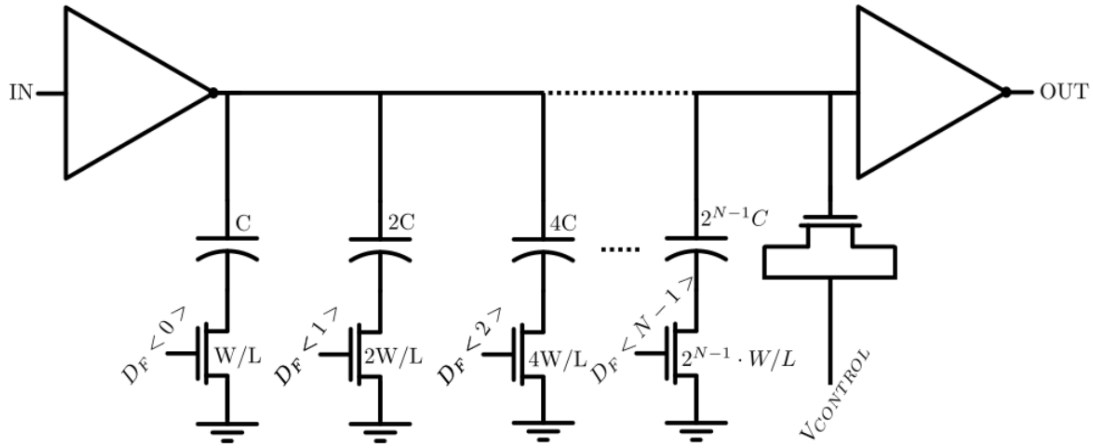


Figure 22: SCI Realization

2.10.4 Current-Starved Inverter

The current-starved inverter (CSI) implementation is a different way of implementing an adjustable delay line and Figure 23 views a general outline of its implementation. The output is delayed by modifying the current flowing through M_{10} and M_{11} resulting in a change in the speed and response of the input inverter [1]. The output inverter resembles the output with the input, but with an added delay. The CSI architecture are comprised of two inverters, two current mirrors $M_7 - M_{11}$, and a set for current sources $M - M_5$. The current source M_6 ensures a continuous bias current is flowing through the current mirrors. By adjusting the control signals it is possible to starve down the current delivered to the current source and make the first inverter stage more slower.

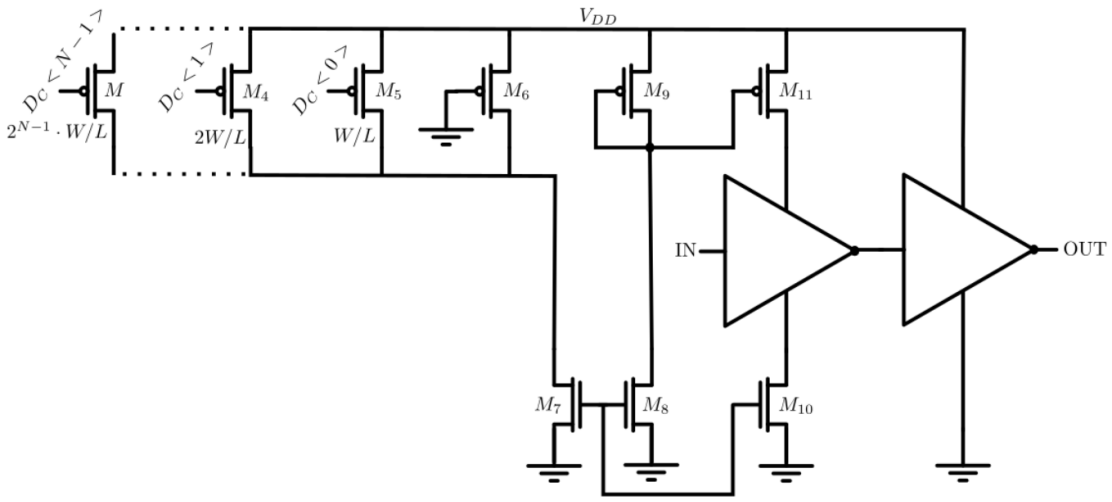
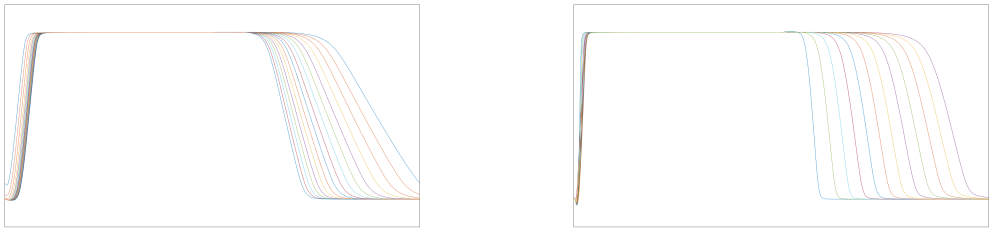


Figure 23: CSI Realization

2.10.5 Circuit Architectures Comparison

It is essential to expose the potential drawbacks of any investigated architectures. A CSI type consumes a continuously current even if it does not inverting. Apart from its current draw, it actually consuming less current when it is starved down. Inverter stage delays the input mostly when the current through M_{10} and M_{11} are most limited. Compared to the SCI type, an SCI delay element stays active when it is inverting and else inactive. Although an SCI implementation seems better considering power consumption it is not the whole picture.

As stated earlier on, the CSI architecture constraint the current delivered to the first inverter stage and resulting in time delay. Figure 24a view a schematic simulation of the first inverter and resemble inverter stage with a square wave input and its delayed outputs were the current is limited by a current mirror for the first inverter stage. The time delay decrease by burning more current, and opposite, the time delay increases by restricting the current delivered to the first inverter stage. The slope transition steepness is an important consideration when characterizing delay line's noise impact. A more gentle slope steepness causes more phase noise which is unwanted. By comparing the plots in Figures 24a and 24b it is observed that the SCI architecture seems better considering slope steepness in the delay transition. Figures 24a and 24b are simulated at a schematic level under more or less the same circumstances, which gives a quite valid compassion assessment basis. Therefore, the SCI architecture is favored above the CSI considering its noise impacts as phase noise causing error in the decision level for the LSB bit determination.



(a) CSI (b) SCI

Figure 24: Characteristics of the CSI and SCI Under Similar Circumstances

2.10.6 Proposed Time Constraint

Figure 25 displays a block overview of the purposed delay line. It is constructed using a coarse delay and fine delay. The coarse delay partitioning delays in coarse segments and the fine delay quantify fine within each coarse code. The coarse delay utilizes the tapped delay principle were each delay block has a select functionality and two series-coupled inverters. The fine delay uses the basis from the SCI type delay element. Adjusting the output load of the first inverter causes different rise times and fall times and the input is delayed by a different amount. The load between the two inverters in an SCI architecture is constructed by utilizing a binary weighted capacitor bank. The coarse segmentation is designed with nine select bits and the fine delay is constructed by utilizing four control bits.

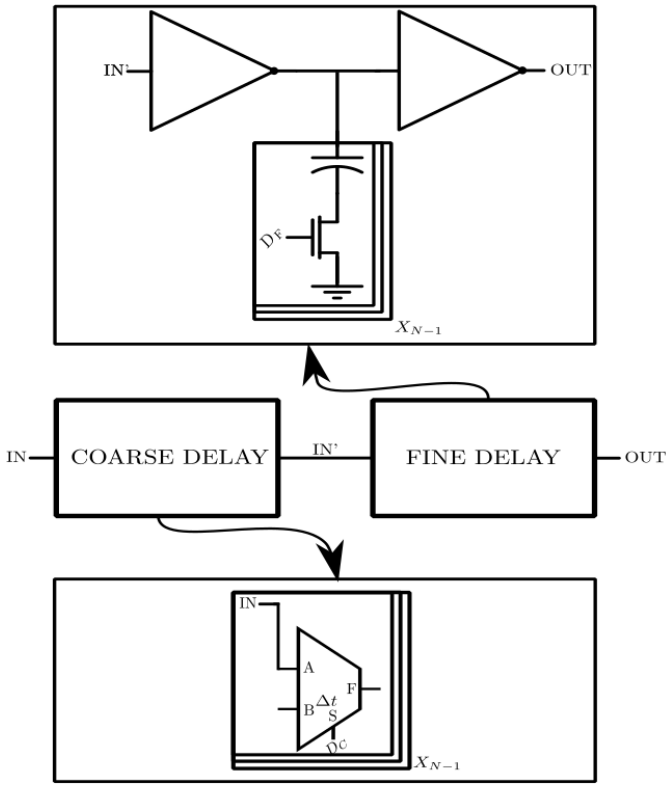


Figure 25: The Purposed Coarse & Fine Delay Line

2.11 Switched Capacitor Bank

In practical switched capacitor circuits, nonidealities such as parasitics are inevitable and set a fundamental limitation. MOSFET transistor as switch exhibit a finite non-zero resistance, R_{on}, and a small enough value is preferred for the capacitor to settle fast enough. Additionally, due to imperfection in the transistor, the switch does not actually turn completely off causing the switched capacitor to always exhibited some amount of parasitic capacitance. A tunable capacitor circuit bank is realized by employing an array of capacitors and switches in a certain pattern. The switches are programmable which controls whether the capacitor is connected or left floating and thus manages the capacitance exhibited by the capacitive array.

The switched capacitor bank is a method for dealing with the tunability of capacitance. Off parasitics limit the achievable off capacitance exhibited by the array. It is more an implementation related circumstance, but it is important to bear in mind at a system level as well.

2.12 Time-Interleaved

To improve an ADC intrinsic conversion speed it is possible to run multiple ADCs blocks in parallel, respectively denoted as time-interleaved architecture. This architecture is more a system that is applicable for ADCs. By running M-ADCs slices in parallel it is possible to increase the overall sampling frequency f_s by a factor M. In other words, each slice runs at f_s and the converters overall conversion speed is improved to $f_s \cdot M$. It is possible to realize high speed and parallel processing by using the time-interleaved principle. It is more costly in terms of power consumption and area, it is respectively inherently more complex [2]. An implementation issue is the routing because propagation delay may introduce time uncertainty considering propagation delay to each ADC slice. Differences in time translate to variations in voltage.

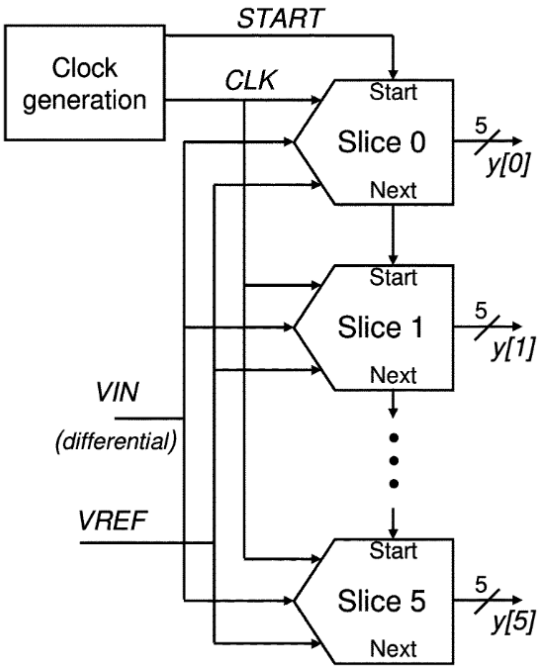


Figure 26: Block view of the 6-way time-interleaved ADC [5]

Figure 26 depicts an conceptual realization of the time-interleaved architecture. All ADC slices share a single clock which simplifies both clock generation and distribution [5]. The slices are synchronous to the global clock were each slice is activated by assigning a token to it. From a conceptual view, by the presence of *START*, $slice_0$ gets its token and it samples the input signal V_{IN} at time t_0. At time T_s after *START*, $slice_1$ get its token and samples V_{IN}. Then, at time $2 \cdot T_s$ $slice_2$ gets its token accordingly and samples V_{IN}. The token distribution causes the ADC slices to operate in a parallel fashion and the converter's overall speed is increased by a factor M. This architecture type is intended in order to satisfy the specified conversion rate with a SAR ADC. This

work does not encounter the time-interleaved principle, but it is denoted as a possible way to improve the intrinsic SAR ADC conversion speed.

2.13 Process and Environmental Variations

Due to imperfection related to manufacturing CMOS circuits, equally drawn CMOS circuits appear differently in silicon. To make the circuit operate as intended in all possible circumstances it is necessary to simulate the circuit under the different possible variations suffered from production, respectively denoted as corners. Components variation due to parameter deviation during fabrication is related to process variation. Process variations are statistically random variations that occur in production across wafers denoted as global variations and across dies on a wafer denoted as local variations. Voltage variation is unavoidable and caused by voltage (IR) drop over the power supply network. Lastly, ambient temperature variation across the chip is modulated by the temperature variation. These variations together are modulated in the abbreviation Process, Voltage, and Temperature (PVT) variations. Simulation of PVT variations across corners is employed to quantify its impact on the implemented circuits. It is necessary to employ trimming of the timeout to realize a PVT trimmable tri-level comparator as the comparator's time constant is very sensitive to PVT variations. A commercial range of PVT variations is employed for this work. Table 7 lists the circumstances for the corners to be simulated.

3 Theory

The purpose of this thesis is to investigate whether applying a time constraint on the comparator decision time and realizing an additional bit in the ADC, yields large enough advantages compared to actually having an additional bit in the ADC and designing the comparator with sufficient long MTBF. This chapter begins by scooping the concept of how an extra bit in the overall ADC resolution is realized. Then, a strategy of realizing a third level detection region in the comparator characteristics decision time behaviour is encountered.

3.1 Tri-level Comparator

The motivation of having a tri-level detection is to determine when the input signals are too small and timeout the comparator, and minimizing the probability of a situation were the comparator becomes metastable. With the ability to estimate the comparator's decision time for an arbitrary input voltage, it is then possible to predict the required timeout threshold to detect when the input signals lie within the middle-level decision region across PVT variation.

SAR ADCs have a quantization accuracy of $\pm$ ½ LSB. A conventional method to add an additional bit of resolution in the converter's output is by increasing the DAC resolution by one bit and adding an additional successive approximation cycle. Instead, to increase a SAR ADC resolution without increasing the internal DAC resolution, the comparator must be capable to decide if the input signal lies within the middle-level boundary. A timeout chosen to correspond to $\pm$ ¼ LSB effectively adds an additional bit of resolution to the converter's output. In other words, it is possible to decrease the DAC's internal resolution by one bit by having a third detection region at $\pm$ ¼ LSB in the comparator and still obtain the same number of bits. By decreasing the DAC resolution by one bit, the number of DAC's units capacitors is diminished by a factor of two, and then settling time and power consumption can be reduced. Additional, the number of comparison steps can be reduced to N-1 [17], were N is the total number of ADC bits.

There are different ways of dealing with a metastability event. Authors in [17] propose to skip the rest of a SAR cycle when the input level lies within the middle-level. However, it depends on the implementation of the detector. The unresolved decision detector becomes also metastable and it is necessary to construct the detector such that its own MTBF is considerably long enough. This is the essence of this thesis, it is constructively possible to design the comparator with a shorter MTBF rate, and utilize a detector to timeout when the comparator is too slow and increase the overall MTBF of the decision entering subsequent circuits.

Techniques presented in the literature is employed to face the issues were the detector itself becoming metastable. This phenomenon is covered later on in paragraph 3.3. Resampling a potentially indeterminate voltage level is a technique to achieve a longer

MTBF rate in the detector. Resampling increases the resolve time for each resample circuit and causes the failure rate to decay exponentially. However, an increase in resolve time involves a delay before the output value is available. Therefore, the detector operates in retrospect of when the comparator actually did a decision. The benefits of skip the rest of the SAR cycle when a middle-level detection is asserted may lapse.

3.2 Time Constraint Compatibility

The purpose of constraining the comparator decision time is to timeout when the comparator has not reach a decision fast enough. There are principally two fields were the timeout rate is adjustable. By analyzing Equation 8, the comparator decision time is adjustable by editing its associated time constant. The time constant is trimmable by employing a capacitor bank at both the differential outputs with individual control signals. The benefit of adding tuneability at the comparator outputs is to reduce the strict requirement on the time constraint circuit. On the other side, parasitics capacitance reduces the comparator intrinsic time constant and it restricts the viability of employing capacitive trimming at the comparator outputs. The second part to adjust the timeout is by constraining the comparator's available decision time by apply a trimmable time constraint. A combination of controlling the time constant and the available decision time are excepted to be most applicable in the realization of a PVT adjustable tri-level comparator.

3.3 Unresolved Decision Detector

An unresolved decision detector is intended to timeout the comparator decision when a decision is not reached within a bounded time and realize a middle-level detection. By employing a time constraint on the comparator decision time it is constructively possible to detect when the comparator input level sits within a certain voltage level, as viewed in Figure 27. In fact, even when the comparator has resolved, unresolved decision is often detected. As it is the time constraint which sets the detecting zone for the third detection region. Authors in [6, 17] present two different implementations of unresolved decision detectors. The concept of the time to voltage relationships in the comparator decision time is common for them both. An unresolved comparison from the comparator is denoted as a metastable output if the time constraint timeouts the comparator decision.

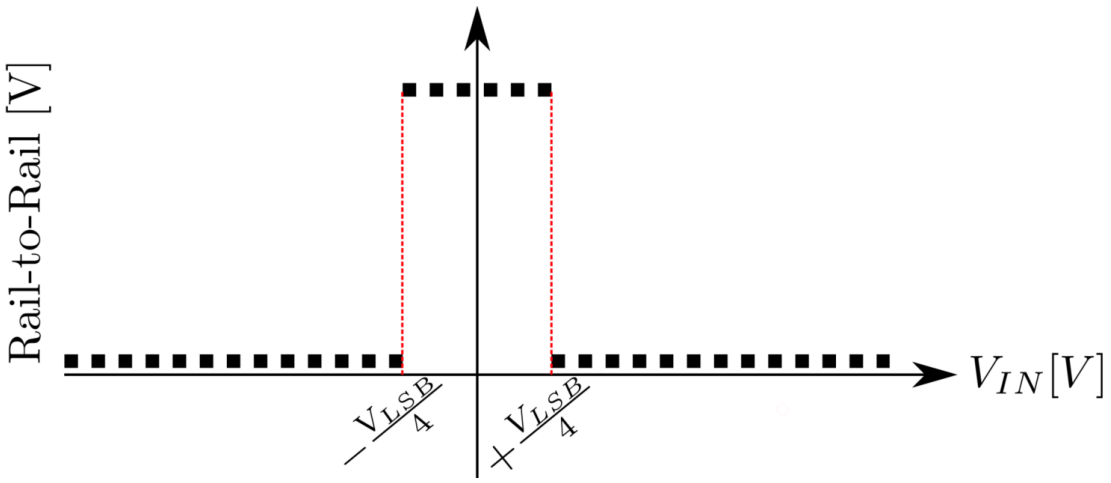


Figure 27: Time Constraint Corresponding to a Certain Input Voltage

The comparator output is asynchronous to the reference clock. The investigated methods of realizing an unresolved decision detection (URD) rely on sampling the comparator buffered output with Flip Flop (DFF) either before comparing it with an XNOR gate or after. Independent of the different implementation methods, all face a potential issue were the detector itself becomes in an unknown or unpredictable state, respectively in a metastable state. Changes in the input signal and the active clock edge appear too close in time causes setup and hold timing violations, resulting in flip flops may become metastable. The DFF input signal must satisfy a certain setup time and hold time to avoid the detector itself becoming metastable. Setup time is related to the data should be stable for a minimum time before an active edge of the clock [18]. Consequently, the data should also be stable for a minimum time after the active edge of the clock, denotes as hold time. Figure 28 depicts were input is legally permitted to change its state with respect to the active clock edge. A violated setup

time or hold time could potentially lead to erroneous data entering subsequent logic [18]. The input signal is not legally permitted to change state ahead and after a certain time of a clocking event to satisfy the setup and hold time. Fundamentally, it is not possible to control when asynchronous signals are legally permitted to change their state, but constructing the interface interacting with asynchronous inputs more robust against unpredictable output states is possible. Figure 28 depicts the time window wherein critical timing combination triggers a metastable event.

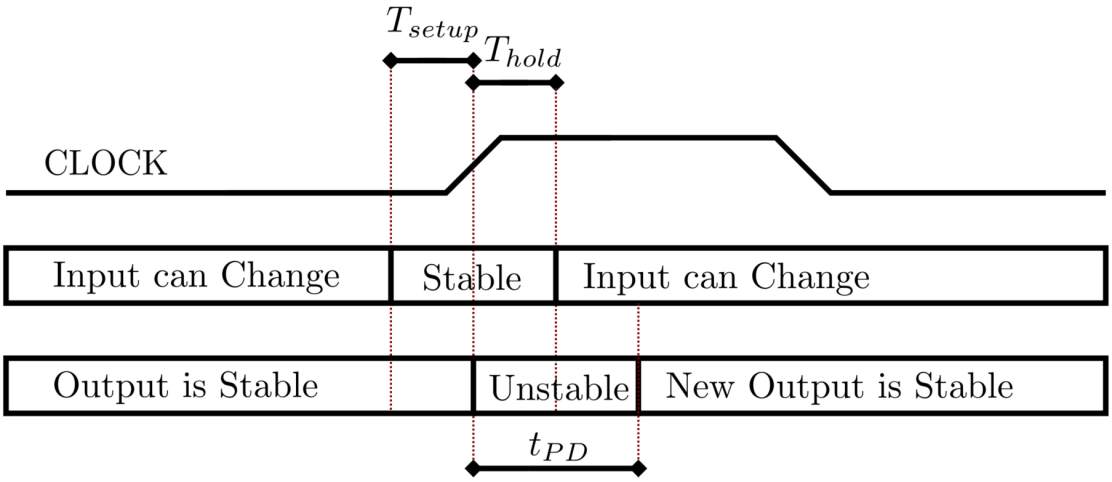


Figure 28: Illustrate the Time Window wherein Critical Timing Combination Triggers a Metastable Event

3.3.1 Synchronizer Circuit

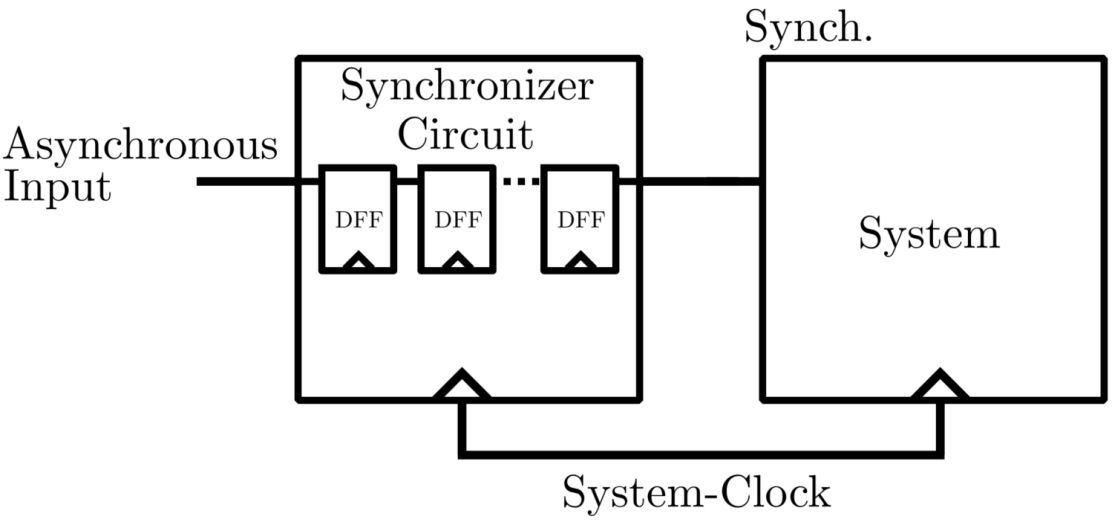


Figure 29: Block Diagram for a Synchronizer

The majority of digital systems have asynchronous inputs since they must interact with external events. External events may be random with respect to internal system activities. The intermittent, random, and nearly untraceable nature of metastable behavior makes these failures particularly troubling and mysterious [9]. Digital system interacting with asynchronous inputs must be constructed such that the metastability failure rate is at an acceptable level. A known approach of deals with asynchronous inputs is to add a synchronization circuit interface between the asynchronous inputs and the sequential circuit [9].

Authors in [9] present different methods of lowering the probability of synchronization failure. Among the presented methods, their evaluations show the use of extended decision time is the most suitable technique for minimizing synchronization failure due to metastability. By utilizing (N+1) cascaded flip flops, the potential metastable output value has the opportunity to resolve to a valid logic level as it is shifted to the next subsequent flip-flop [9]. Resampling with flip flops effectively increases the DFF's resolve time, resulting in a longer MTBF rate. Figure 29 displays a block view of the extended decision time synchronize method. As a designer, it is important to be aware the existence of metastable behavior in digital circuits is unavoidable and it is a consequence of a wide range of failure rates [9].

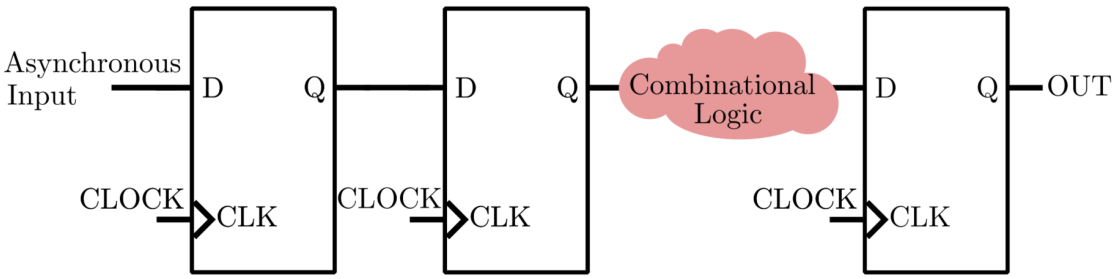


Figure 30: Cascaded DFF

The principle of utilizing cascaded D Flip Flop (DFF) is displays in Figure 30. Once a DFF become metastable, the probability of it actually being metastable after some time decays exponentially. The metastability failure rate of the input value to the second DFF is reduced by adding a second DFF, resulting in a low metastability generation rate to the second stage [18] and a longer MTBF. Regardless, the second DFF could potentially be metastable if the first DFF persists metastable until the next active clock edge. Therefore, synchronization failure due to metastable behavior is not unavoidable even with the well known extended decision time synchronization method.

3.3.2 Unresolved Decision Detection Method [17]

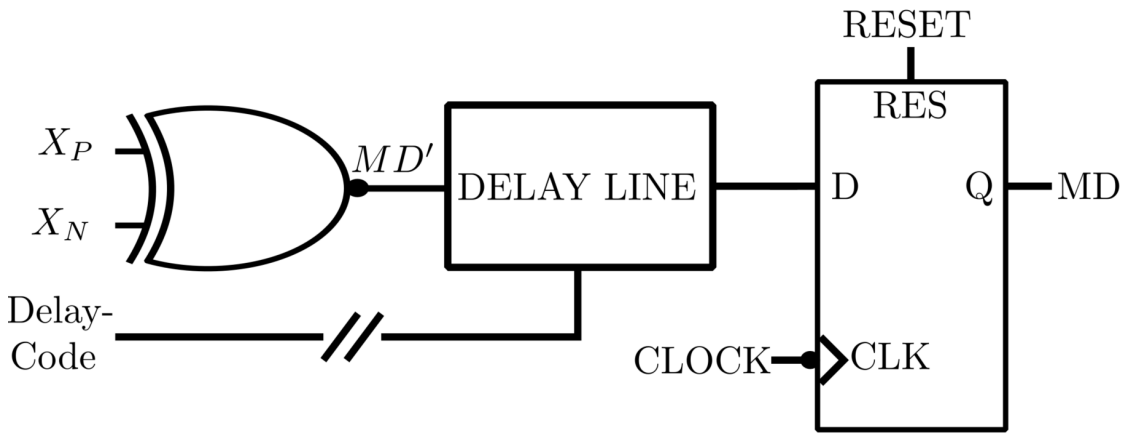


Figure 31: Unresolved Decision Detector Method [17]

Author in [17] present an relative simple implementation of an unresolved decision detector. From a general view, nodes X_P and X_N are cleared ahead of a comparison and stay at the most negative potential, respectively ground. By the presence of a solved comparison, nodes X_P and X_N flip rail-to-rail in opposite polarity, and MD' goes from high to low. The subsequent time constraint delays MD' by a pre-selected amount of time. If the input at the DFF stays low at the DFFs' active clock edge indicates a solved solution. Opposite, by the presence of a logic high input value at the DFF's active clock edge, the MD flag is raised and indicates an input level were within the middle-level at a successive approximation step. A raised MD flag skips the rest of the SAR cycles and the MD bit is inserted as the LSB bit. Therefore, a logic high MD flag indicates an unresolved decision detection was present and the ADC conversion is finished.

This implementation has two different ways of finishing a SAR cycle. It operates as the conventionally binary search algorithm by the presence of a solved comparison. Alternatively, by the presence of a raised MD flag, the rest of the SAR conversion cycle is skipped and MD is added as an additional bit signal.

The interval between the comparator's active clock edge and the DFF's active clock edge, and the additional delay from the delay line sets the exponential time to voltage mapping. The input voltage threshold were the URD detects an unsolved decision is adjustable by controlling the delay line.

This method of realizing a URD circuit does not utilize the principle of extended decision time to minimize the probability of synchronization failures due to metastable behavior. Metastability occurring in the URD is unwanted because it introduces errors around the boundary were the time constraint timeouts the comparator decision. The

authors discuss the probability of the detector itself becoming metastable. They argue as follows, if the setup and hold time for the DFF is much shorter than the delay step of the delay line then the probability of the unresolved decision detection becoming metastable is negligible. Flip flop's time window wearing critical timing combination triggers a metastable event is technology depended.

3.3.3 Unresolved Decision Detection Method [6]

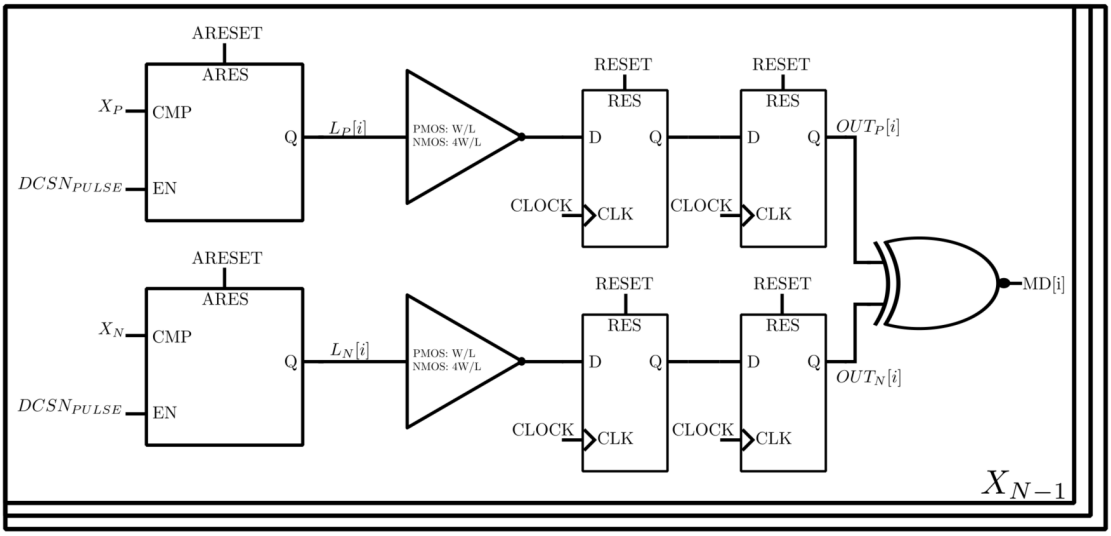


Figure 32: Unresolved Decision Detector Method [6]

This method has similarities in terms of its principle, but the implementation differs greatly. The buffered comparator differential outputs are captured and stored by a decision latch which is transparent during active $DCNS_{PULSE}$. This latch preserves the comparator time to voltage relationship. Subsequent logic treats the captured and stored comparator decision value. Skewed inverters are added to ensure the subsequent synchronization circuit gets a valid logic level even in the presence of noise and device mismatch. The captured differential output value is resampled by two cascaded D flip-flops prior to comparing it with an XNOR gate. The URD flag is raised if OUT_P and OUT_N are identically causing the generated ADC output code to ignore subsequent comparator decisions. The detector's own metastability rate is made significant low by resampling prior to comparing with the XNOR gate, adding only an amount of latency to the ADC output since the detection operation at the next subsequent SAR cycle [6].

It is more expensive considering the required circuit components, but it is fundamental to design circuits such that the probability of failures occurs seldom enough. Potentially metastability in the detector causes misjudgments around the middle-level detection,

and degrade the validity of actually timeout the comparator and realizing an additional bit of resolution in the converter's output. Therefore, this method is further investigated when a suitable unresolved decision detection is to be selected and implemented.

3.3.4 MTBF Evaluation

The essential purpose of this work is to quantify the overhead required to design the comparator with an intrinsically low time constant compared to introducing an unresolved decision detector. The intention of introducing a timeout scheme is to guarantee the SAR cycle to complete and achieve a long MTBF. Equation 14 displays a general method of calculating the MTBF for flip flops and Table 2 describes the equation's variables.

$$MTBF = \frac{e^{(t_r/\tau)}.}{t_{PD} \cdot f_c \cdot f_d} \tag{14}$$

Variables	Description
t_r	Resolve Time
τ	Time Constant
t_{PD}	Propagation Delay
f_c	Clock Frequency
f_d	Data Frequency

Table 2: Variable Description

Figure 33 shows the building block for the time constrained latch and the subsequent resample flip flops employed to increase the detector's own MTBF as the resolve time t_r increases. The unresolved decision detector's MTBF is analyzed from the node X and to D_{OUT}. The available resolve time for the flip flop is approximately a clock period, and were the latch's worst case resolve time is equal to a half clock period. The resolve time assumption is valid as the time window were the input is illegally permitted to change its state is small fraction of the clock period. The logic circuit between the two last flip flops causes a reduced resolve time and it is taken into account. For clarification, Figure 33 does not show the skewed inverter, but it is integrated into the latch for simplify the drawing.

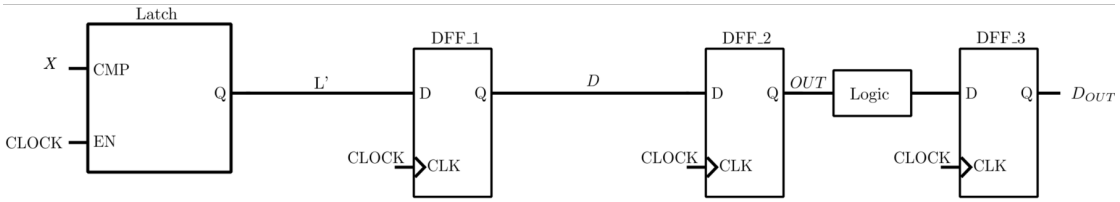


Figure 33: Critical Timing Circumstances in the Unresolved Decision Detector

Equation 15 scheme a theoretical analysis of the detector's MTBF. The key takeaway is the increased resolve time as an effect of the employed cascaded flip-flops. It turns out to be difficult to get a valid estimate of the time window wherein critical timing combination triggers a metastable event in flip flops, respectively the t_{PD}. A try at applying critical timing combinations at the detector input has been investigated, but the yield of the estimated t_{PD} causes ambiguities. Therefore, the theoretical calculation of the detector MTBF is omitted due to the vagueness to get a valid estimate of the unknown t_{PD}. Opposite, the detector's MTBF is expected to be tremendously large by viewing the exponential increase in resolve time.

$$\begin{aligned} MTBF &\approx \frac{e^{(t_{r0}/\tau_0)} \cdot e^{(t_{r1}/\tau_1)} \cdot e^{(t_{r2}/\tau_2)} \cdot e^{(t_{r3}/\tau_3)}}{t_0 \cdot f_c \cdot f_d} \\ &\approx \frac{e^{(½ \cdot T_{clock}/\tau_0)} \cdot e^{(T_{clock}/\tau_1)} \cdot e^{((T_{clock}-T_{Logic})/\tau_2)} \cdot e^{(T_{clock})/\tau_3)}}{t_{PD} \cdot f_c \cdot f_d} \end{aligned} \tag{15}$$

3.3.5 Calibration of the Unresolved Decision Detection Level

As mentioned, this thesis does not encounter a calibration circuit for the timeout. The time constraint circuit asserts the time to voltage decision level, respectively the threshold level between the middle-level, and the high-level and low-level. These decision levels are trimmable by adjusting the observed rate at which the unresolved decision detection flag is asserted, respectively by adjusting the $DCSN_{PULSE}$ active pulse width. The timeout rate should be tuned such it occurs 50 % of the SAR cycles when the input voltage can be assumed to be uniformly distributed and the comparator resolves input voltages at $\pm$ ½ LSB. By setting the timeout to occur 50%, the middle level range is set to $\pm$ ¼ LSB. The timeout rate is adjustable by modifying the active pulse width of the time constraint. The time constraint persists in an inverse proportionality of the decision level. The middle-level decision level diminishes by longer delay and expands by shorter delay. The timeout flag becomes always meta-stable if the time constraint is too short. On the other hand, the timeout flag is rarely detected by the presence of a too long time constraint.

4 Implementation

The implementation of the unresolved detection circuit and integration in a SAR ADC needs to be as simple as possible, but simultaneously reliable to introduce minimum amount of overhead compared to actually designing the comparator with an intrinsic low time constant. This chapter grasps the implementation strategy of the proposed unresolved detector, first at a system level and thereafter in a detailed view. Then, integration of it into a fully differential SAR ADC is given. Design considerations are presented along the way. This chapter views the two implemented integrated circuits, respectively the PVT trimmable tri-level comparator and a differential 8-bit SAR ADC with the adopted PVT trimmable tri-level comparator. The designs are presented together due to some of the implementations are common for both.

4.1 The Proposed Tri-Level Comparator

Figure 34 displays the core structure of a PVT trimmable tri-level comparator. The comparator's differential outputs are fed to an unresolved decision detector. The detector constraint the available comparison time and timeout comparison for input voltages ranging within the middle level. By the presence of a fast enough comparison, the nodes X_{POUT} and X_{NOUT} flip rail-to-rail in response to the differential input voltage. A middle-level detection is asserted if nodes X_{POUT} and X_{NOUT} are at equal voltage potential. Figure 32 views the detector's internal building blocks.

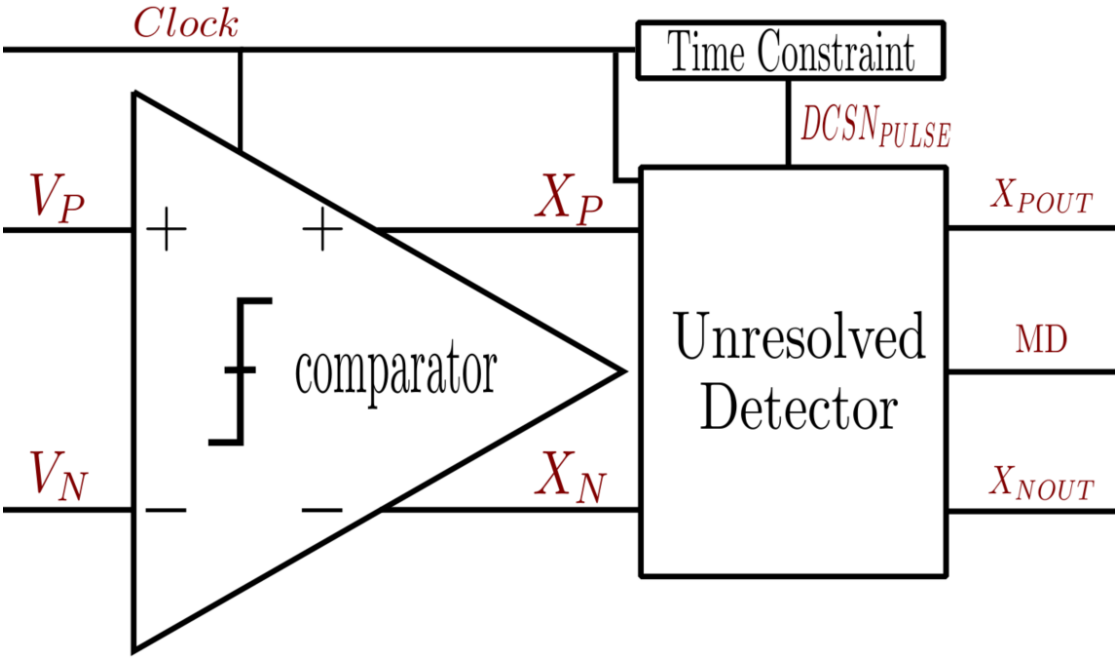


Figure 34: The Proposed Tri-Level Comparator

Figures 35 and 36 shows two cases were the differential input is large and small, resulting respectively in fast and slow comparison. It is assumed that a reset is asserted ahead of the comparison such that internal nodes initially starts in their default mode. For simplicity, only one comparator comparison is sketched although the illustration shows it being clocked multiple times. The figures view the time constrained decision as it propagates through the subsequent detector. The comparator decision is available three subsequent clock periods after it actually began the comparison. The MD flag is asserted if a comparison was not reached within the time constraint period, or else by the presence of a solved solution then the MD flag stays low.

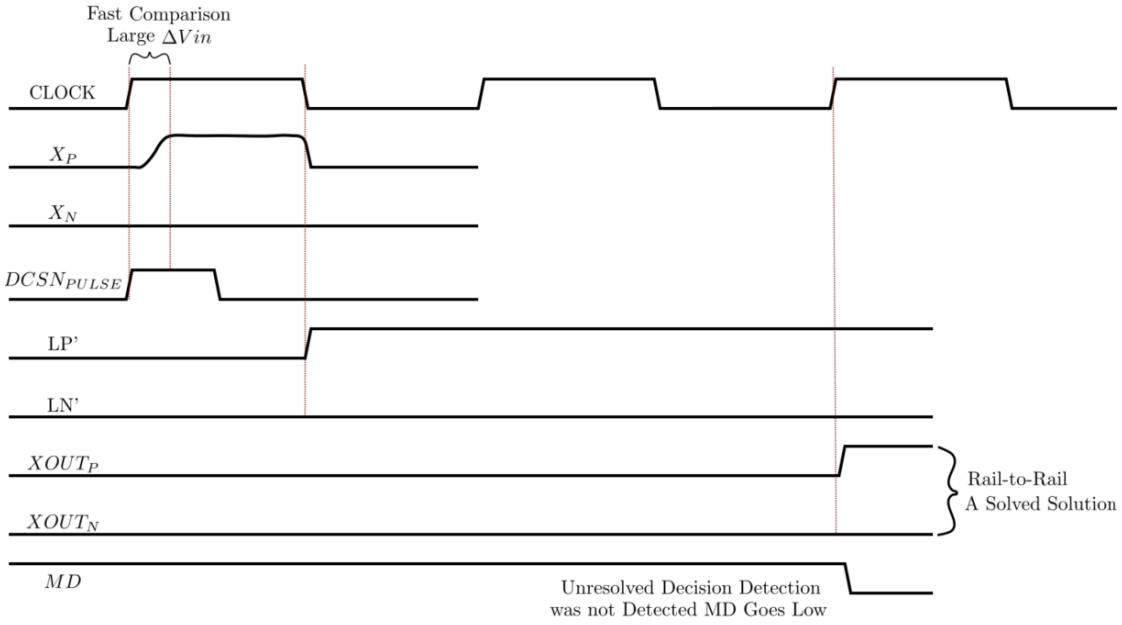


Figure 35: Fast Comparator Compassion, MD Flag is Set Low

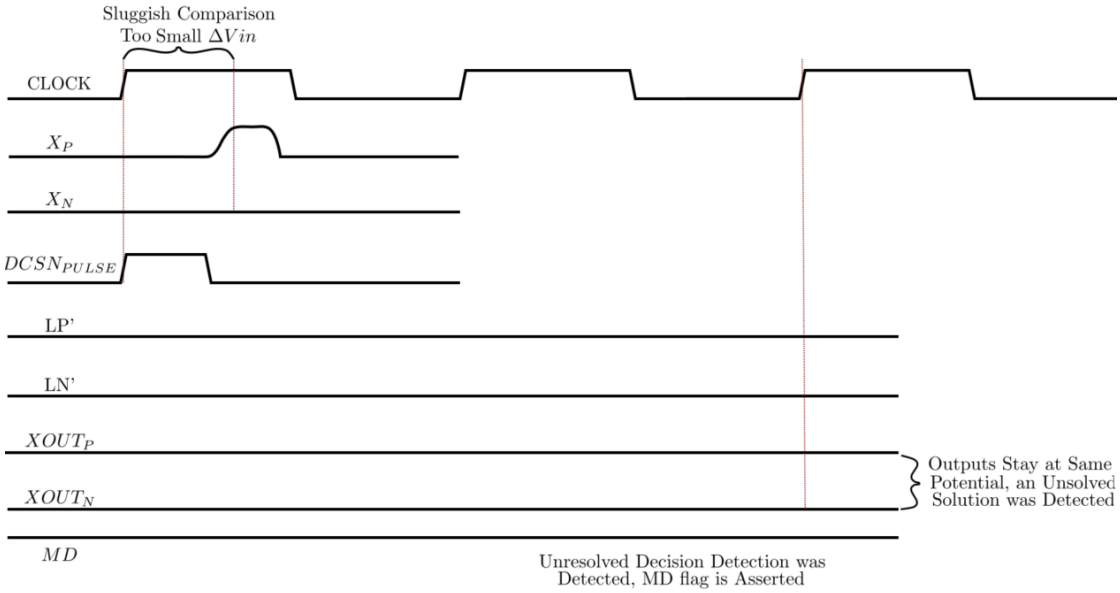


Figure 36: Slow Comparator Compassion, MD Flag is Asserted

The tape-out circuit for this part of the project is as viewed in Figure 34. The structure of the presented tri-level detector is extended when multiple bits are to be solved in a SAR ADC. It is required a detector for each bit to be solved by the comparator as the detector captures and stores the time constrained decision, and evaluates if the output were metastable in retrospect of the comparison. Enabling the detector in sequence driven by the SAR logic accomplishes only the corresponding detector for the bit to be solved is activated. The adopted PVT trimmable tri-level comparator presented in this paragraph is employed in the design of a SAR ADC in the upcoming paragraph.

4.2 Fully Differential SAR ADC with the Proposed Tri-Level Comparator

Figure 37 views a fully differential SAR ADC implementation with the adopted unresolved decision detector to detect when the comparator solves input voltage within middle-level boundary. The unresolved decision detector is employed to consequently minimize the probability of a metastable decision by the comparator entering subsequent circuits. The converter's digital output is driven by the SAR logic in the conventional realization of SAR ADCs. The digital output code is rather fed out from the detector as it carries information concerning a middle-level decision detection was asserted at a successive approximations step. A differential split capacitor structure DAC is employed to equalize the amount of consumed energy for an "up" and "down" switching transition during the successive approximation cycles.

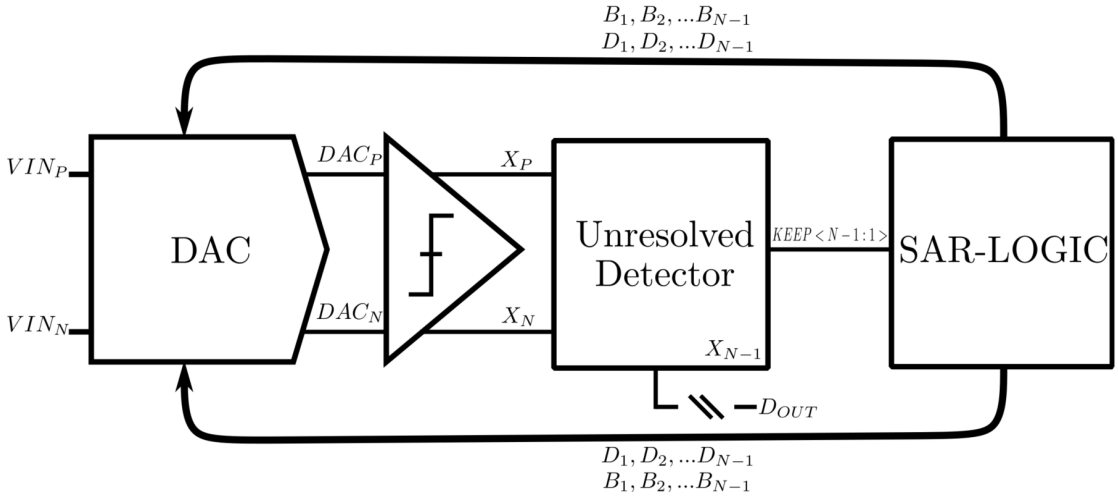


Figure 37: Block Diagram of the Proposed Tri-Level Comparator Adopted in a Fully Differential SAR ADC

Figure 39 displays a detailed block view of the implemented SAR ADC utilizing the split DAC switching scheme and the unresolved detector. Each URD block is enabled separately in sequence, starting with enabling $CYCLE < 0 >$ for the MSB bit determination. By the presence of an unresolved decision flag during the successive-approximation steps, the LSB bit ($D_{OUT} < 0 >$) is set high, indicating the input level was within the middle-level region at a successive-approximation step. E.g. if the input level is within the middle-level region at the second bit determination and MSB were detected as high, the binary output word is then 0b10000001 at an end of conversion for a 8-bit ADC. Figure 38 displays the generated enable signal to each URD during the successive approximation cycle. The bit cycling becomes apparent after a start of conversion (SOC) is set high.

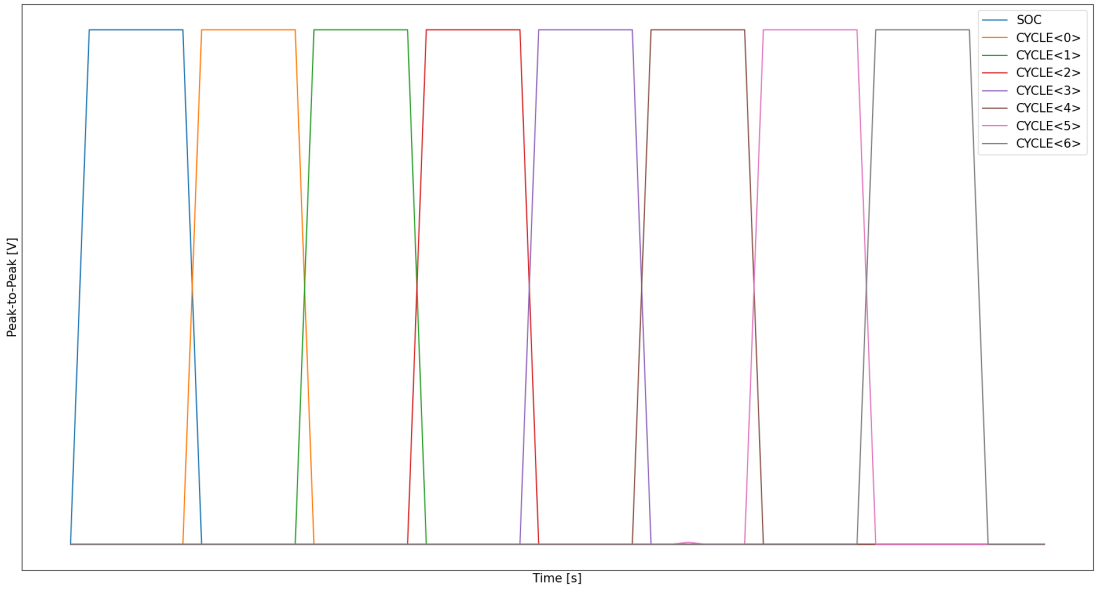


Figure 38: The CYCLE<0:6> Timing Pattern for the Comparator's Bit Determinations

4.2.1 Detail View of the Differential SAR ADC with the Proposed Tri-Level Comparator

Figure 39 displays a block overview of the differential SAR ADC with the tri-level detector. The implemented SAR ADC does not have an external SOC signal to start the ADC to introducing less amount of complexity. The ADC is designed to set the SOC signal internally if a global active low reset is set high and by the presence of a global clock signal. Then, it continuously set the SOC high a half clock period after the EOC flag is asserted. This choice is favored to introduce less amount of overhead in the realization of a prototype tape-out.

The SAR ADC internal memory circuit is designed in the following manner. At an asserted EOC, the binary word from the URD is parallel loaded into a serial shift register. Then, it is shifted sequentially into a 512-bytes large memory shift register at the subsequent SAR conversions. Samples are shifted out from the memory bank by pulling the system clock low and then activating the shift register clock (CLK_{SR}).

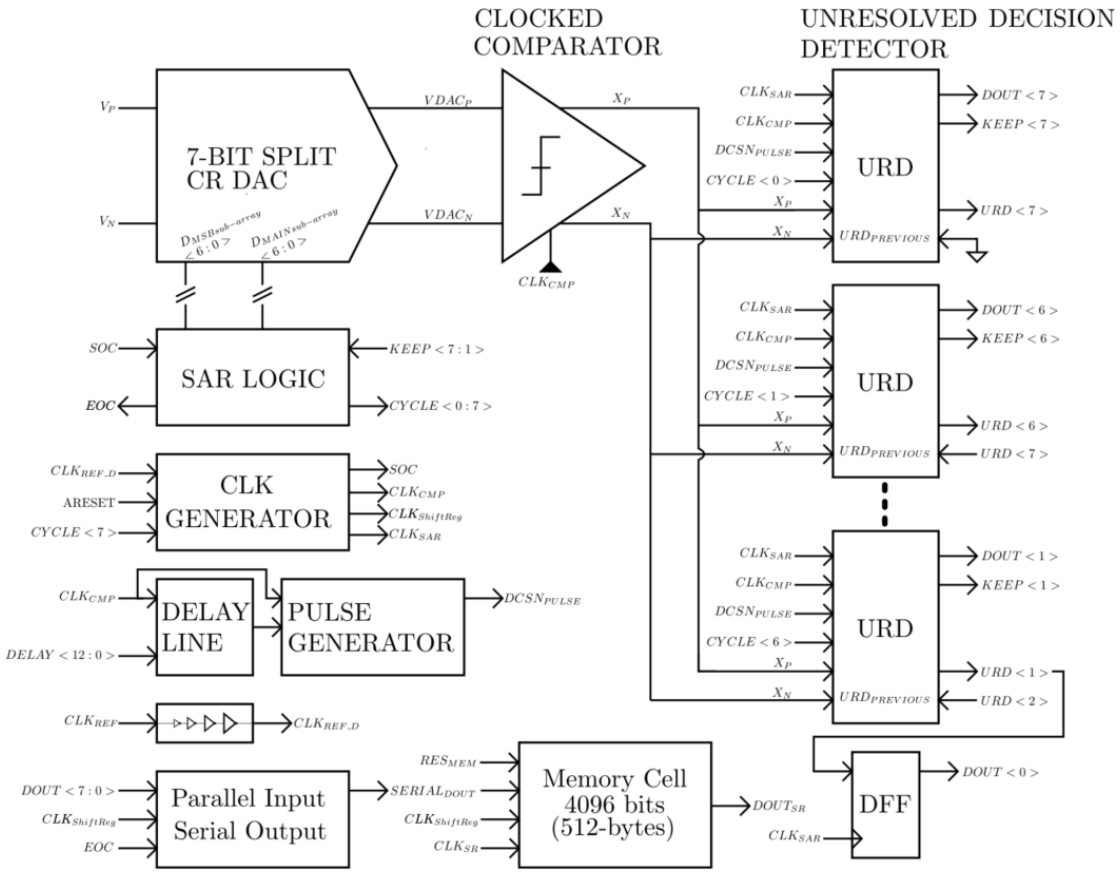


Figure 39: Block Overview of a PVT Trimmable Tri-Level Comparator in Realization of an 8-Bit SAR ADC

The $KEEP < 7:1 >$ signals are a buffered version of the time constrained comparator decision. Each KEEP signal for the current bit determination is sampled by the SAR logic at the subsequent falling edge after the comparator active rising edge.

It is necessary to establish internal clock signals to accommodate the SAR ADC functionally. All internal generated clock signals are driven from a buffered version of the input reference clock to the chip, respectively CLKREF_D. Figure 40 depicts the employed timing signals utilized for different functions. The number of comparisons performed by the comparator is N-1 with the employed tri-level decision detector and it is required N-1 active clock edges which is denoted as CLK_CMP. To shift the N-bits digital word into memory it is required N-clock periods which have given origin to the CLK_SHIFT_REG signal. As mentioned, the employed unresolved decision detector operates in retrospect of a comparator decision and two additional clock periods are required to complete the unresolved decision detection and gives rise to CLK_SAR.

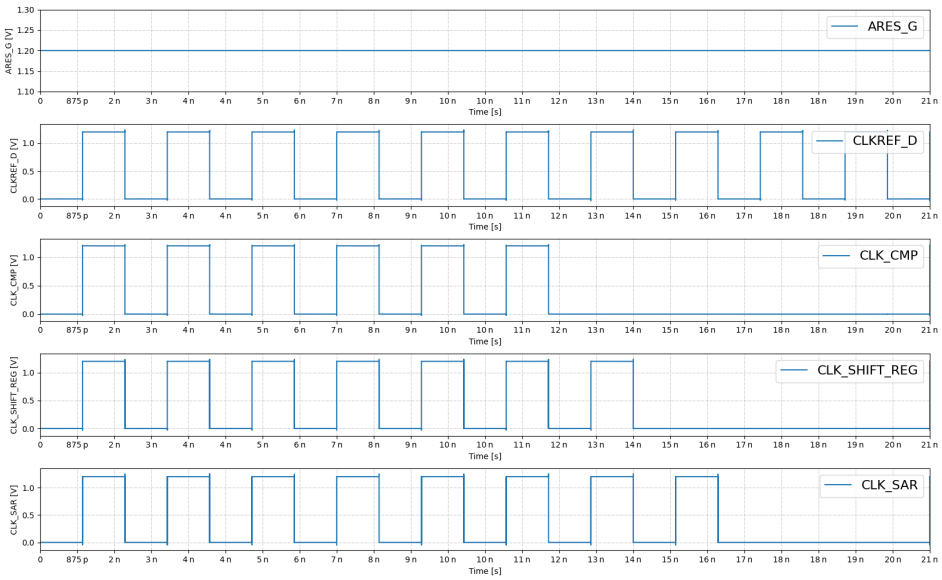


Figure 40: Clock Timing Pattern for the Internal Clock Generation for the 8 bit SAR ADC

4.2.2 Unresolved Decision Detector

Figure 41 displays the block view implementation of the URD presented in [6]. The comparator outputs after the skewed inverters are captured by a decision latch which is transparent at high $DCSN_{PULSE}$ and then nontransparent. Therefore, this latch constraints the available decision time for the comparator to output a valid logic level. Furthermore, the captured value at $L_{P[i]}$ and $L_{N[i]}$ are buffered by utilizing skewed inverters. The output of the skewed inverters are further re-sampled by two cascaded DFF to minimize the probability of the unresolved detector becoming metastable as outlined in paragraph 3.3.1. The captured comparator decision is delayed by two clock periods, adding latency to the digital code. If nodes $OUT_{P[i]}$ and $OUT_{N[i]}$ are at similar logic levels, then $MD[i]$ flip logical high indicating the comparator utilized too long time to complete a comparison and the input level were within $\pm$ ¼ V_{LSB}. Opposite, if nodes $OUT_{P[i]}$ and $OUT_{N[i]}$ are at different logic potential then a solved comparator decision is detected and $D_{OUT[i]}$ flips high at the next active clock edge, and $URD[i]$ stay low. If URD[i+1] is set high, the remainings URD are accordingly set high, signalizing the differential DAC input signal was just too close at a successive-approximation step.

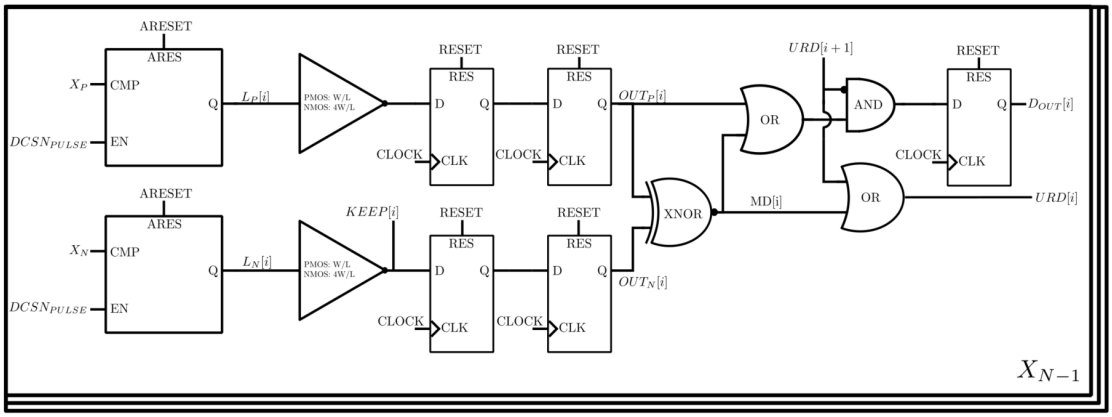


Figure 41: Block View of the Unresolved Decision Detector

4.2.3 Decision Latch

The circuit which applies the time constraint on the comparator decision time is preferred to close its transparency rapidly when the timeout appears. Figure 42 displays the implementation of the latch which preserves the exponentially time to voltage relationship in the comparator decision time by controlling the timeout rate. In the existence of a logical low *ARES*, the latch state is reset to a logic high level at node *Q*. This latch implementation keeps an inverted version of the input node *CMP* when *EN* goes low. It is essential that the latch is reset ahead of the enable *EN* is set as the NMOS M_2 is inevitability to drive the node Q at a logic high level. In fact, in the presence of a logic high *CMP* and *EN*, the latch can only lower the output node *Q*. In the existence of a logic low *CMP* while the enable signal *EN* is set, the output of the two cross-coupled output inverters remain at its initial state, respectively at a logic high level. Two cross-coupled inverts are added to store the latch's decision until an active low reset *ARES* is asserted.

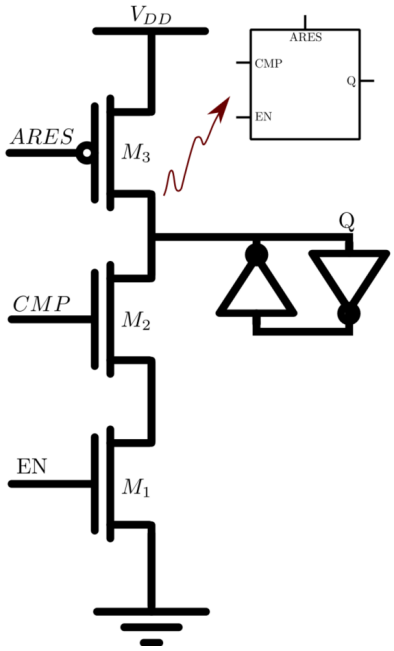


Figure 42: Decision Latch Capturing the Comparator Output During its Active Phase

4.3 StrongARM Latched Comparator

The comparator transistor dimension determines its intrinsic decision speed. The comparator aspect ratio (W/L) reported in Table 3 is the used transistor dimension for this design were the transistor names maps accordingly to as viewed in Figure 7. It is consequently engendered by iterative sweeping different transistor widths concerning realizing a relative small enough time constant.

Transistor	Width per Finger (W)	Length (L)	Multiplier	Finger	Dummy
S_{1-4}	0.25 µm	60 nm	4	1	4
M_{5-6}	0.41 µm	60 nm	4	4	4
M_{3-4}	0.5 µm	60 nm	4	4	4
M_{1-2}	1.9 µm	60 nm	4	5	4
M_7	1.9 µm	60 nm	6	5	2

Table 3: Transistor Dimension of the StrongARM Latched Comparator

The implemented StrongARM architecture is a differential structure. Differential circuits require proper matching between each differential pair. The layout technique common centroid is employed for the differential pairs. Common centroid reduces any linear gradient degradation in the horizontal and vertical, and diagonal dimensions suffered from process variations related to mismatch in production. Dummy transistors are added around each structure such that each unit transistor "sees" the same surroundings. Figure 43 displays the layout of the StrongARM comparator, were the sub-index of each transistor relates to the multiplier number.

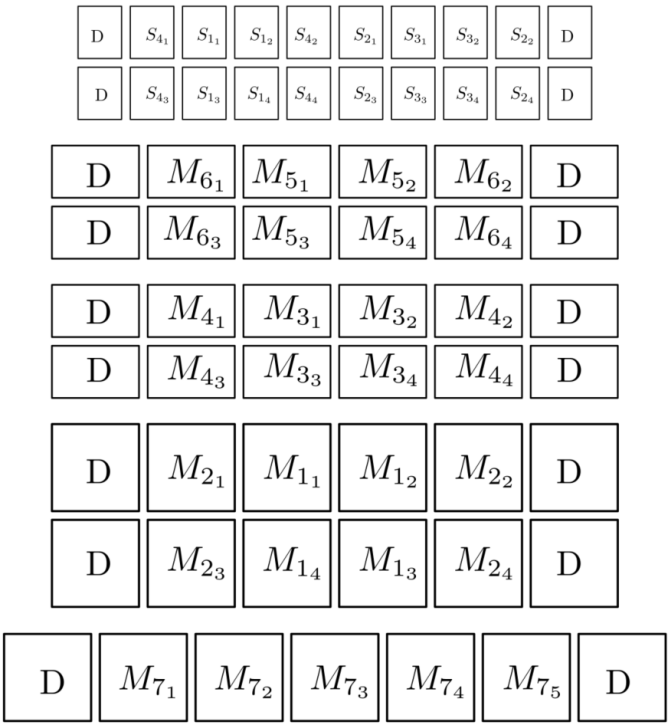


Figure 43: StrongARM Comparator Layout

4.4 Skewed Output Inverters

A technique to minimize the occurrence of unresolved decisions from the comparator is by factorizing one edge over another by adding skewed inverter at the outputs of the comparator. The inverter edge threshold is moved by making the drive strength of the NMOS stronger than the PMOS. The comparator needs to output well defined logically level to switch the output of the skewed inverters. Therefore, if the comparator does output a weak logic decision level, then the output of the skewed inverters stays unchanged. It is fundamental to have a well defined logic level from the decision circuit to avoid the subsequent unresolved decision detector circuit from actually becoming metastable. The skewed inverters ensure the subsequent circuits using the comparator decision gets a well-defined logical level even in the presence of noise and device mismatch and ensuring unresolved decision is always detectable [6]. Apart from that, it delays the comparator decision, but it is acceptable because it just adds a fixed delay in the comparator decision time.

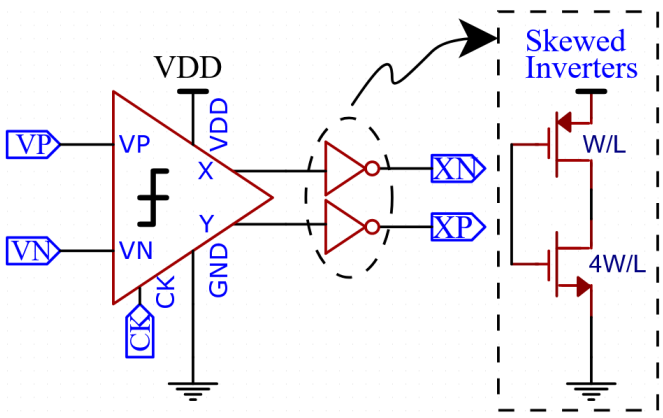


Figure 44: Skewed Outputs

Transistor	Width per Finger	Length	Multiplier	Finger
PMOS	0.25 µm	60 nm	4	1
NMOS	0.25 µm	60 nm	4	4

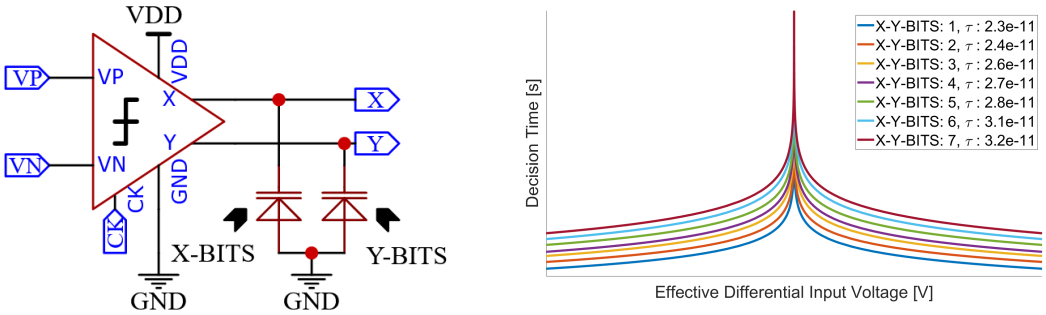
Table 4: Transistor Dimension of the Skewed Output Inverters

4.5 Comparator Output Load

Comparator's decision time is highly dependent on the time-constant τ. By adding additional capacitive load at the output of the comparator is the time constant modulated and it is possible to make the comparator decision slower. The required time resolution demands by increased output load, as the exponential slope rises and falls slower. There is required a certain voltage resolution to distinguish different input voltages.

A certain voltage resolution is directly related to a certain time resolution. Steeper exponential slope requires finer time resolution to preserve a certain input voltage step. Therefore, by adding capacitive load it is possible to increase τ and reduce the required time resolution for the time constraint.

Figure 45b views an increase in the decision time and τ with an increase in the output capacitance at both the X and Y nodes. It is intended to have individual controls signal for the capacitive trimming circuits. Figure 45a views a schematic view of the capacitive output load.



(a) Output Load

(b) Decision Time with Different Output Loads

Figure 45: Comparator Output Load & and its Impact on the Time Constant

4.6 CR Split DAC

The aforementioned DACs are depicts utilizing the bottom plate sampling technique. Consider the bottom-plate sampling, the number of switches that treat the input signal ranging from $VREF_N$ to $VREF_P$ are N for the conventional single ended (SE) CR DAC and 2·N for the investigated Split SE CR DAC. Contrary, a SE CR DAC which employs a top plate sampling technique requires only one sampling switch to treat the input signal. The fundamental difference is whether the input signal is fed to the bottom plate of the capacitors or at the top plate. The input signal must be capable to drive all the sampling switches and it is highly emphasized to employ as few switches as possible that embrace the input signal. The top plate versus bottom plate sampling technique is a layout related trade off. Concerning its advantages related to simplifying the input signal routing, and the number of switches that embrace the input signal is the top plate sampling technique employed for this work. Figure 46 depicts a differential realization of a Split CR DAC utilizing top-plate sampling technique.

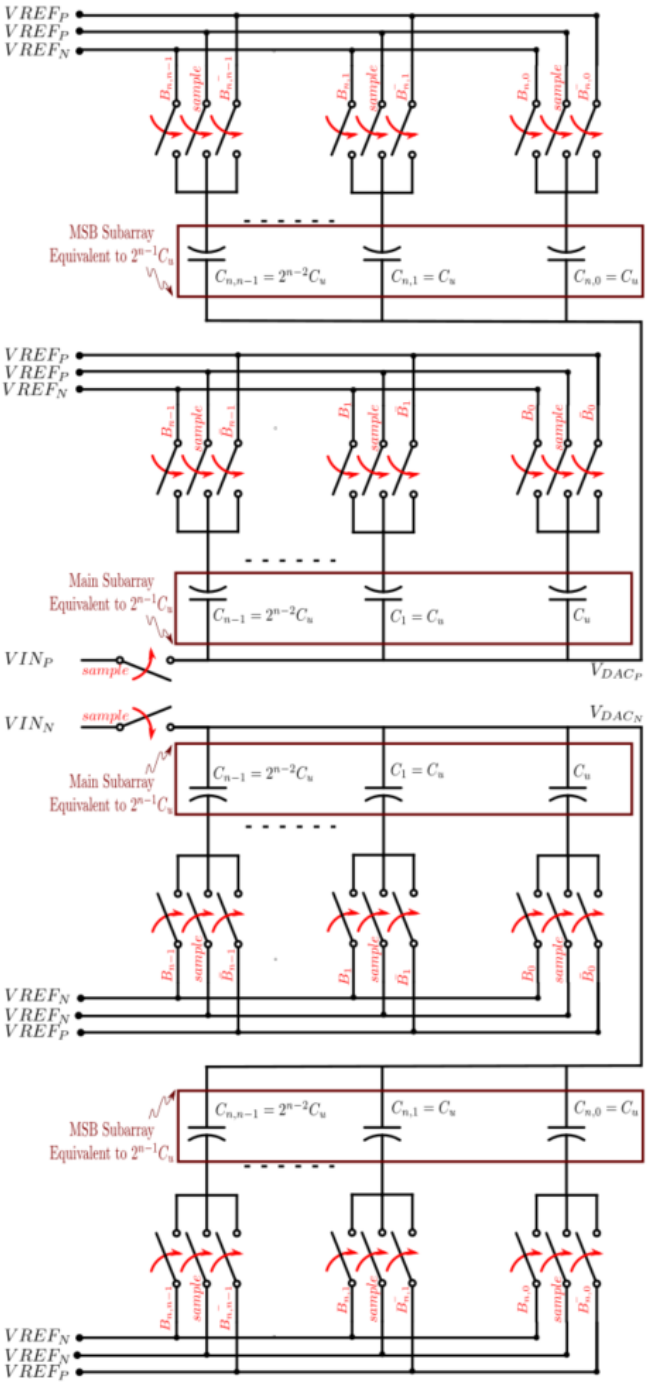


Figure 46: N-bit Fully Differential Split CR DAC Utilizing the Top-Plate Sampling Technique

4.6.1 Split CR DAC Layout

The primary source of linearity degradation in terms of DNL and INL in a conventional SAR ADC is due to mismatches between binary weighted capacitors in the DACs' capacitive array. A rule of thumb, larger devices match better, but the compromises are larger power consumption and die area. Compromise concerning selecting an appropriate unit capacitor size is encounted in paragraph 4.7. It is essential to apply the known methods to lower the systematic mismatch errors between capacitors related to the physically drawing of the capacitor array. Therefore, the common centroid method is utilized and a double dummy ring is added around the structure such that capacitors "sees" the same surrounding. The main subarray and MSB subarray are cross coupled across its common intersection point, making the capacitive array more resilient to systematic process mismatch suffered from production. Figure 47 displays the layout drawing for this work.

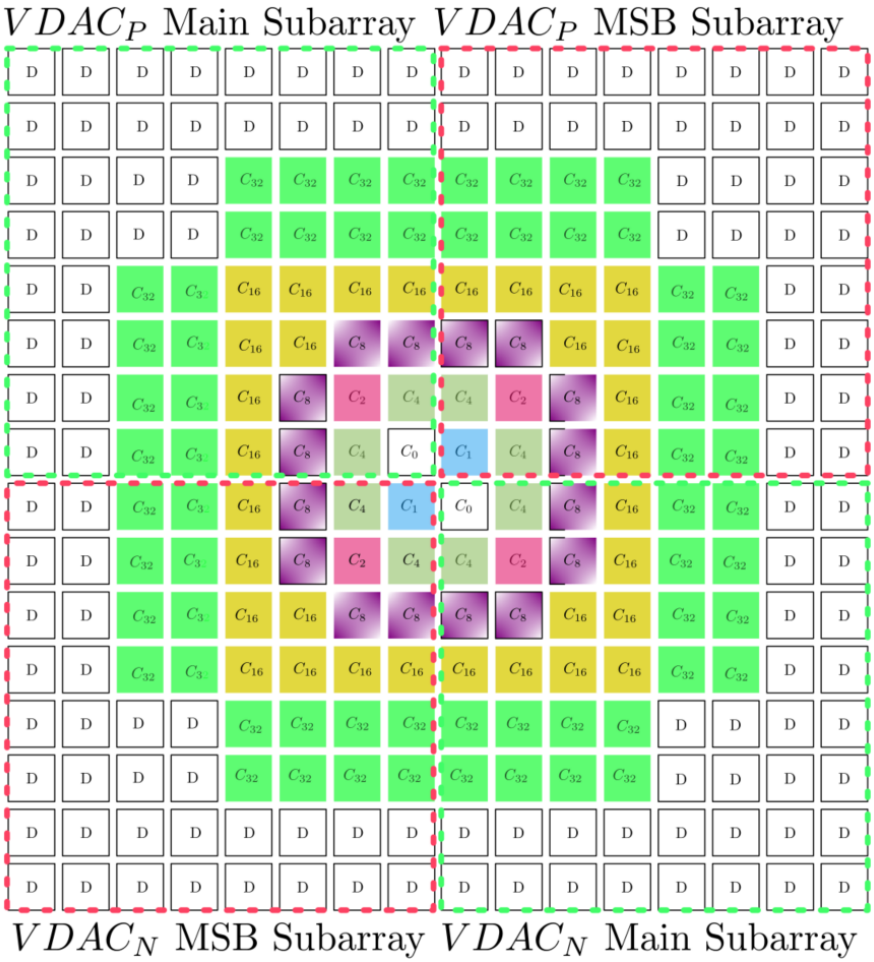


Figure 47: 7-bit Split CR DAC Layout apply the Common Centroid Drawing Principle

4.7 DAC Mismatch

A capacitive realization of a CR DAC is composed of an array of binary weighted capacitors. The DAC output voltage changes accordingly to the switch configuration. Shifting between two adjacent digital codes causes the output voltage to ideally increase by V_{LSB}. Imperfections related to mismatches between the capacitors and parasitics causes the output voltage to change by a different amount. Therefore, any mismatch between the capacitors cause linearity degradation in an ADC and degrades the attainable ADC resolution. The primary source of DNL and INL errors in a SAR ADC are mismatches between the capacitors in a capacitive DAC [10]. There is an inverse proportionality between the capacitor size and how well the component matches. The numerical analysis method presented in [3] is employed to quantify the design trade-off between capacitor size versus acceptable DNL and INL. This method is used to estimate the impact of DAC non-idealities on the ADC static nonlinearities [3].

Figure 48 displays the classic realization of a CR DAC and the corresponding mismatch model. The capacitance of each binary weighted capacitor is given by the binary weight times the unit capacitance and additionally stray capacitance between the capacitor's top plate and the switch's bottom node.

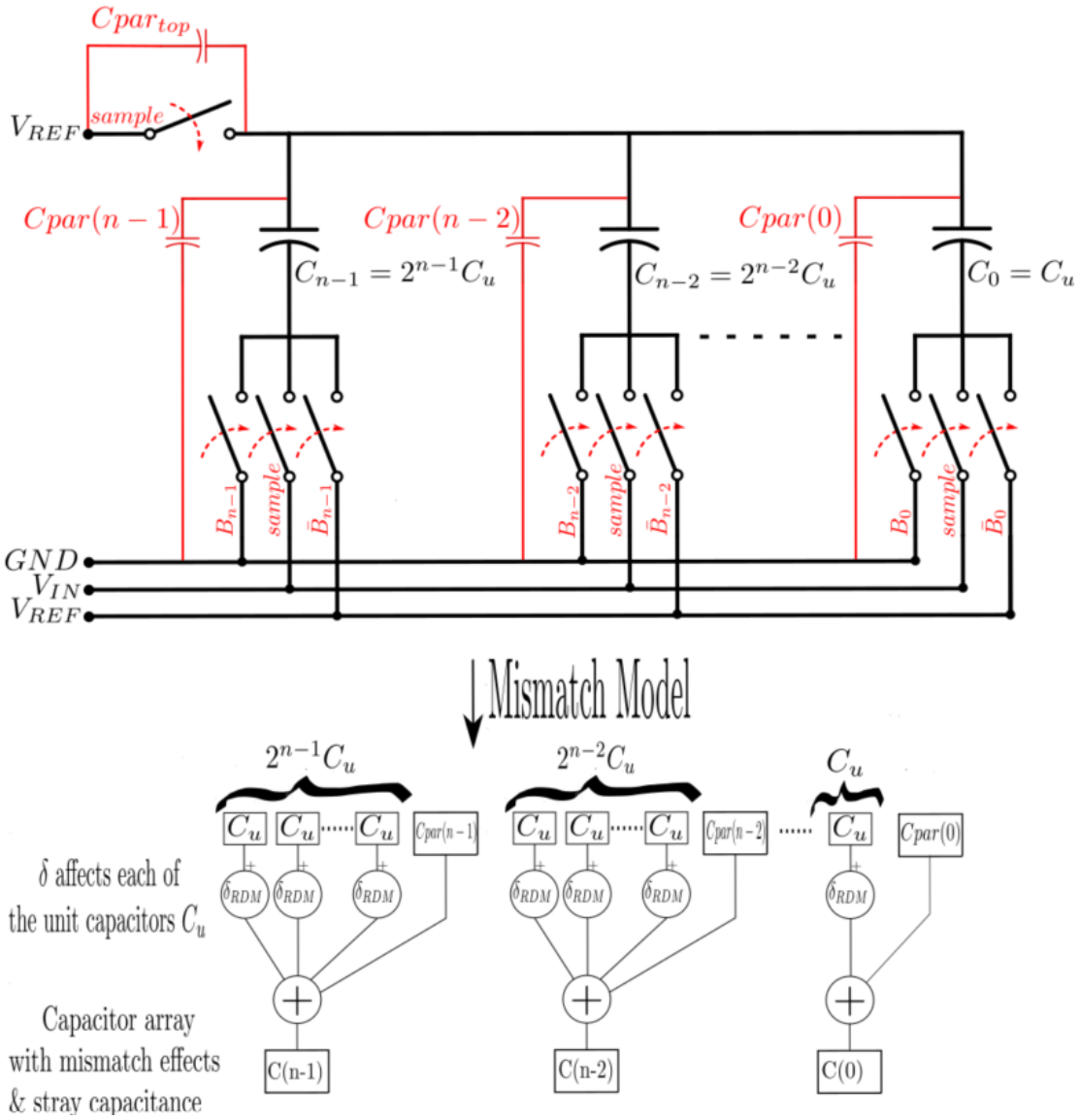


Figure 48: The Classic CR DAC and an Mismatch Analysis Model

The presented mismatch model in Figure 48 illustrates the effects of mismatch between the unit capacitors causing the binary weighted capacitors to deviate. The model adds normal distributed random variations δ_{RDM} based on the process mismatch for a certain unit capacitor C_u. The random variation adds statistical variations based on the estimated pelgrom mismatch coefficient. As a result, the capacitor weight $C(n-1), C(n-2)...C(0)$ is modulated with a random normal distributed variation for given process technology and unit capacitor C_u. The capacitor array is further employed to estimate the required unit capacitor size for an acceptable introduced linearity degradation.

4.8 kT/C Noise

Capacitors introduce thermal noise in a DAC and it adds static mismatch between the capacitors. The best-case rail-to-rail signal swing amplitude is considered when the impact of kT/C noise related to static mismatch is analyzed. By employing the RMS value of the best-case signal swing is $V_s = (V_{REF}/2)/\sqrt{2}$ formulated. Furthermore, the noise source V_n is equal to $\sqrt{kT/C}$. Equation 16 formulates the power of both V_n and V_s. Noise increase by an factor $\sqrt{2}$ and signal by a factor 2 in a differential case. By using the definition of SNR yields Equation 18.

$$V_n^2 = \frac{kT}{C}, V_s^2 = \left(\frac{V_{REF}}{2\sqrt{2}}\right)^2 \tag{16}$$

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{\text{signal power}}{\text{noise power}}\right)\text{[dB]} = 10 \cdot \log_{10}\left(\frac{V_s^2}{V_n^2}\right)\text{[dB]} \tag{17}$$

$$SNR = 10 \cdot \log_{10}\left(\frac{C\,V_{REF}^2}{8\,k\,T}\right) \tag{18}$$

By solving Equation 18 consider the capacitor C yields:

$$C = \frac{2^{\frac{\text{SNR}+30}{10}} \cdot 5^{\frac{\text{SNR}}{10}} k\,T}{V_{REF}^2} \tag{19}$$

4.8.1 Mismatch Versus kT/C

Both terms introduce non idealities in the ADC and limit the effective number of bits (ENOB) and must be analyzed. The aforementioned kT/C analysis is related to the total capacitance in the array and it is important to bear in mind. Contrary, mismatch related to process variation is related to one unit capacitor. To compare the impact at SNR it is necessary to scale the required process mismatch capacitor with a factor of 2^N to have the same assessment basis. By back relating the required capacitor size (Equation 18) for both the kT/C and mismatch analysis is the most dominating part considered when C_u is to be chosen.

With the number of bits set to N=8, yield an SNR related to quantization noise in an ideal ADC equal to be approximately 50dB. The following kT/C static mismatch is calculated by considering the worst-case DAC reference voltage.

$$C = \frac{2^{\frac{50+30}{10}} \cdot \quad 5^{\frac{50}{10}} \cdot \ 1.38 \cdot \ 10^{-23} \cdot (273.15 + 70)}{0.9^2} \approx 4.68\,\text{fF} \tag{20}$$

Figure 49 shows linear degradation in the DAC's transfer function caused by mismatch between unit capacitors in a capacitor array. It is constructed by extracting the process mismatch for a certain unit capacitor from Monte Carlo simulation. Then modulate mismatch variation between unit capacitors and sum the contribution into its associated binary weighted capacitor. This process is depicted in Figure 48. Further, evaluate the impact in terms of DNL and INL suffered from random normal distributed variation between unit capacitors. The appropriate capacitor size is evaluated to be 20.01 fF by accepting DNL and INL error contributions less than 0.5 LSB.

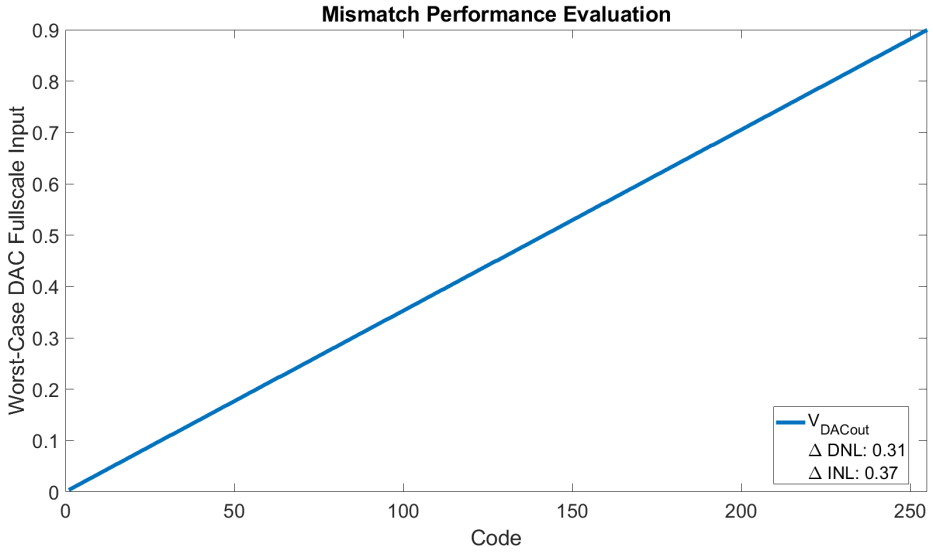


Figure 49: Mismatch Analysis

The dominating static nonlinearities are suffered from static mismatch between unit capacitors. The DAC is implemented with an unit capacitor of 20.01 fF.

4.9 Modified SAR Logic for the Employed Split Capacitor CR-DAC

The investigated Split CAP CR-DAC switching scheme to improve the overall DAC power consumption operates differently from the conversationally DAC and require a modified SAR algorithm. Rather than test each bit in sequence starting with the MSB bit, the split scheme requires two sets of control bits to manage the main subarray and the MSB subarray. Figure 50 displays a block view of its realization. As explained in paragraph 2.6, the MSB subarray capacitors are attached ahead of the first successive approximation step and the main subarray capacitors remain disconnected. Therefore, all control bits in the MSB subarray need to be activated at the first successive approximation step. At the active edge of a *SOC*, the flip flops for the MSB subarray are set high and the flip flops for the main subarray are reset which set the bits to a logic low

state.

The reader may wonder if the flip-flops states influence the MSB subarray and main subarray during the sampling phase, respectively by the presence of a *SOC*. By adding additional switch logic it is ensured that the sampling switches have higher priority by the presence of a logic high *SOC* signal, ensuring the whole capacitor array is connected during the sampling phase. In absence of a *SOC*, the flip-flops manages the capacitor's state and change the binary weighted capacitors with emphasis of minimizing the error between $VDAC_P - VDAC_N$.

The residual most significant bit to be determined is re-evaluated after a comparator comparison. The MSB subarray flip flop remains logic high by the presence of a high *COMPARATOR OUTPUT* and the main subarray flips high. Opposite, by the presence of a logic low *COMPARATOR OUTPUT*, both current control bits evaluation changes to logic low. The modified SAR logic version requires more flip-flops, but its advantages defend its budget compromise considering the reduced DAC power dissipation.

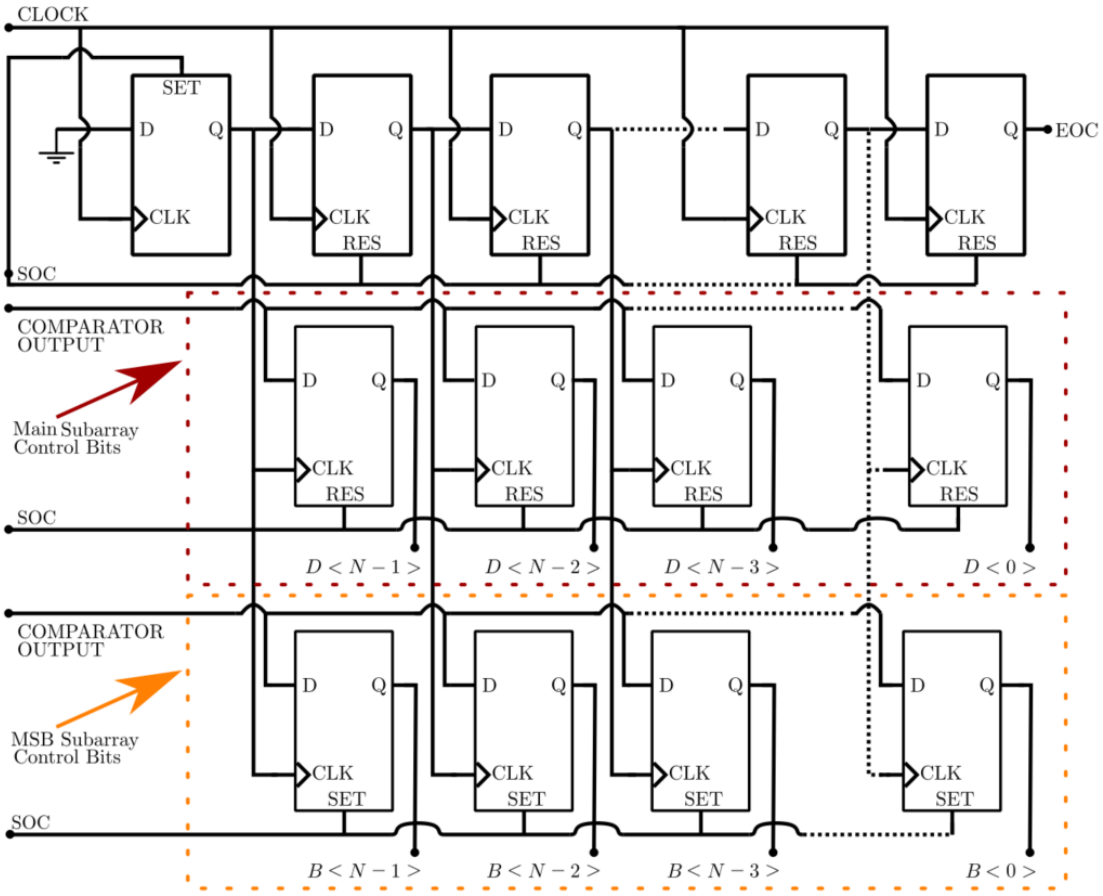


Figure 50: N-Bit Split-CAP Flip-Flop Based SAR-Logic

4.10 SAR ADC System as a Whole

Figure 51 displays a block view of the implemented SAR ADC were all components are encountered in the previous paragraphs. Clock signals have been omitted for this figure, but appear as mentioned in Figure 39.

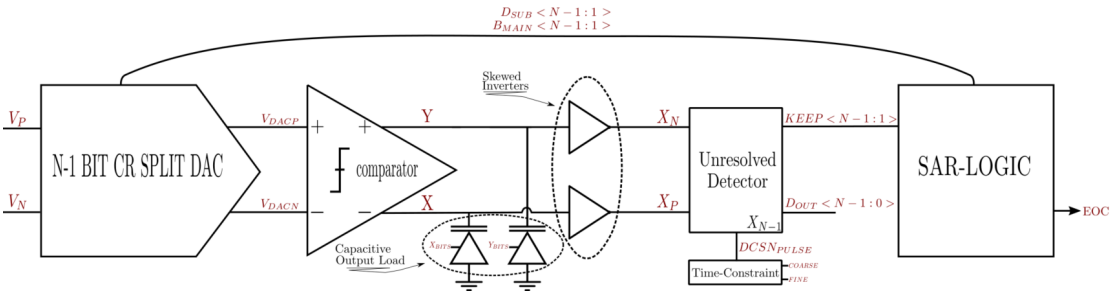


Figure 51: Implemented SAR ADC With the Tri-level PVT trimmable Comparator

4.10.1 Layout

Figures 52 and 53 shows the layout of the implementation of the SAR ADC with the employed PVT trimmable tri-level comparator and the PVT trimmable tri-level comparator to the right side of the figure.

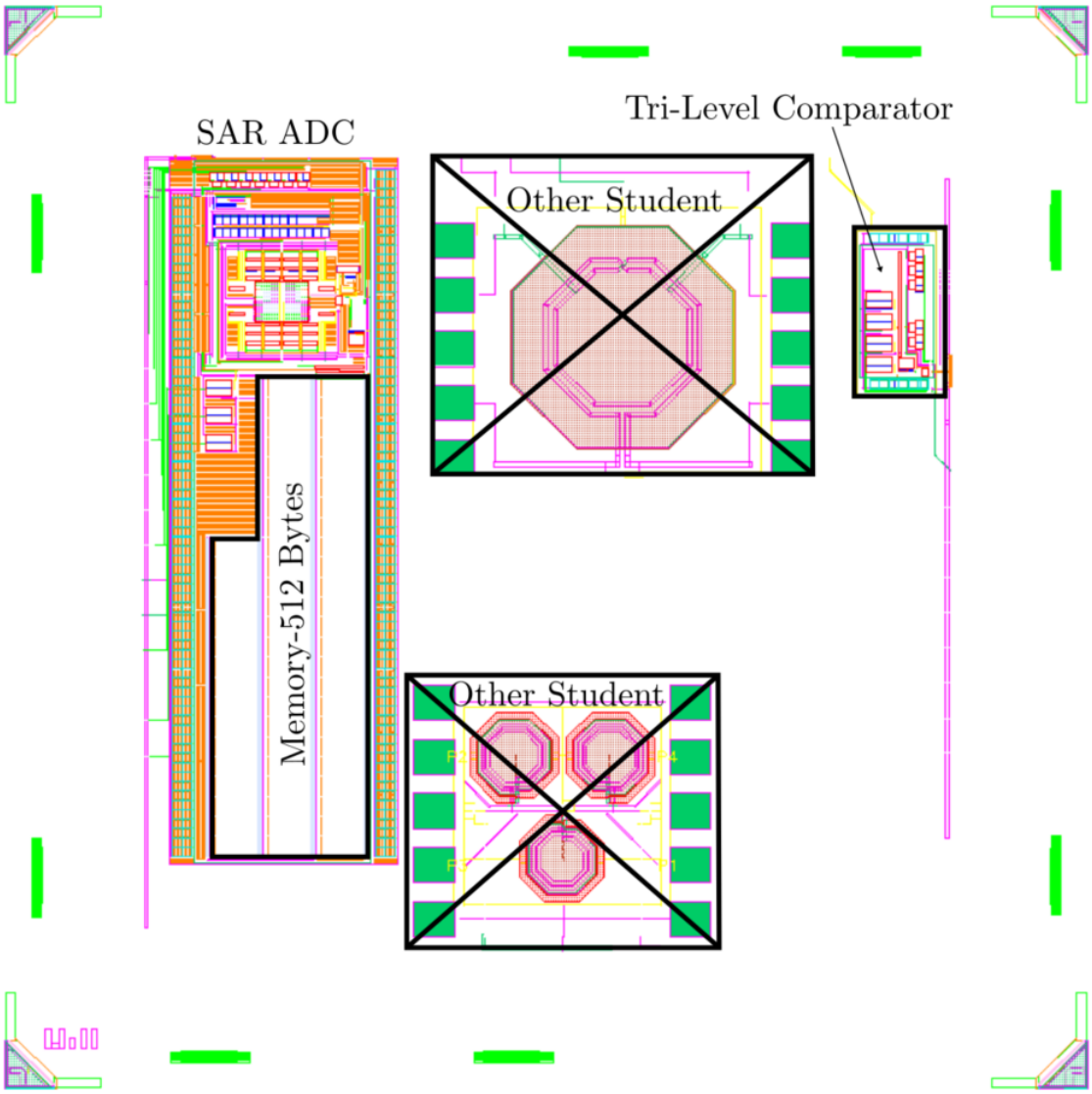


Figure 52: Implementation in the Pad Frame

4.10.2 Enlarged View of the Layout

Figures 52 and 53 displays an enlarged overview of the aforementioned SAR ADC.

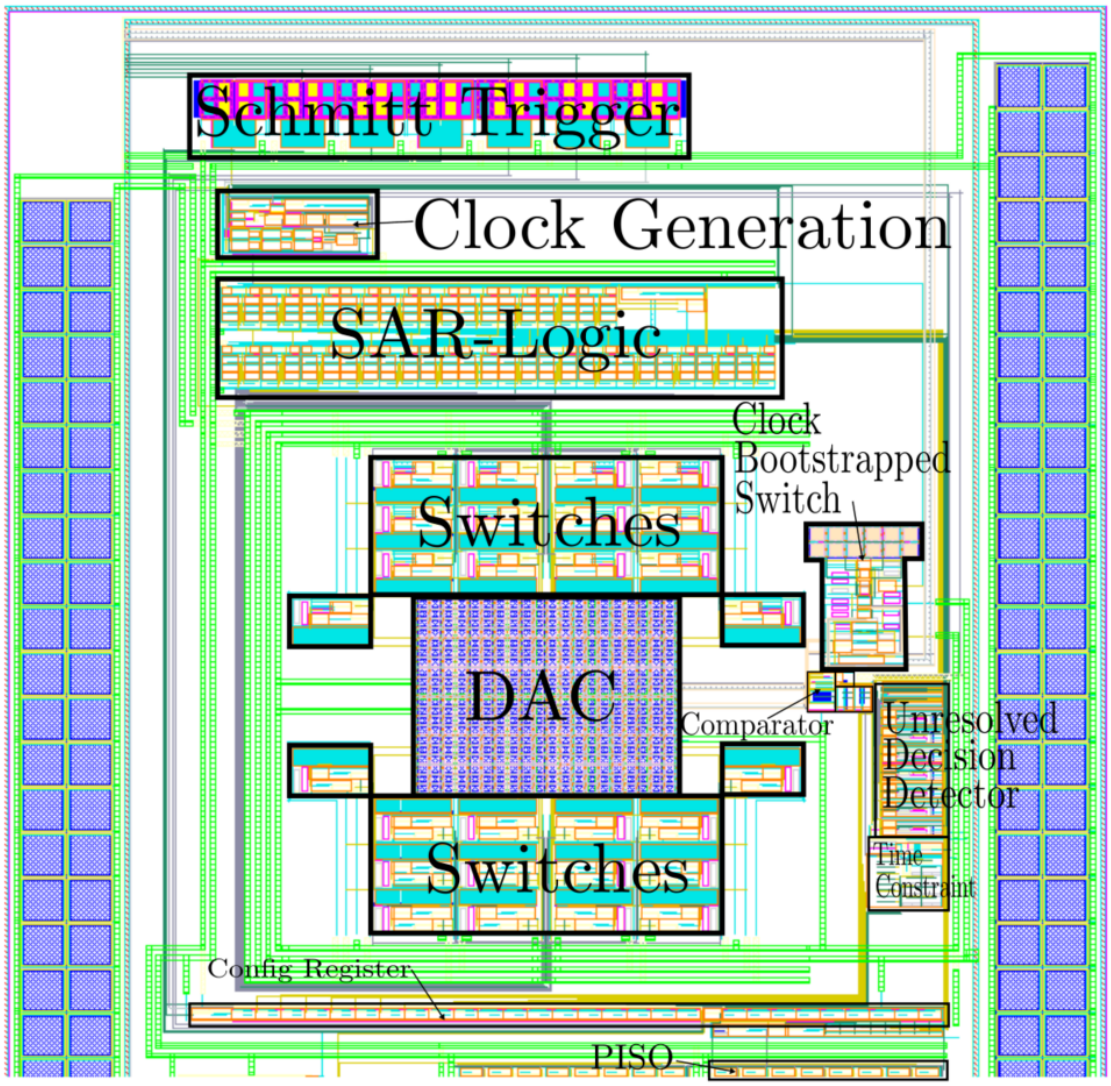


Figure 53: Enlarged view of the SAR ADC with block Descriptions

4.11 Layout Specific Consideration

This paragraph presented some design strategy for the layout and elaborations of why these layout techniques are taken into account.

4.11.1 Schmitt Trigger

All digital inputs have a Schmitt trigger in cascade with a fanout buffer between the chip's pads and before signals enter into the chip. The Schmitt trigger is employed to ensure potential noise and short-term supply drop does not change the present digital input value to the chip. Schmitt triggers adds hysteresis such that the input must move further above or below the ½ V_{DD} threshold to flip digital value.

4.11.2 Output Buffer

Internal signals in the chip which is routed to pads must be capable to drive load such as coupling between external pad and Printed Circuit Board (PCB). All digital signals are buffered ahead of the pads. The buffers are constructed to drive a load of 10 pF to 20 pF, and accommodate a rise time and fall time approximately 10% - 20% of the frequency of the output signal.

4.11.3 Shield

A shielded "box" around the routing traces to enclose the inputs from the surroundings is employed. The shield is added to reduce the impacts of electrical noise and electromagnetic interference at the analog input signal. Figure 54 display an illustration of the differential input pairs enclosed in a grounded shield.

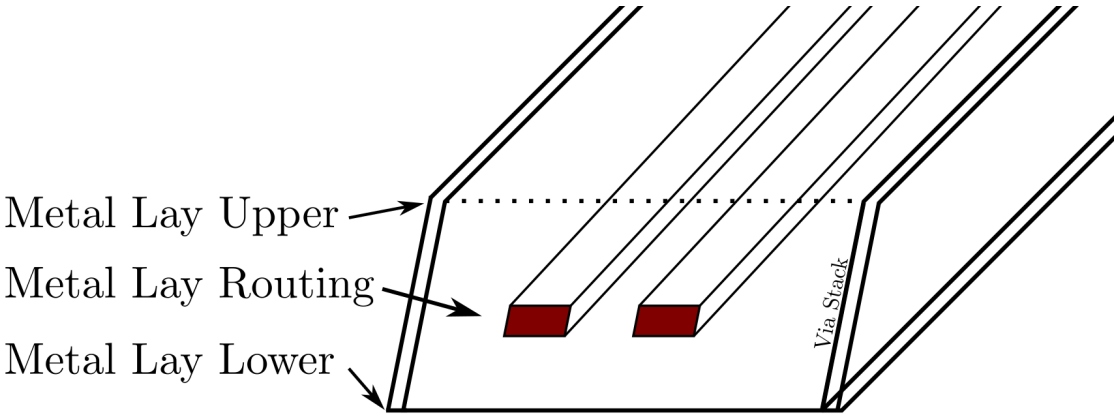


Figure 54: Shield Around the Differential Analog Input Pairs

4.11.4 Substrate Isolation

Both IC designs are isolated from the global substrate by enclosing the circuit design within a P-SUB guard ring with an N-WELL guard ring. Isolating the circuit design from the global substrate avoids intentionally shorted substrate from other designs on the same wafer shorts the local substrate, and additionally reduces cross talk noise to couple into the local substrate. Figure 55 scheme the structure of the local substrate.

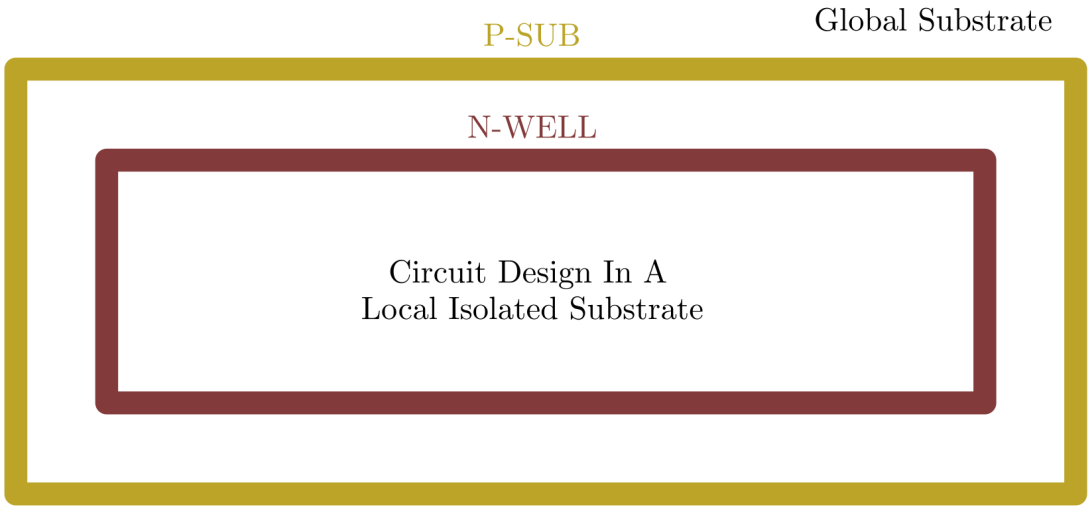


Figure 55: Isolating the Circuit Design in a Local Isolated Substrate

4.11.5 Decoupling Capacitor

Power supplies are undesirably noisy and to mitigate the noise impact is unused layout areas covered up with decoupling capacitors for all power supplies used in the design. The used decoupling capacitor is a thick oxide capacitor which have less leakage, respectively a nmoscap_25 with a W/L ratio of $\frac{16\,\mu m}{16\,\mu m}$.

5 Test and Validation

The purpose of this chapter is to validate the implemented circuit design at simulation level. This chapter first scopes the results of simulation, analysis, and explanation of design considerations. Simulation is used to quantify the performance of the time constraint circuit. A simulation model is developed for extracting the the comparator exponentially decision time behaviour. Thereafter, simulations of the adopted PVT trimmable tri-level comparator in a fully differential SAR ADC is presented. Simulation results are gathered by simulating the circuit with post layout extraction (PEX) at the top level, including physical information about parasitics as resistance and capacitance from the physical layout. If PEX is not brought into account, then else is specified.

5.1 Simulation & Validation of Time Constraint Circuit

Table 6 encounter the simulation setup utilized for the PVT trimmable tri-level comparator. The number of random normal distributed input values applied to the comparator is 1200, resulting in the same amount of measured comparator decision time points. Table 5 outlines the employed tools for simulation and data analysis.

Equipment	Description
Matlab	Results Analysis and Generating Plots. Version: 9.11.0.1809720.
Cadence	Chip Design, Simulation, Post-Layout Extraction, and Layout. Version: IC6.1.7-64b.500.15.

Table 5: Simulation & Analyse Tools

Variables	Description
T_{CLK}	1/0.5 GHz
V_{DD} & Temperature	Follows the Specified Values for the PVT Corners in Table 7
Number of Simulation Points (Periods)	1200

Table 6: Simulation Variables

Corner	Specification
Best-Case (bc)	P = fast–fast (ff) V = 1.1*V_{dd} T = 0 °C
Nominal (nom)	P = typical-typical (tt)) V = V_{dd} T = 27 °C
Worst-Case (wc)	P = slow-slow (ss) V = 0.9*V_{dd} T = 70 °C

Table 7: Supposedly Extreme Edges of PVT Corners

5.1.1 Time Constraint Simulation and Estimation

Equation 8 describe the exponential time to voltage relationship in the comparator decision time behaviour. This equation have three different unknowns (τ, V_{offset}, and t_d) which must be estimated to calculate the decision time for an arbitrary input voltage. There are different methods of estimating the unknown variables. The most straightforward methods rely on utilizing circuit simulations to estimate the unknown coefficients that best coincide with the exponential time to voltage characteristic. Alternatively, a program solver searching for the minimum error of a problem is applicable. Both mentioned methods are investigated, but the program solver is utilized due to its simplicity considering estimating the best fitted unknown coefficients.

The Matlab® utility function fminsearch is an algorithm that searches for the minimum of an problem for an unconstrained multivariable function by initializing the functions with an initial guess for the unknowns to be estimated. The fminsearch algorithm for this optimization problem viable to employ as ones have a certain knowledge about which region the unknown values are located.

Equation 21 schemes the estimating problem to be solved. The error function minimizes the error between the simulated points and the equations unknowns to be estimated. Function $T_{DCSN(sim)}$ is extracted from circuit simulation and function $T_{DCSN(calc)}$ is the function with unknown values to be estimated. The simulated decision time values ($T_{DCSN(sim)}$) are gathered by applying randomly normal distributed input values with $\sigma = \pm 2 - 3\ V_{LSB}$ around the comparator effectively differential input voltage, and measure the time difference between the comparator active clock edge and when the differential outputs cross the ½ V_{DD} threshold. The testbench of this setup is displayed in Figure 56. Equation 22 is executed in Matlab® which outputs the unknown coefficients (τ, V_{offset}, and t_d) that best coincide with the comparator's exponential decision time behaviour. Appendix A shows the implementation of the estimation problem.

$$
\begin{aligned}
errfun\_msq &= \sum_{n=1}^{n} \left(T_{DCSN(sim)} - T_{DCSN(calc)} \right)^2 \\
&= \sum_{n=1}^{n} \left(T_{DCSN(sim)} - (-\tau \cdot log(|V_{in} - V_{offset}|) + t_d) \right)^2
\end{aligned} \tag{21}
$$

$$
\tau, V_{offset}, t_d = fminsearch(errfun\_msq, [\tau, V_{offset}, t_d]) \tag{22}
$$

5.1.2 Comparator Decision Time Extraction

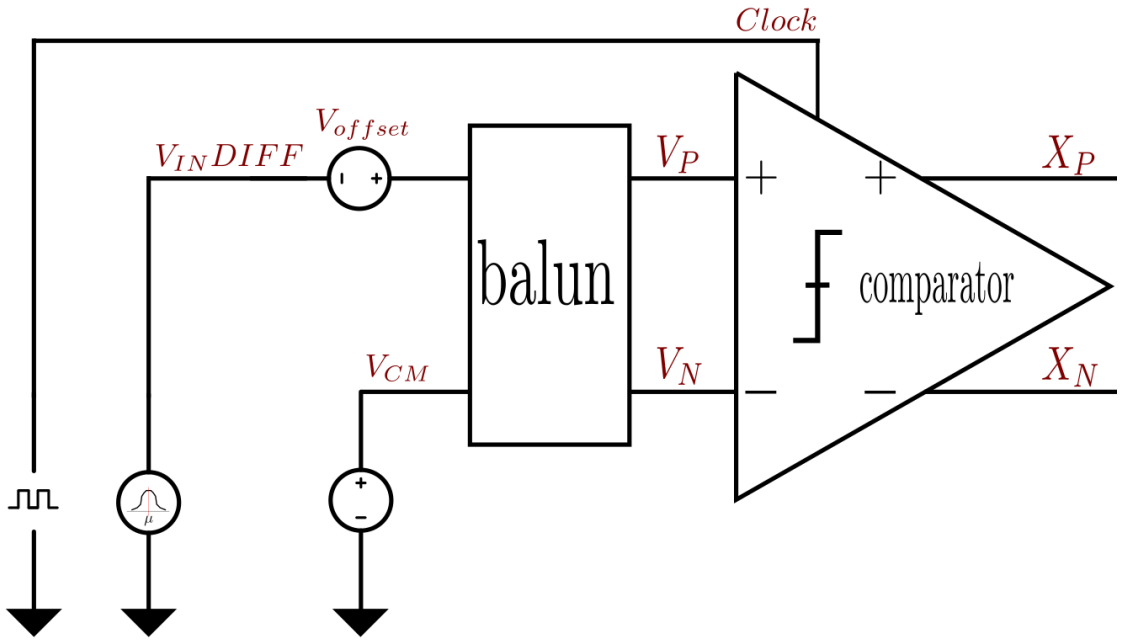


Figure 56: Test-bench of the Comparator Decision Time Simulation

Figures 57 to 59 views the simulated decision time and the calculated decision time across PVT variations. Decision time points are calculated by inserting the estimated unknowns and the simulated input values in Equation 8. The blue scatter points shows the simulated decision time and the orange scatters points displays the calculated decision time points. Figures 57 and 58 views good coherence between the simulated and calculated points. Deviations is present in the *wc* corner as viewed in Figure 59 which is discussed later on.

It has been observed that the initial guess values applied to the fminsearch algorithm affect the coherence between the simulated points and the calculated points. Meaning, the deviations between the simulated points and the calculated values inserted in the exponential decision time equation. The obtained results are gained by iterative

applying an initial guess of the unknowns coefficients and re-evaluating the guessed coefficients based on the coherence between the simulated and calculated points.

Mismatch in the comparator layout is recognized to be extremely critical concerning decision time mismatches between the left side and right side of the comparator's effective offset voltage. Results shown in Figures 57 to 59 were obtained by repeated comparator layout attempts with a emphasis of reducing mismatches between the left side and right side of the offset voltage.

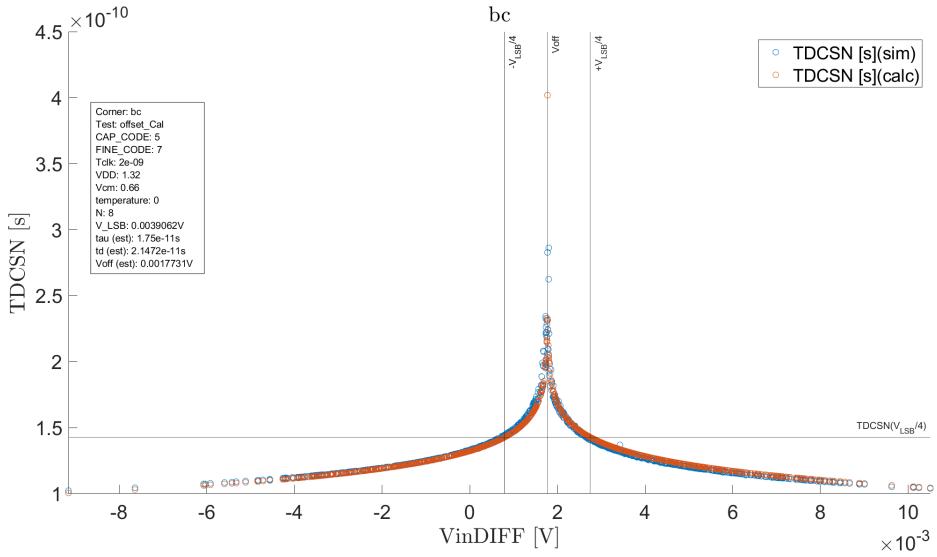


Figure 57: Comparator Decision Time Measured at the Skewed Output Inverters, X_P & X_N

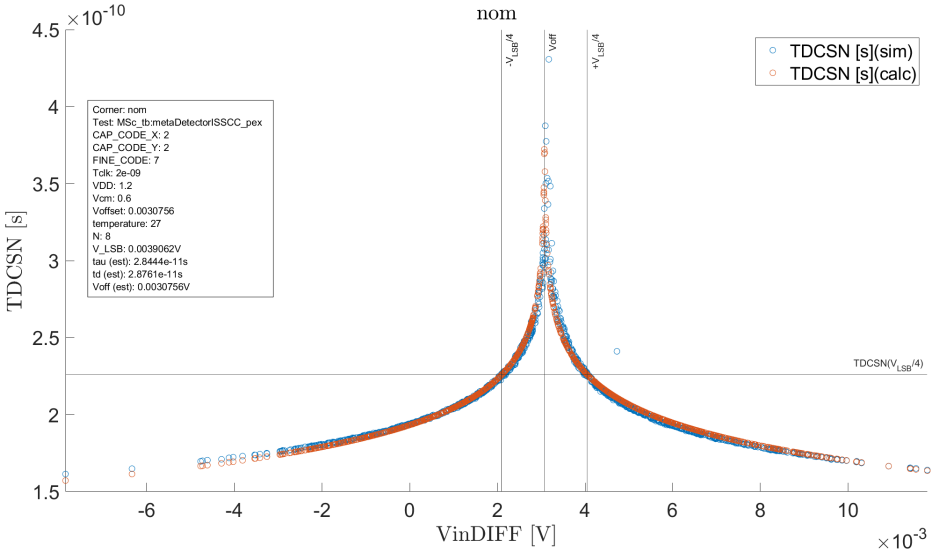


Figure 58: Comparator Decision Time Measured at the Skewed Output Inverters, X_P & X_N

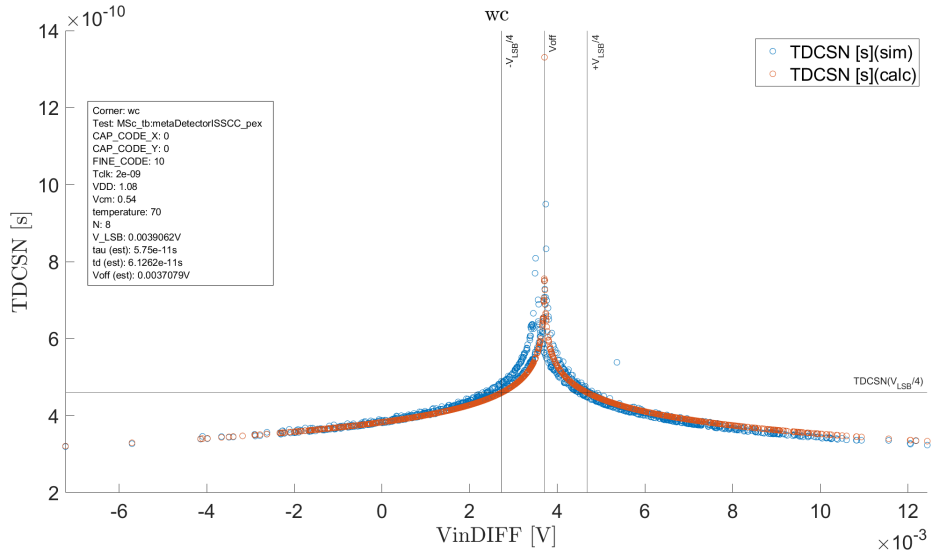


Figure 59: Comparator Decision Time Measured at the Skewed Output Inverters, X_P & X_N

Time Constraint Considerations

The required time span to detect input signal ranging within ± ¼ LSB is calculated utilizing the estimated values shown in Table 8 and inserting for the estimated unknowns (τ, V_{offset}, and t_d) into Equation 8 and solve concerning ¼ LSB. The time constraint circuit must satisfy a certain time resolution between delay codes. It is a part of the implementation causing linearity degradation (DNL and INL) in a SAR ADC as it adds time uncertainty to the LSB determination. An error source margin of $LSB/8$ is acceptable for each sub contribution of supplied error source. With this error margin in mind, the time constraint resolution is calculated by inserting the acceptable error source around the ± ¼ LSB boundary, and mapping it to the time variation through the exponential time to voltage relationship in Equation 8 denoted as Δt. The frames of the time constraint circuit is then established. It must cover a span and resolution across PVT variations as viewed in Table 8. The finest time resolution step (Δt) needed is based on the acceptable an error contribution of $V_{LSB}/8$.

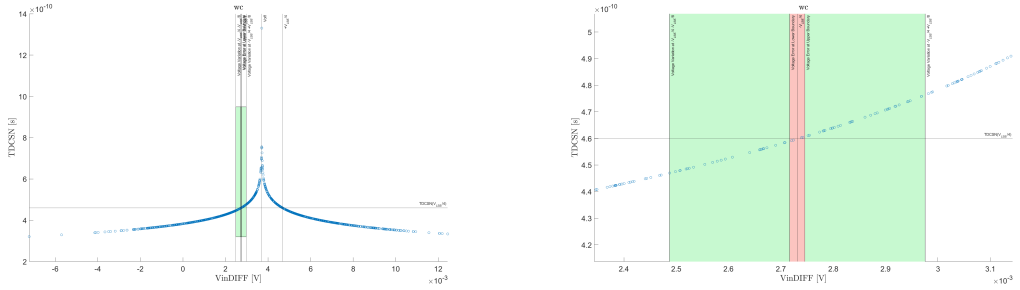
τ	V_{offset}	t_d	Δt	$T_{DCSN}(V_{LSB}/4)$	**Corner**
20.67 ps	2.55 mV	20.19 ps	8.38 ps	163.44 ps	bc
31.06 ps	3.45 mV	30.662 ps	12.59 ps	125.95 ps	nom
58.03 ps	3.77 mV	58.380 ps	23.53 ps	460.64 ps	wc

Table 8: Estimating Results

5.1.3 Time Constraint Delay Circuit

The time delay circuit is constructed utilizing coarse time segmentation and fine trimming within the coarse segments. Time constraint span is simulated by applying the lowest and tallest delay code, and measuring the range across corners. Simulation verifies the time constraint span is covered across corners. Time resolution between two delay codes is extracted by finding the neighbor delay code causing the largest time step. Phase noise is simulated by utilizing the analysis type *pnoise and pss* in Cadence® and employing Equation 9 were RMS jitter is calculated by adopting Equation 10. The phase noise extraction method presented in [7] is used in this analysis were the integrated phase noise power is integrated over the frequency range of interest ranging from 1 Hz up to 0.25 GHz.

Phase-Noise/Jitter Evaluation



(a) Large View (b) Enlarged View

Figure 60: View the Error Around the Boundary -¼ LSB Caused by Phase Noise in the Time Constraint

Results in Table 9 outline the simulation results for the time constraint delay circuit respectively the delay line. Results were gathered by utilizing circuit simulation tools as specified in the aforementioned paragraph. The introduced nonidealities on the LSB bit determination caused by phase noise in the delay line are back annotated by employing the method outlined in paragraph 2.8.3. Figures 60a and 60b views the plotted characteristics of the comparator's exponential time to voltage behavior with cursors at the lower middle-level threshold, and the outermost voltage error variation caused by phase noise. The green area depicts the acceptable region of introduced voltage error related to LSB/8. The red region displays the location of voltage error caused by encountering one standard deviation of simulated jitter in the delay line. The results lay well within the green region and by considering one sigma RMS jitter yields the phase noise to be at an acceptable level. Impact of phase noise is only taken into account at -¼ LSB decision threshold because it is similar at the decision threshold at +¼ LSB due to symmetry around the comparator's effective offset voltage.

Resolution	Error Contribution in Terms of LSB	RMS Jitter	Corner
6.5 ps	0.00702	290 fs	bc
11.4 ps	0.00499	310 fs	nom
22.6 ps	0.00738	857 fs	wc

Table 9: Time Constraint Circuit Simulation Results

5.1.4 Simulation & Validation of the Time Constrained Comparator Decision Time

Figure 61 depicts the simulation setup for analyzing the unresolved decision detector. The URD outputs three signals, were the *MD* is the middle-level detection ranging between $\pm$ ¼ LSB, and X_{POUT} and X_{NOUT} are the decision level ranging outside the middle-level boundary. A digital controllable time constraint circuit timeouts the comparator decision when the input lies within the $\pm$ ¼ LSB detection region and realizing a tri-level detection. The middle-level detection is simulated by applying randomly normal distributed input values with $\sigma = \pm 2 - 3\ V_{LSB}$ around the comparator effectively differential input voltage and scatter the unresolved decision detector middle-level flag as a function of the corresponding input signal.

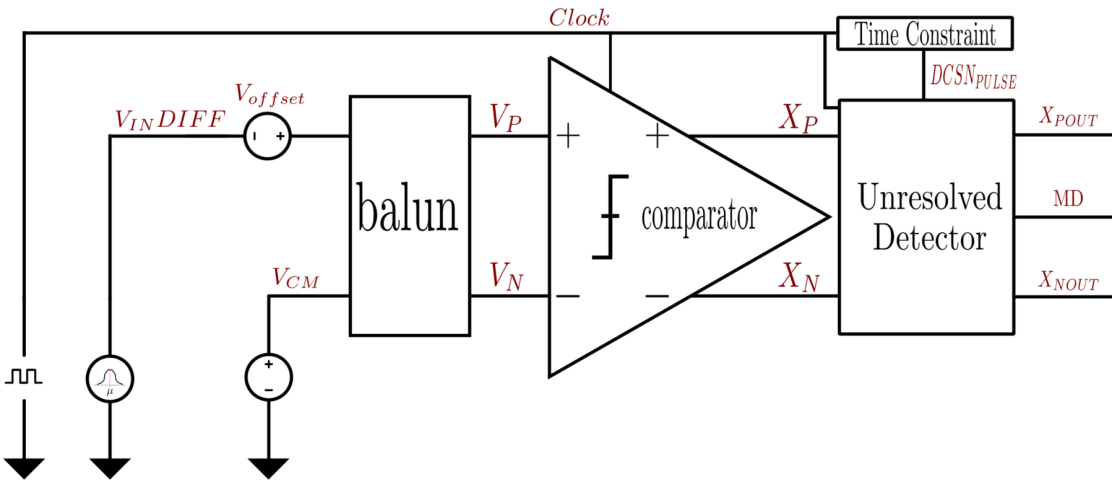


Figure 61: Simulation Testbench of the Unresolved Decision Detector

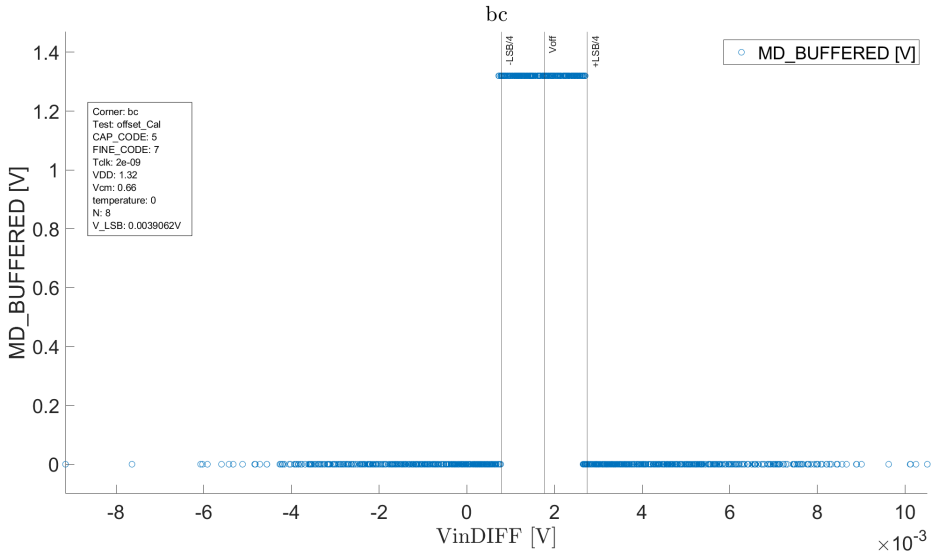


Figure 62: Unresolved Detector Constraining the Comparator Decision Time at ± ¼ V_{LSB}

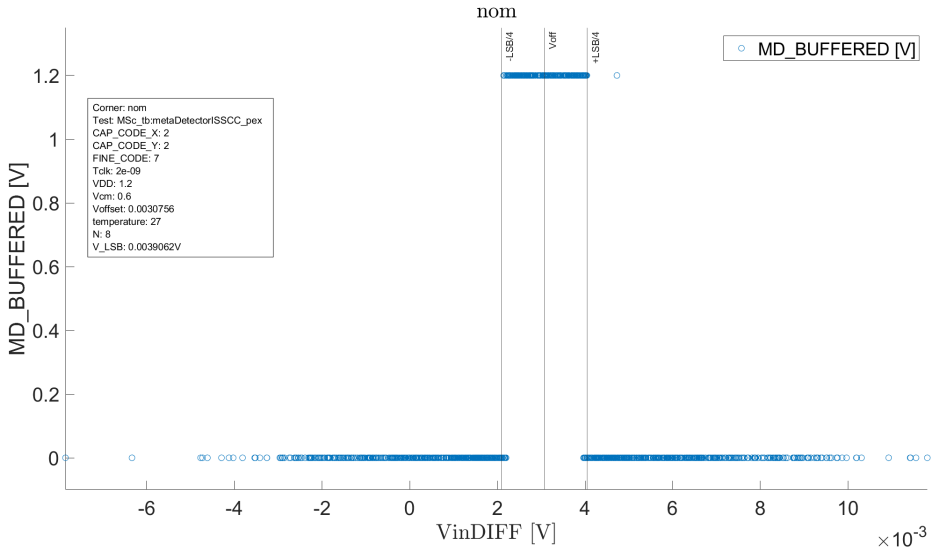


Figure 63: Unresolved Detector Constraining the Comparator Decision Time at ± ¼ V_{LSB}

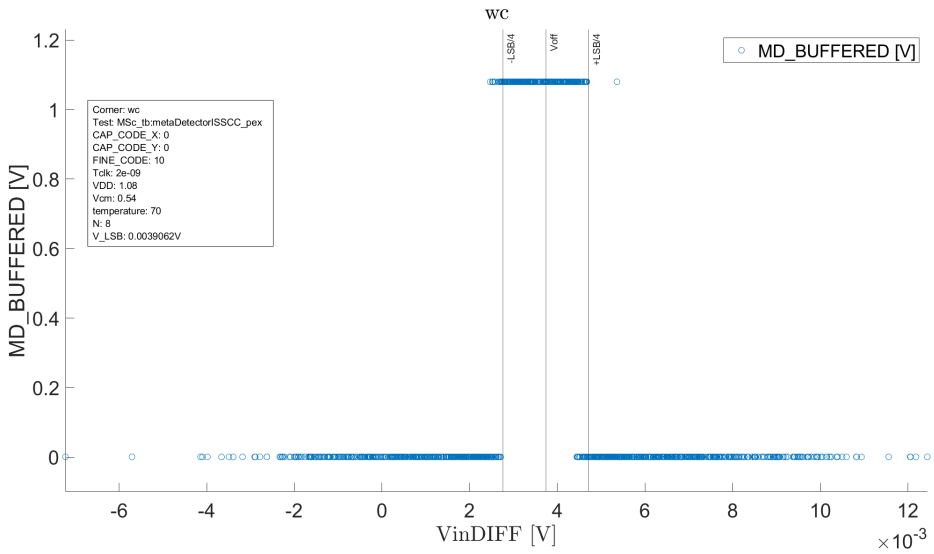
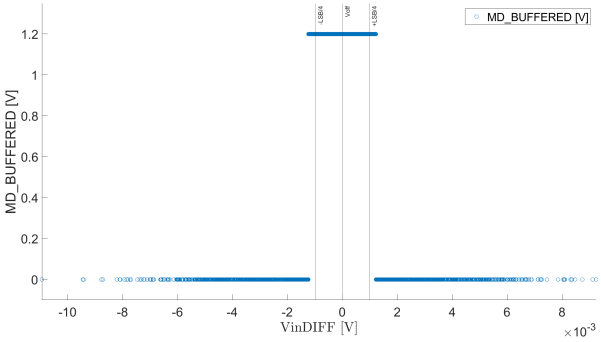


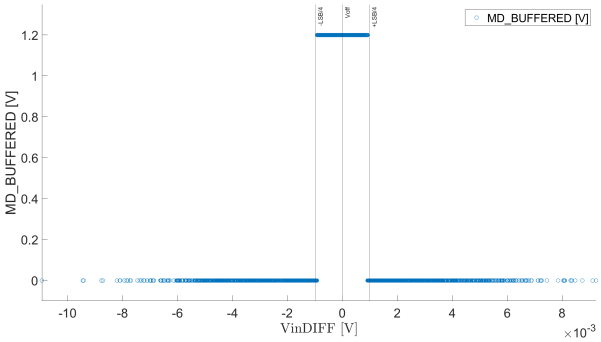
Figure 64: Unresolved Detector Constraining the Comparator Decision Time at $\pm$ ¼ V_{LSB}

Trimming of The Time Constraint

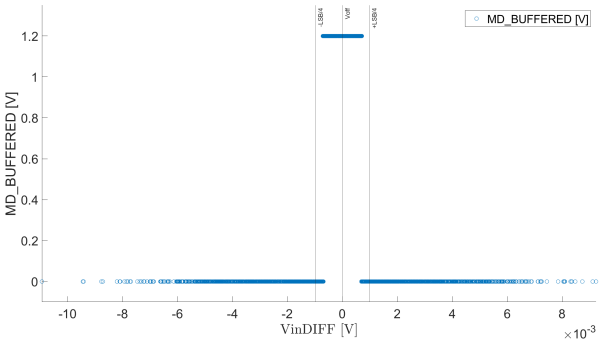
Figures 65a to 65c displays simulation results were the time constraint is adjusted between adjacent delay codes. The middle-level detection region is controlled by adjusting the time constraint.



(a) Timeout Rate Set too Short



(b) Timeout Correspond to the Middle-Level Decision Region



(c) Timeout Rate Set too Long

Figure 65: Schematic Simulation: Trimming of the Time Constraint Between Adjacent Delay Codes

5.2 Simulation & Validation of the SAR ADC

Table 10 encounter the simulation setup utilized for the SAR ADC simulation. The SAR ADC operates in a serial fashion and resolves a bit at each successive approximation step. The converter's sampling rate is limited by the number of successive approximation step, the allocated time used for sampling the input signal ahead of the conversion, and the amount of time to assert EOC before the next conversion begins. The converter's sampling rate is limited by the (N+1) successive approximation step with the implemented PVT trimmable tri-level comparator. Additional, a clock period for sampling the input signal and asserting EOC. This forms the (N+3) factor used for calculating the converter's sampling rate as viewed in Table 10.

Variables	Description
T_{CLK}	$1/0.5\,\text{GHz}$
Number of Bits (N)	8
Sampling Rate	$1/(T_{CLK}\cdot(\text{N+3})) = 45.45\,\text{MSa/s}$
V_{DD} & Temperature	Follows the Specified Values for the PVT Corners in Table 7
Number of Simulation Points (Periods)	1-5

Table 10: Simulation Variables

5.2.1 ADC Performance Evaluation

Figure 66 depicts the testbench employed for evaluating performance metrics for the designed ADC. The implemented SAR ADC has a dynamic full scale input range equal to 1V. The internal circuit is supplied by V_{DD} from an external reference. A fundamental sinusoidal signal with 1 Vpp is applied at the input of the ADC with a frequency corresponding to an integer of the sampling frequency. The configuration register set the time constraint of the LSB bit determination. The digital output from the SAR ADC is fed to an ideally DAC which converts the digital word into voltage. Then the DAC's output voltage is sampled in Cadence and extracted to further performance analysis in Matlab.

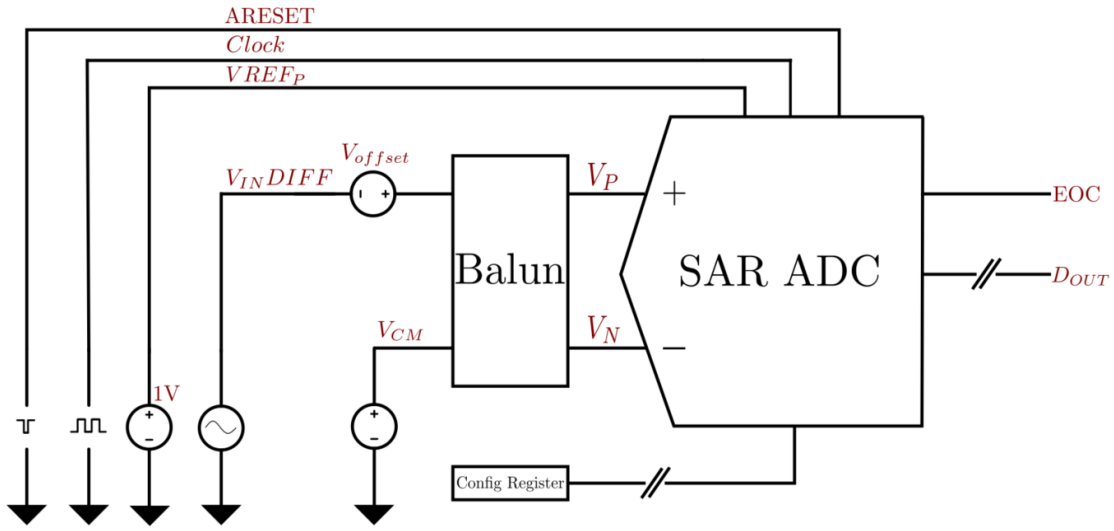


Figure 66: Test-bench of the SAR ADC Simulation

Figure 67 displays simulation results at schematic level of an ADC conversion. In the presence of a SOC, the DAC begins immediately to continuously track and hold the differential input signals in the DAC until a falling edge of SOC. The nodes VDACP and VDACN track the input signal until the SOC signal disappears, resulting in a proportion of the input signal being stored at the DAC's differential outputs. The voltage errors between VDACP and VDACN approaches zero as the comparator search binary for the closest quantized representation of the sampled input voltage. The converter's digital output code is read out from the memory cell at an end of conversion.

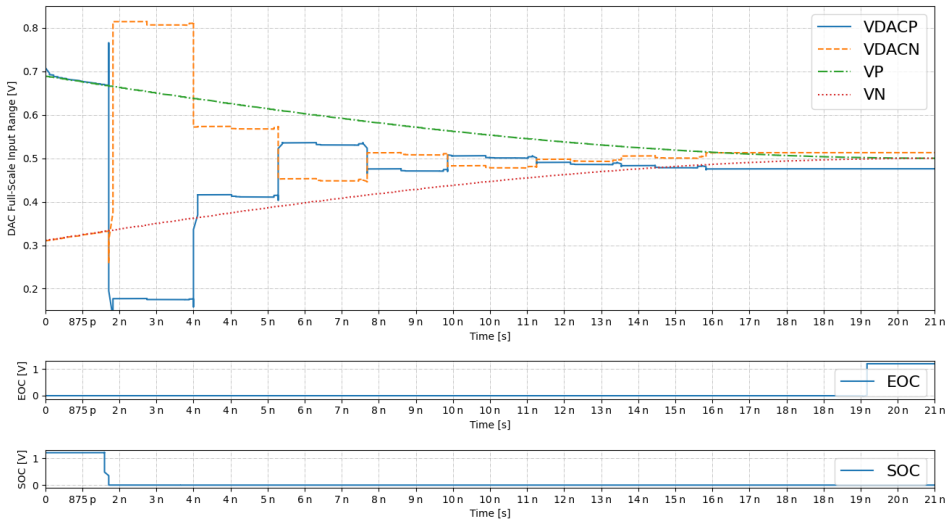


Figure 67: SAR ADC Waveform

5.2.2 Spectral Performance Evaluation

The SAR ADC is simulated with a number of conversions equal to the power of N to make it feasible to employ the Fast Fourier Transform (FFT) algorithm, were N is an integer. It were observed that post layout extraction model simulation requires an unreasonable number of days to drive a small number of conversions, therefore the spectral performance analysis of the ADC is more or less only simulated at a schematic level. Even running only central components of the ADC at post layout extraction level and leaving all digital circuits at schematic level turns out to be too time consuming. However, it is achievable to run the PTV trimmable tri-level comparator at post layout extraction and leave the other parts of the design at schematic level to overcome a reasonable long simulation time.

The designed ADC is evaluated by calculating its spectral performance metrics from simulation results. An ADC's signal to noise (SNR) ratio is fundamentally limited by quantization noise. Quantization noise is actually not physical noise, but it is a result of segmentation of the ADC's dynamic full scale input range into quantization levels resulting in a rounding of error appearing as noise. Real ADCs have electrical noise on top of the quantization noise, resulting in a deteriorated SNR. SNR is a measure of the input signal relative to the power of the noise when the harmonics are masked out and without the DC component.

Spurious free dynamic range (SFDR) is a parameter to measure the distance between the input signal and whatever other tones appear in the spectrum. Ideally, it appears

to be harmonics tones, but it could e.g. be tones coupled in from clock signals running at other places in the circuit. It is specified relative to the carrier [dBc] or relative to the ideally full scale input range in [dBFS].

Total harmonic distortion (THD) is a metric were the input signal is only viewed related to harmonics. The V_1 is the fundamental signal and $V_2, V_3, ..., V_N$ are the harmonics.

$$THD = 20 \cdot log(\sqrt{(V_2^2 + V_3^2 + ... + V_n^2)}/V_1) \tag{23}$$

Signal to noise and distortion (SINAD) is a measure of everything appearing in the spectrum. Technically, it is signal to noise and distortion, but it is taken to the input signal versus everything else appearing in the spectrum. It is calculated by integrating the power of all other tones appearing in the spectrum and back relate it to the input signal power, resulting in the SINAD.

$$SINAD = -10 \cdot log(10^{-SNR/10} + 10^{THD/10}) \tag{24}$$

Theoretically converter's SNR is limited by the number of bits from the following relationship, SNR = 6.02· N + 1.76. In practice, electrical noise deterioration the actual SNR to an ADC and reduce the effective number of bits (ENOB). ENOB is calculated by encountering signal to noise and distortion. It is signal versus everything else appearing in the spectrum. The ENOB metric is referred to what it actually had been in a system only limited by quantization noise.

$$ENOB = \frac{SINAD - 1.76 \text{ dB}}{6.02 \text{ dB}} \tag{25}$$

The figure of merit (FoM) is used to compare similar data converters with different performance specifications, were there exist different types of FoM. The Walden FoM is utilized in this work as it is preferred for data converters with SNDR > 50dB [12] as is this case. FoMs state the expected power consumption for a certain number of bits, and Nyquist frequency. The Nyquist frequency is half of the sampling frequency.

$$FoM = \frac{Power}{2^{ENOB} \cdot f_N} \tag{26}$$

Figures 68 and 69 displays the spectrum of the implemented ADC with a sinusoidal input signal. Harmonics from the fundamental frequency are suppressed by the quantization noise. The code employed for the spectral performance evaluation is presented in Appendix F. Results show a small change in the metrics from the core schematic simulation and were the PVT trimmable tri-level comparator is simulated with PEX contributions.

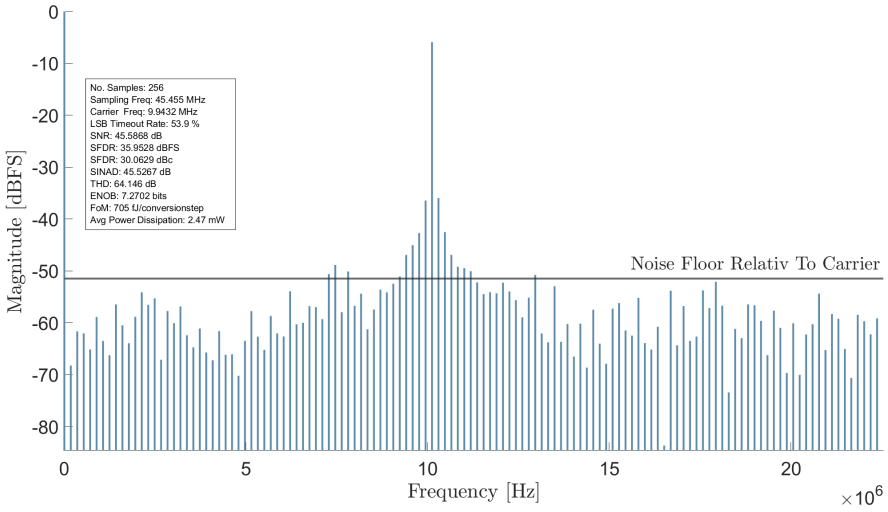


Figure 68: Spectral Performance Simulation Schematic Level

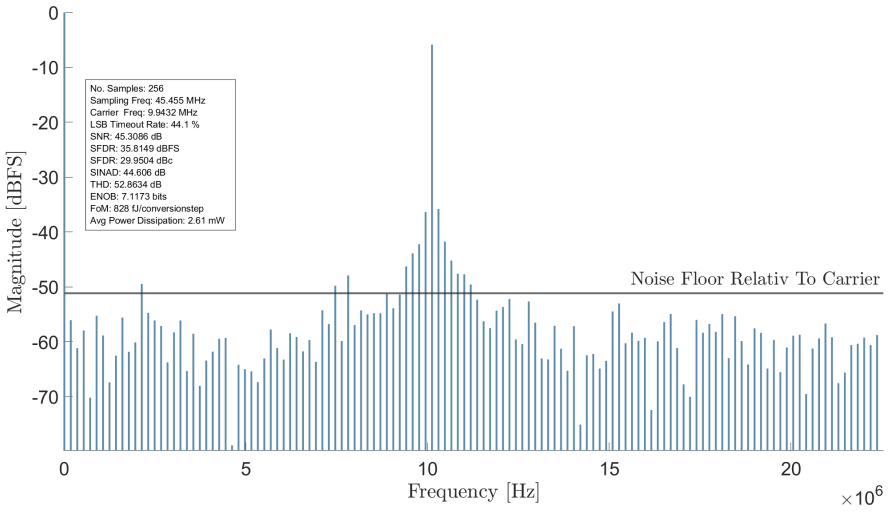


Figure 69: Spectral Performance Simulation of the Tri-Level Comparator at PEX and the other parts at Schematic Level

5.3 Planned Chip Validation

In this paragraph, planned verification strategies for the tape-outed circuit are presented. The PVT trimmable tri-level comparator and the SAR ADC are presented. The designed Printed Circuit Board (PCB) is physically verified by applying external input sources and then verifying the function is as intended. Table 11 list the equipment that was intended to be used for the verification of the designed integrated circuit.

Equipment	Description
Microcontroller	Nordic Semiconductor nRF52832 DK. Micro USB Cable to Computer FreeRTOS Kernel V10.0.0.
Debugger/Flash tool	Segger Embedded Studio 5.42a.
Oscilloscope	Agilent Technologies DSO6034A. Sample Rate(max): 2 GSa/s. Agilent: 10073C Voltage Probe.
Precision Source/Measure Unit (SMU)	Keysight B2901BL Source and Measure Both Voltage and Current. Source and Measurement Resolution Down to 10 fA and 100 nV.
Python	Analysis of the measurements. Source and Measure Both Voltage and Current. SCPI commands to write and read, to Oscilloscope and SMU. Version: 3.9.13.

Table 11: IC Testing Equipment

5.3.1 Printed Circuit Board Verification

The Printed Circuit Board (PCB) is constructed to interact with the designed integrated circuit and surroundings. It is necessary to utilize level shifters for the digital inputs and outputs to interact between two different power domains as the IC's internal voltage reference supply is at a different domain level than the test equipment. The employed voltage level translator is the TXB0108PW ordered from Texas Instrument. Figure 70 displays a view of the printed circuit board. The connectors J2 are the BNC connectors and P1 is 15 pin-header with inputs and outputs to the chip. The designed integrated circuit contains a PVT tri-level comparator and the SAR ADC. The PCB is drawn with separated power nets such that independent current measurement is feasible to attain. Decoupling capacitors are added nearby the power supply pads for the surface mounted devices (SMDs).

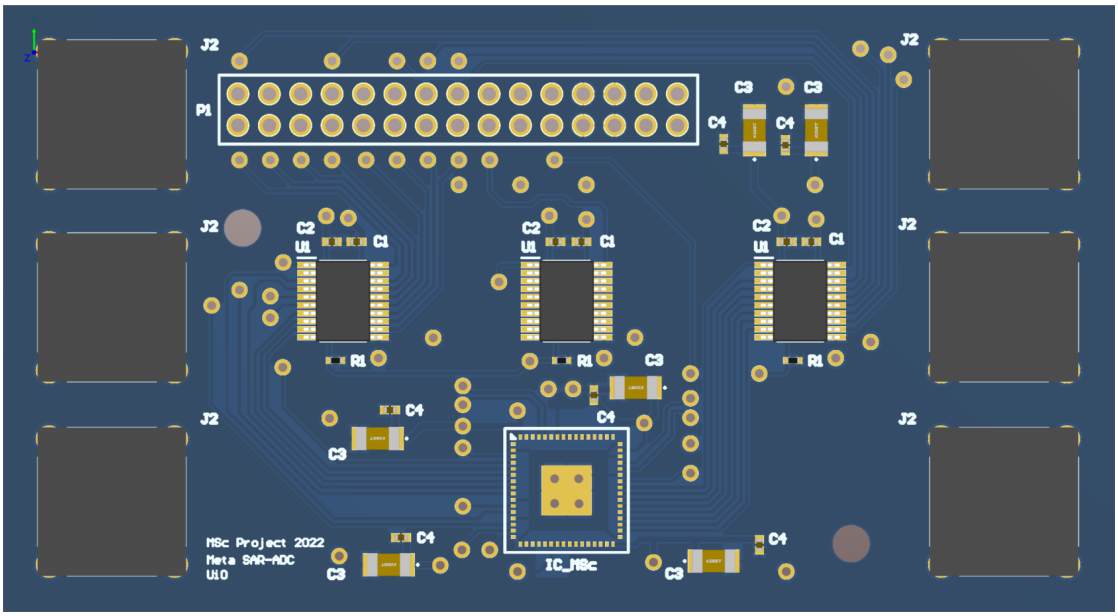


Figure 70: PCB for the Designed IC

The comparator's time constraint is configured through a shift register. The IC's internal shift register is reset ahead of being configured. At the falling active clock edge, the bitstream to be written to the chip is left shifted one position, and were the IC's internal shift register samples the input at the rising clock edge. Figure 71 views the measured clock, bitstream, and reset signal output from the employed microcontroller. Figure 72 displays the measured signals at the output of the employed voltage level shifter with another bitstream configuration. The LSB in the configuration register is connected to an external pad with a buffer between such it is possible to verify the shift register operates as intended.

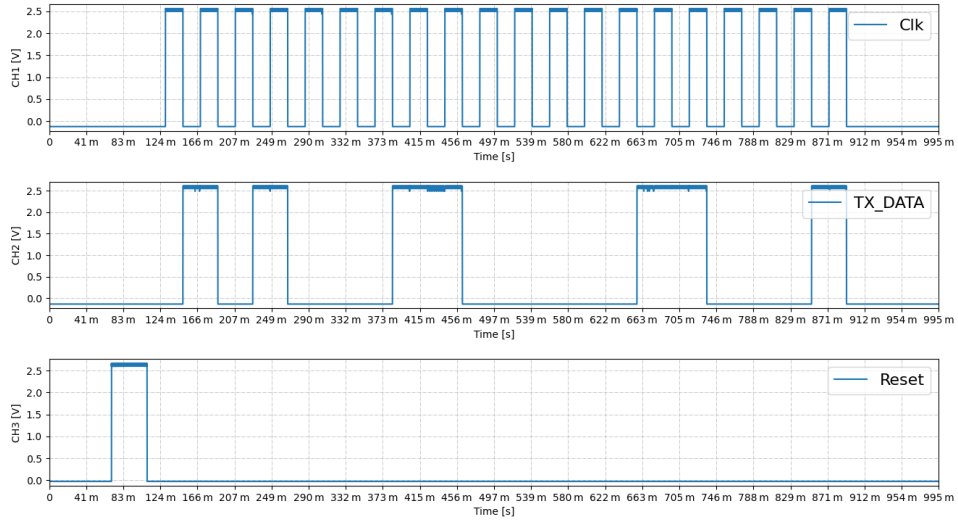


Figure 71: Shift Register Waveforms Generated From the nrf52DK

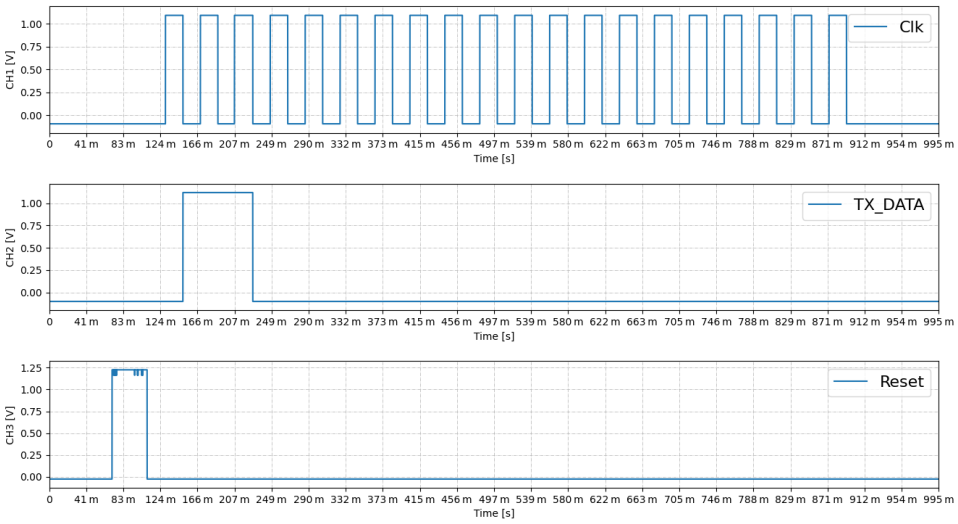


Figure 72: Shift Register Waveforms Though the Voltage Level Shifter

5.3.2 PVT trimmable Tri-level Comparator

The implemented PVT trimmable tri-level comparator is designed with an unresolved decision detector attached to the comparator's outputs. The intention of the test procedure for this design is to verify the middle-level detection range is suitable to be realized in silicon. Therefore, only one unresolved decision detector is applied at the output of the comparator as shown in the PVT trimmable tri-level comparator implementation outline in Figure 34. As mentioned, the detector resamples the comparator decision which adds a latency before the middle-level flag is available. A method of verifying the middle-level detection is to apply an input voltage to the comparator and reevaluate the middle-level flag after three clock periods. Then reset the latch and apply a new input value. This test procedure is not very dynamic. A different test procedure is to resample the latch's output at the falling active clock edge and reset the latch a certain time ahead of the next comparator active clock edge. Then it is possible to continuously run the PVT trimmable tri-level comparator and sample the detector's middle-level flag continuously by a microcontroller. Figure 73 shows the generated signal from the microcontroller were the latch is reset a certain time before the comparator active clock edge. Figure 74 views an enlarged view of the generated signal in continuously executing mode. The last mentioned test procedure is the same employed for simulation.

Signals Generated in Continuously Running Mode

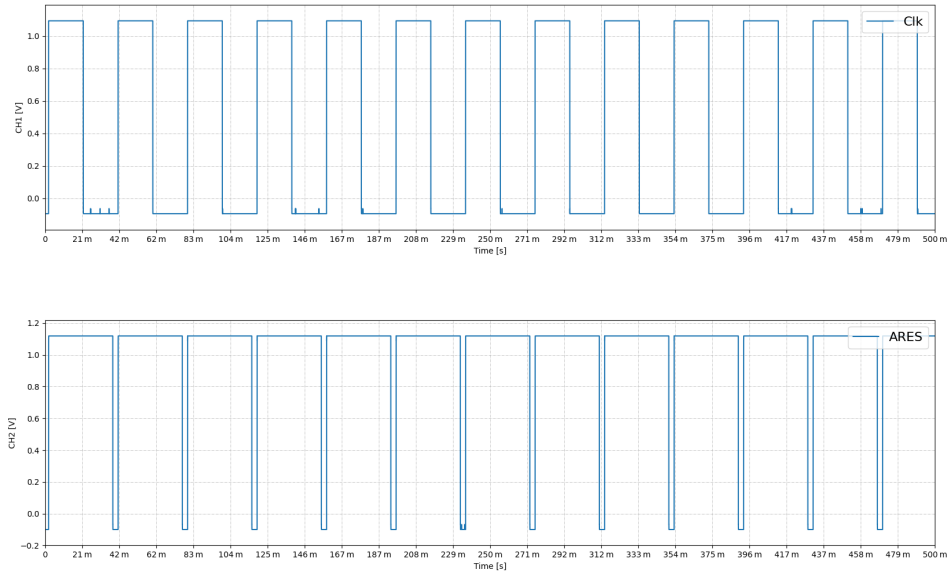


Figure 73: Clock Signal and Active Low Reset

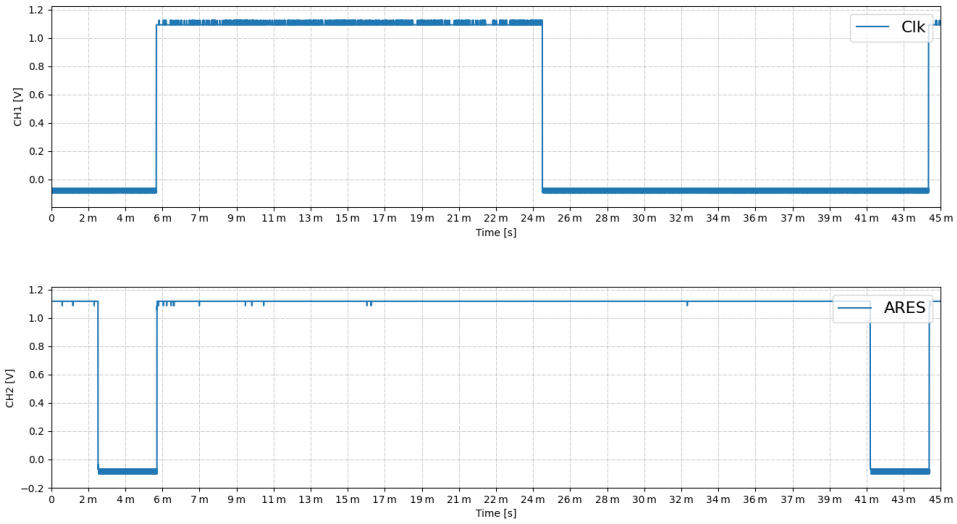


Figure 74: Enlarged View of Clock Signal and Active Low Reset

5.3.3 Tri-level PVT trimmable Comparator Employed in a Fully Differential SAR ADC

The test method for the implemented SAR ADC has similarities to the PVT trimmable tri-level comparator. It is designed with an equivalent configuration register and the chip configuration method is intended to be reused. Apart from that, the test purpose of the SAR ADC is different. It is intended to apply a fundamental sinusoidal input signal with 1 Vpp to the ADC, then configure the ADC to enter continuous sampling mode, and consequently shift the conversions code into internal chip memory. The microcontroller is intended to count the number of EOC by the ADC and shift conversion codes from the chip to the microcontroller after (512+1) EOC flags are registered. Further, shift the samples to a computer and perform spectral performance evaluation.

6 Discussion

The preceding sections present the design approach for the two integrated circuit designs, emphasizing design methodology, implementation, and performance evaluation. A theoretical analysis of metastability is first investigated and viewed in context of high speed comparator related to the exponential time to voltage characteristic. The applied time constraint circuit's range and resolution are found from the mathematical decision time Equation 8. Furthermore, the SAR ADC functional principle is studied and implemented utilizing the adopted PVT trimmable tri-level comparator. Simulation results prove the viability of detecting a middle-level region by constraining the comparator's decision time. An additional bit of resolution in the converter's output is achievable by employing the investigated timeout scheme by constraining the comparator's decision time corresponding to $\pm$ ¼ LSB. As a consequence, the SAR cycle is guaranteed to complete. The obtained results do not fulfill the requirement of the converter's sampling rate and possibilities for improvements exist. Implementation of the SAR ADC is verified against spectral performance metrics and were the tri-level PVT trimmable comparator is verified by achieving the specified middle-level decision region with acceptable error margins across PVT variations. Besides that, design mistakes and possible improvements have been recognized in retrospect which is grasped in this chapter.

The central purpose of this work was to investigate the feasibility of imposing a constraint on the comparator decision time and guaranteed the SAR cycle to complete. Further, gain an additional bit of resolution in the converter's output by introducing a timeout scheme on the decision time chosen to correspond to ± ¼ LSB. Furthermore, introducing an unresolved decision detector translates to designing the comparator with a less critical time constant as the comparator is time constrained when solving input voltages ranging within the middle-level region. A long MTBF in a conventional comparator realization is achievable by designing it with a small enough time constant. Instead, a PVT trimmable tri-level comparator actively uses a potentially metastable comparison to realize an additional bit of resolution and still gain a long MTBF rate.

Simulation results prove a PVT trimmable tri-level comparator is feasible to be realized by constraining the available decision time. Figures 62 to 64 displays that a PVT trimmable tri-level comparator is realized. Additionally, Figures 65a to 65c schemes the feasibility of trimming the comparator decision time with the emphasis on timeout the comparison for input voltages ranging within the middle-level detection region.

An important part of this project was to quantify the overhead required to introducing a timeout scheme on the comparator decision time and actually designing the comparator with a small enough time constant to achieve a long MTBF. The theoretical MTBF analysis brought to forth the effect of employing resampling flip flops to increase their resolve time and increase the overall MTBF for the detector. It is essential that the overhead of introducing a time constraint on the comparator decision time does not intro-

duce a shorter MTBF rate compared to actually designing the comparator with a small enough time constant. The yield of introducing a timeout scheme on the comparator's decision time would then had been disfavored supposing it resulted in smaller MTBF compared to actually designing the comparator with a sufficiently small enough time constant.

The MTBF analysis is a theoretical evaluation of the MTBF contribution of each component within the detector which has critical timing combinations at its input. Error sources can not be totally eliminated in the quantified overhead compared to actually designing a comparator with a sufficiently small time constant. The outcome of utilizing the resampling technique is to make the detector's own metastability rate sufficient low. The presented analysis was intended to be used to state the yield of introducing a timeout scheme on the comparator decision time compared to the required time constant to satisfy the same MTBF as with the unresolved decision detector. However, it has been difficult to get a valid estimate of the detector MTBF.

6.1 PVT trimmable Tri-Level Comparator Architecture Selection

Selecting an unresolved decision detector architecture is the core part of this task. Selecting an inappropriate detector architecture could cause performance degradation for the middle-level detection. A detector is selected by accomplishing literature regarding metastability when interacting between asynchronous input and synchronous circuit. The achieved results show that the selected unresolved decision detector has been a great choice concerning reducing the detector's own metastability rate, although other design architectures may exist.

6.2 Practicality

Simulation results for the PVT trimmable tri-level comparator shows it is feasible to realize and gain an additional bit of resolution in the converter's output. It is realized by employing a trimmable time constraint and the ability to constrain the comparator's available decision to the middle-level detection region.

Timeout chosen to correspond to ± ¼ LSB effectively adds an additional bit of resolution to the converter's output. Consequently, the DAC size is diminished by factor one, resulting in a reduced number capacitors in the DAC of a factor two, shorter settling time, and lower power dissipation. It is required an additional successive approximation cycle as a consequence of the utilized resampling technique employed in the detector. The additional SAR cycle reduce the achievable speed of the converter and its a downside.

On another side, the key benefit of the investigated timeout scheme is to achieve a long MTBF with the tri-level comparator and simultaneously allowing to design the comparator with a modest time constant. Additionally designing the comparator with a

less critical time constant directly reduce the comparator's power consumption.

Assessment of the overhead concerning introducing the timeout scheme is a comprehensive case to evaluate. The compromise is design dependent, there may exist other applicable methods to realize the metastable detector and the time constraint circuit. Introducing additional complexity results in a budget compromise, one gain and lose something. The yield of introducing additional complexity is viable if the outcome of contribution becomes positive. The investigated timeout scheme is assumed to be viable to realize by not encountering the required PVT tracking circuit. A PVT tracking circuit that continuously tracks and adjusts the time constraint such that the timeout rate occurs correctly is not encountered in this work. By assuming the complexity introduced by the tracking circuits is feasible to overcome, it is expected that the timeout scheme is viable to integrate considering the complexity it involves and its benefits.

6.3 Number of ADC-slices

There is a requirement for a certain sampling rate for the SAR ADC. The time interleaved principle of running multiple ADC slices in parallel is intended to be used. Even by running ADCs in parallel, it is expected to be inapplicable to achieve the required sampling rate with the implemented ADC. The SAR ADC intrinsic conversions speed should be improved to make it practical to satisfy the specified sampling rate.

The implemented design accommodates a core clock frequency of 0.5 GHz. As the reader may interpret, it is designed with too slow clock frequency to satisfy the required ADC sampling rate. To accommodate the specified sampling rate utilizing the presented time-interleaved principle it is required (2 GSa/s)/(45.4545 MSa/s) = 44 SAR ADC slices which is an unwanted number of slices to be realized. The required sampling rate is specified in Paragraph 1.3 and the achieved sample rate is specified in Table 10.

Although the comparator is time constrained when solving input ranging within the middle-level region it is fundamentally that it must at least be fast enough to solve input voltages ranges outside of the middle-level region across PVT variations. In other words, the comparator's high clock period must be longer than the time constraint utilized to timeout the comparator at the middle-level, else the comparator would be incapable to solve fast enough for input voltages outside the middle-level detection region. This behavior has been observed at the *wc* corner, especially in the exhibition of parasitic capacitance from the trimmable capacitor bank even if it is coded zero digitally. Therefore, the core clock frequency has been reduced after a digital controllable capacitor bank was employed, to accommodate a comparator timeout that is feasible to detect across PVT variations. The used clock frequency is 0.5 MHz, to reduce the number of required ADC slices to a reasonable amount, it is expected that the system

clock frequency should be at least improved by a factor of four or even more.

Therefore, there is a compromise between the applicable amount of capacitive load added at the output of the comparator and its speed, respective the comparator's time constant. This is especially an issue at the *wc* corner as it is intrinsic slower.

6.4 Existing Unresolved Topics

As mentioned in the initial chapter of this thesis, there is a risk of not being able to perform chip measurements due to global semiconductor chip shortages. It has been impossible to perform chip measurements before the final submission as the chip is stuck at the manufacturer in the bounding and packing process. There was set aside approximately one month for testing, but as the chip has not arrived is the planned measurements omitted from the report. It is disheartening to view the amount of effort used to actually realize the design in silicon in retrospect when chip measurements have been impossible to perform. However, it is not possible to predict such event and it must just be accepted. If the chip is received a reasonable time before of the defense, an effort will be put into getting measurements done and included in the final presentation.

6.5 Results

Repeated simulations are needed when moving satisfying design at schematic level and toward a post layout extraction model, both at a cell view and a top level view. Different testbenches have been constructed for each application area, were the test benches are illustrated ahead of the presented simulation results. The subsequent paragraph enters a view of potential design mistakes and improvements that have been recognized in retrospect of the design process.

6.5.1 Comparator Decision Time Extraction Mismatches

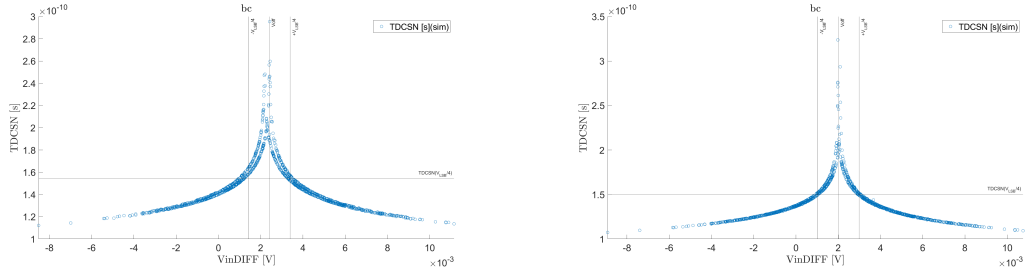
Figures 57 to 59 displays the results from both the simulated and calculated points. The calculated points are quantified by inserting the estimated unknowns of Equation 8 and the simulated input voltages, extracting its corresponding decision time. The characteristics of the simulated points and the estimated points should ideally correlate perfectly, but deviations can be seen. The mismatch between the right side and left side of offset voltage in the simulated points at the *wc* corner is mainly caused by an asymmetrical layout. Simulated decision time points at *wc* corner in Figure 59 has spread around the determination of the middle-level region. The simulated points appears in two exponential slopes within the middle-level detection region.

The spread around the determination of the middle-level region from simulated points is expected to arise from hysteresis caused by the comparator to not be completely reset

ahead of the next comparison. The reset time of the comparator seems to be largely influenced by the amount of capacitance at the comparator outputs. Reducing the amount of capacitive load results in the comparator being more properly reset and reduces hysteresis. Due to the amount of parasitic capacitance exhibited at the *wc* corner being too large, the comparator does not reset sufficiently causing spreads in decision time within the middle-level detection region. The PVT trimmable tri-level comparator principle by constraining the decision time is still valid, but the results in *wc* corner are caused by an unobservant design choice employed in the design of the capacitor bank utilized to trim the comparator time constant.

6.5.2 Capacitive Load at the Comparator's Differential Outputs

The effect of adding capacitive load at the comparator outputs has been encountered in paragraph 4.5, it influences the comparator time constant which leads to a change in decision time. This effect is viewed in Figures 75a and 75b, were larger capacitive loads increase the comparator decision time. As assumed, the effect of spreads of simulated decision time points within the middle-level region is caused by adding too large amounts of capacitive load as Figures 75a and 75b schemes. The design of the capacitive load range is not corrected due to this behavior being recognized after the chip was delivered for production.



(a) Large Capacitive Load

(b) Reduced Capacitive Load

Figure 75: Characteristics of too Large Capacitive Output Load at the Output of the Comparator

6.5.3 Converter Speed

By viewing the converter speed limitation considering the comparator speed, it appears that its time constant set a fundamental limitation of its comparison time. As aforementioned, to improve the sampling rate for the designed SAR ADC it is necessary to make the comparator intrinsic more faster. The supervisors mentioned that the use of a faster transistor as the Low Threshold Voltage (LVT) devices could be a choice to improve the comparator's intrinsic speed. It was not correct due to a limited amount of time before the final tape-out deadline. Additionally, there has been observed a large change from

schematic to post layout extraction simulation in the time constant extraction. It is expected to be caused by a combination of the amount of added metal in the layout, and design choices as buffering.

The DAC's settling time limits the achievable converter speed. The DAC needs to settle sufficiently before the comparator actually performs the comparison, else the comparator may output an incorrect comparison if the DAC outputs are in a transition. Under the *wc* corner, it is observed that the switching of the two MSB capacitors transition does not settle fast enough ahead of the comparator performing its comparison. There is a need of redesigning the switches controlling at least the two MSB capacitors such that it settles fast enough.

6.6 Challenges

The essence of this thesis is to get an understanding of metastability in high speed comparators and its circumstances related to failure rate in high speed SAR ADCs. Naturally, knowledge within the field of research increases along with the amount of invested effort. It was difficult to understand investigated papers in depth in the beginning. A lot of detailed information is given. In the end, it's all about clearing out the fog and harvesting the essence.

My biggest challenge has been to progress from a theoretical concept to a circuit that is simulable. Concepts do often sound clear when it is presented, but it turns out to be more complex than expected to construct a reasonable simulation model. In the end, it's a part of the process of gaining new knowledge and being able to work independently.

7 Conclusion

In the preceding chapters, the viability of introducing a timeout scheme on the comparator decision time was presented along with theory, implementation, and finally integration into a SAR ADC. The emphasis of this chapter is to conclude upon the conducted work and the viability of introducing a timeout scheme, based on the theoretical background and simulation results. In the end, suggestions for future work are presented.

In this report, a timeout scheme of uncompleted comparison in the SAR ADC's binary searches for the closest quantized representation of the sampled input voltage is investigated. The approach of introducing a timeout scheme allows designing the comparator with a more modest time constant, resulting in lower power consumption for the comparator, and still an attainable specified MTBF. An unresolved decision detector constraints the available decision time the comparator has to output a well defined logic level. Furthermore, an additional bit of resolution in the converter's output is attainable by timeout the comparator when solving input voltages ranging within a middle-level detection region, respectively at ± ¼ LSB. Further, the SAR cycle can guaranteed to complete by imposing a time constraint on the comparator decision time corresponding to the middle-level detection region.

The obtained result is gained by running simulation model as depicted ahead of the presented results. As shown in the validation and simulation chapter, a PVT trimmable tri-level comparator is achievable by constraining the comparator decision time, and gaining an additional bit of resolution in the converter's output by constraining the decision time corresponding to the middle-level detection region. It is constructively possible to diminish the the number of capacitors in the DAC by a factor of two, resulting in reduced DAC settling time and power consumption.

As explained in the introduction, the viability of introducing a timeout scheme is to reduce the strict requirement of a fast comparator with a small enough time constant to achieve a long MTBF rate. Instead, this method allows designing the comparator with a more modest time constant and hence timeout the comparator comparison when it solves input voltages ranging within the middle-level region, actually the region were the decision time is at its largest. The unresolved detector introduces supplementary complexity to a conventional SAR ADC, as it imposes a constraint on the comparator decision time. Additional circuits such as time constraint, unresolved decision detector, and calibration circuit are required, were the last one mentioned is not directly a part of this thesis. This method imposes additional complexity, but concerning its advantages is expected to compromise its budget, related to simplifying the comparator design, reducing comparator power consumption, diminishing DAC size, and the most essential part, the SAR cycle can be guaranteed to be complete. However, the unresolved decision detector is also associated with a MTBF rate, as it also is inevitability against

metastability. The implemented detector employs the aforementioned resampling technique to achieve a long MTBF for the detector.

Chip measurements have been impossible to perform due to the integrated circuit has not arrived before the final submission of this thesis. A lot of effort has been invested in realizing a tape-out circuit, PCB, and test software. It is disheartening to not be able to physically test the chip after a large amount of work has been put into it. Additional, it is impossible to get a clear view of the practicality of introducing a timeout scheme in a SAR ADC, because inaccuracies in the simulation model and very time consuming simulations are required when running post layout extraction of the implemented SAR ADC.

In summary, the investigated approach of introducing a timeout scheme can be expected to be viable if the required calibration circuit does not introduce too much complexity. As design experience is gradually acquired during the thesis, one becomes more observant that improvements exist in the whole chain from a schematic view up to a final tape-outed circuit.

7.1 Future Work

The unwanted decision time spread behavior as discussed should be viewed in perspective of an alternative implementation of the capacitive bank. Additionally, the comparator layout should be even more optimized considering mismatches, causing different decision times for a positive and negative input value, which introduces error in the middle-level detection as is especially present in figure 64.

This work does not encounter a calibration circuit for performing foreground and background calibration of the time constraint. An interesting part would be to investigate a method of realizing the calibration circuits with the emphasis to achieve 50 % middle-level detection with a uniform distributed input signal. Knowledge of the additional complexity it introduces will give a better estimate of the viability of introducing a timeout scheme and gain an additional bit of resolution and guarantee the SAR cycle to complete.

References

[1] Bilal I Abdulrazzaq et al. "A review on high-resolution CMOS delay lines: towards sub-picosecond jitter performance." In: *SpringerPlus* 5.1 (2016), pp. 1–32.

[2] S. Bashir et al. "Analog-to-digital converters: A comparative study and performance analysis." In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. 2016, pp. 999–1001. DOI: 10.1109/CCAA.2016.7813861.

[3] Stefano Brenna et al. "An efficient tool for the assisted design of SAR ADCs capacitive DACs." In: *Integration* 53 (2016), pp. 88–99.

[4] B.P. Ginsburg and A.P. Chandrakasan. "An energy-efficient charge recycling approach for a SAR converter with capacitive DAC." In: *2005 IEEE International Symposium on Circuits and Systems*. 2005, 184–187 Vol. 1. DOI: 10.1109/ISCAS.2005.1464555.

[5] Brian P. Ginsburg and Anantha P. Chandrakasan. "500-MS/s 5-bit ADC in 65-nm CMOS With Split Capacitor Array DAC." In: *IEEE Journal of Solid-State Circuits* 42.4 (2007), pp. 739–747. DOI: 10.1109/JSSC.2007.892169.

[6] John P. Keane et al. "16.5 An 8GS/s time-interleaved SAR ADC with unresolved decision detection achieving -58dBFS noise and 4GHz bandwidth in 28nm CMOS." In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 284–285. DOI: 10.1109/ISSCC.2017.7870372.

[7] Walt Kester. "Converting oscillator phase noise to time jitter." In: *Tutorial MT-008, Analog Devices* (2009), p. 2009.

[8] Walt Kester. *Mixed-Signal and digital signal processing ICS | Analog Devices*. June 2005. URL: https://www.analog.com/media/en/analog-dialogue/volume-39/number-2/articles/the-right-adc-architecture.pdf.

[9] Lindsay Kleeman and Antonio Cantoni. "Metastable Behavior in Digital Systems." In: *IEEE Design Test of Computers* 4.6 (1987), pp. 4–19. DOI: 10.1109/MDT.1987.295189.

[10] Zengjin Lin et al. "Modeling of capacitor array mismatch effect in embedded CMOS CR SAR ADC." In: *2005 6th International Conference on ASIC*. Vol. 2. 2005, pp. 982–986. DOI: 10.1109/ICASIC.2005.1611492.

[11] W. Evans E. Naviasky H. Tang B. Allison J. Matsuzaki. *Comparator metastability analysis - designer's guide*. 2021-30-12. [Online]. URL: https://designers-guide.org/analysis/metastability.pdf.

[12] Boris Murmann. *A/D Converter Figures of Merit and Performance Trends*. ISSCC Videos. 2017. URL: https://web.stanford.edu/~murmann/adcsurvey.html.

[13] Boris Murmann. *ADC Performance Survey 1997-2021*. URL: http://web.stanford.edu/~murmann/adcsurvey.html.

[14] Taimur Rabuske and Jorge Fernandes. *Charge-Sharing SAR ADCs for Low-Voltage Low-Power Applications*. Springer, 2017.

[15] Behzad Razavi. "The Bootstrapped Switch [A Circuit for All Seasons]." In: *IEEE Solid-State Circuits Magazine* 7 (June 2015), pp. 12–15. DOI: 10.1109/MSSC.2015.2449714.

[16] Behzad Razavi. "The StrongARM latch [a circuit for all seasons]." In: *IEEE Solid-State Circuits Magazine* 7.2 (2015), pp. 12–17.

[17] Akira Shikata et al. "A 0.5V 1.1MS/sec 6.3fJ/conversion-step SAR-ADC with tri-level comparator in 40nm CMOS." In: *2011 Symposium on VLSI Circuits - Digest of Technical Papers*. 2011, pp. 262–263.

[18] Chris Wellheuser. *'metastability performance of Clocked Fifos' - ti.com*. Accessed on 02.05.2022. 1996. URL: https://www.ti.com/lit/an/scza004a/scza004a.pdf.

Appendices

A Implementation of Estimation Problem Solver Utilizing the Matlab® utility fminsearch

```
1
2  clear
3  clc
4  close all
5
6  errfun_msq = @(x) sum( (TDCSN_SIM_raw.value -(-x(1)*log(abs(VIN_DIFF.value-x(2))) + x(3)) ).^2);
7
8  lsbFun = @(g) (g(1)/(2^(g(2)))); %LSB equation; VDD/(2^N)close all
9
10 N = 8; % number of bitsTHe
11 reportPlot = false;
12 plotEstCurve = true;
13 resPath = "C:\Users\Ole-Georg\Documents\UIO\MSc_PROJECT\042022\18042022\run1\";
14
15 corners = unique(readtable(resPath+"corners.csv", 'Delimiter', ','));
16 params = unique(readtable(resPath+"param.csv", 'Delimiter', ','));
17
18 % table to store results
19 len=size(corners,1);
20 z=zeros(cast(len, 'int16'),1);
21 str=strings(cast(len, 'int16'),1);
22 res_t = table(z,z,z,z,z,z,str, 'VariableNames', {'tau [s]' 'V_{off} [V]' ...
23 'T_{d} [s]' 'TDCSN(LSB/4) [s]' 'delta t [s]' 'lsb [V]' 'corner'});
24
25 % Type in the expected tau over PVT, bc, nom, wc
26 tau_LUT = table(tau_bc, tau_nom, tau_wc, 'VariableNames', {'bc' 'nom' 'wc'});
27
28 if(~reportPlot)
29
30 for k = 1 : length(corners.Corner(1:end))
31
32     VIN_DIFF_raw   = readtable(resPath + "\" + corners.Corner(k) + "\" + "VinDIFF.csv");
33     TDCSN_SIM_raw  = readtable(resPath + "\" + corners.Corner(k) + "\" + "TDCSN.csv");
34     CompOffset_raw = readtable(resPath + "\" + corners.Corner(k) + "\" + "CompOffset.csv");
35
36     % create axis description by combinding output name and unit
37     xAxis = char(unique(VIN_DIFF_raw.output(1:end)))+" ["+ char(unique(VIN_DIFF_raw.unit(1:end)))+"]";
38     yAxis = char(unique(TDCSN_SIM.output(1:end)))+" ["+ char(unique(TDCSN_SIM.unit(1:end)))+"]";
39
40     % scatter plot
41     figure('units','normalized','outerposition',[0 0 1 1])
42     figure(k);
43     scatter(VIN_DIFF_raw.value(1:end), TDCSN_SIM.value(1:end))
44     xlabel(xAxis); ylabel(yAxis); title(corners.Corner(k))
45
46     for param_i = 1 : height(params)
47         if (strcmpi(params.Test(param_i), unique(VIN_DIFF_raw.test(1:end))) ...
48         && strcmpi(params.Corner(param_i),unique(VIN_DIFF_raw.corner(1:end))) )
49             break; % find index which params match to
50         end
51     end
52
53     corner_i=char(corners.Corner(k));
54     corner_i(~ismember(corner_i,['A':'Z' 'a':'z'])) = ''; %remove any special char
55     tau_i = find(string(tau_LUT.Properties.VariableNames) == corner_i); %find index in the lut which contains a specific corner
56
57     [V, I] = max(TDCSN_SIM.value); % find value and index for max(TDCSN_SIM)
58     tau = tau_LUT{:,tau_i};        % expected tau for a given corner
59     voff = VIN_DIFF_raw.value(I);  % input value that cause the comparator to spend most time
60     %voff=CompOffset_raw.value;
61     td = min(TDCSN_SIM.value)/5;  % delay for everything else
62     errfun_msq = @(x) sum( (TDCSN_SIM.value -(-x(1)*log(abs(VIN_DIFF_raw.value-x(2))) + x(3)) ).^2);
63     res_t{k,1:3} = fminsearch(errfun_msq, [tau, voff, td]);
64
65     % The estimated unknown variables
66     tau_est    = res_t{k,1};
67     offset_est = voff;%res_t{k,2};
68     td_est     = res_t{k,3};
69
70     % TDCSN function
71     TDCSN_fun = @(x) (-tau_est*log(abs(x)) + td_est);
72     TDCSN_fun_plot = @(x) (-tau_est*log(abs(x-offset_est)) + td_est);
73
74     % Calculate lsb and lsb/2
75     lsb=lsbFun([1, N]);
76     lsbHalf= lsb/2;
77
78     % Add data to table
79     res_t{k,4} = TDCSN_fun(lsbHalf/2);  % Calculate decision time for vin=lsb/4
80     %res_t{k,5} = TDCSN_fun(lsbHalf/2); % Calculate decision from comparator to MD detector with vin=lsb/4
81     res_t{k,5} = res_t{k,4}-TDCSN_fun(lsbHalf/2+lsb/8); % Calculate required time resolution with a fault error of lsb/8.
82     res_t{k,6} = lsb;
83     res_t{k,7} =string(corners.Corner(k));
84
85     % phase noise impact on voltage fault related to LSB.
86     x=lsb/(10^((TDCSN_fun(lsbHalf)-td_est)/-tau_est)-  10^((TDCSN_fun(lsbHalf)-td_est+6e-12)/-tau_est)); % x is the error factor below lsb, as lsb/x
87     %error_source = lsb/x;
88
89     if (plotEstCurve)
90         hold on;
91         scatter(VIN_DIFF_raw.value, TDCSN_fun_plot(VIN_DIFF_raw.value));
92         %fplot( @(x) (-tau_est*log(abs(x-offset_est)) + td_est), 'LineWidth',4)
93         xlim([min(VIN_DIFF_raw.value) max(VIN_DIFF_raw.value)])
94         xline([voff-lsbHalf/2 voff voff+lsbHalf/2],'-',{'-LSB/4','Voff','+LSB/4'})
95         yline([res_t{k,4}],'-',{'TDCSN(LSB/4)'})
96
97         legend(yAxis+"(sim)",yAxis+ "(calc)")
98
99         cell_array=[];
100        for j = 1 : width(params(param_i,:))
101            column=char(params(param_i,j).Properties.VariableNames);
102            row = params(param_i,j).Variables;
103            cell_array{j} = [string(column)+': '+string(row)];
104        end
105        cell_array{j+1} = ["N"+': '+string(N)];
106        cell_array{j+2} = ["V_LSB"+': '+string(lsb)+"V"];
107
108        dim = [0.15 0.5 0.3 0.3];
109        annotation('textbox',dim,'String',cell_array,'FitBoxToText','on', 'Interpreter', 'none');
110        set(gca,'FontSize',20) % Creates an axes and sets its FontSize to 18;
111        saveas(figure(k), resPath + "\" + corners.Corner(k) + ".png");
112    end
113 end
114   writetable(res_t,resPath + 'res.csv')
115 end
```

B Implementation of Plotting the Time Constraint Region

```
clear
clc
close all

lsbFun = @(g) (g(1)/(2^(g(2)))); %LSB equation; VDD/(2^N)
N = 8; % number of bits

plotEstCurve = true;
URD = true;
resPath = "C:\results_location_were_files_are_located\";

corners = unique(readtable(resPath+"corners.csv", 'Delimiter', ','));
params  = unique(readtable(resPath+"param.csv",   'Delimiter', ','));

% table to store results
len=size(corners,1);
z=zeros(cast(len, 'int16'),1);
str=strings(cast(len, 'int16'),1);
res_t = table(z,z,z,z,z,z,z,str, 'VariableNames', {'tau [s]' 'V_{off} [V]' 'T_{d} [s]' 'TDCSN(LSB/4) [s]' ...
'delay range [s]' 'delta t [s]' 'lsb [V]' 'corner'});
if URD
for k = 1 : length(corners.Corner(1:end))

    VIN_DIFF_raw =   readtable(resPath + "\" + corners.Corner(k) + "\" + "VinDIFF.csv");
    MD_SIM_raw =     readtable(resPath + "\" + corners.Corner(k) + "\" + "MD.csv");
    TDCSN_SIM_raw =  readtable(resPath + "\" + corners.Corner(k) + "\" + "TDCSN.csv");
    CompOffset_raw = readtable(resPath + "\" + corners.Corner(k) + "\" + "CompOffset.csv");

    VIN_DIFF=VIN_DIFF_raw(1+1:end-2, :);  % remove first sample because of a reset in the first sample
    MD_SIM=MD_SIM_raw(1+1+2:end, :);      % 2: because MD is 2 clk behind. 1: reset

    % create axis description by combinding output name and unit
    xAxis = char(unique(VIN_DIFF.output(1:end)))+" [" + char(unique(VIN_DIFF.unit(1:end)))+"]";
    yAxis = char(unique(MD_SIM.output(1:end)))  +" [" + char(unique(MD_SIM.unit(1:end)))+"]";

    % scatter plot
    figure('units','normalized','outerposition',[0 0 1 1])
    figure(k);
    scatter(VIN_DIFF.value(1:end), MD_SIM.value(1:end))
    xlabel(xAxis); ylabel(yAxis); title(corners.Corner(k))

    for param_i = 1 : height(params)
        if (strcmpi(params.Test(param_i), unique(VIN_DIFF.test(1:end))) ...
        && strcmpi(params.Corner(param_i),unique(VIN_DIFF.corner(1:end))) )
            break; % find index which params match to
        end
    end

    % The comparator voltage offset is located at the largest decision time
    %[V, I] = max(TDCSN_SIM_raw.value(2:end-2, :)); % find value and index for max(TDCSN_SIM)
    %voff = VIN_DIFF.value(I);      % input value that cause the comparator to spend most time

    % if one rather wants to use the simulation estimated offset value
    voff=CompOffset_raw.value;
    corner_i=char(corners.Corner(k));
    corner_i(~ismember(corner_i,['A':'Z' 'a':'z'])) = ''; %remove any special char

    %voff=CompOffset_raw.value;
     % Calculate lsb and lsb/2
    lsb=lsbFun([1, N]);
    lsbHalf= lsb/2;

    if (plotEstCurve)
        xlim([min(VIN_DIFF.value) max(VIN_DIFF.value)])
        xline([voff-lsbHalf/2 voff voff+lsbHalf/2],'-',{'-LSB/4','Voff','+LSB/4'})
        saveas(figure(k), resPath + "\" + corners.Corner(k) + ".png");
        %legend(yAxis+"(sim)",yAxis+ "(calc)")
        cell_array=[];
        for j = 1 : width(params(param_i,:))
            column=char(params(param_i,j).Properties.VariableNames);
            row = params(param_i,j).Variables;
            cell_array{j} = [string(column)+': '+string(row)];
        end
        cell_array{j+1} = ["N"+': '+string(N)];
        cell_array{j+2} = ["V_LSB"+': '+string(lsb)+"V"];
        dim = [0.15 0.5 0.3 0.3];
        annotation('textbox',dim,'String',cell_array,'FitBoxToText','on', 'Interpreter', 'none');
        set(gca,'FontSize',20) % Creates an axes and sets its FontSize to 18;
        saveas(figure(k), resPath + "\" + corners.Corner(k) + ".png");
    end
end
  writetable(res_t,resPath + 'res.csv')
end
```

C Binary Search Veriloga Voltage Offset Estimator

```
// VerilogA for MSc, binarySearch, veriloga
// Written by: Ole-Georg Berg 19.10.2021


`include "constants.vams"
`include "disciplines.vams"

module binarySearch(clk, in, out);

input clk, in;
output out;
electrical clk, in, out;

real CompIn, VTH, VTL;

parameter real slack = 10p from [0:inf];
parameter real VDD   = 1.2 from [0:inf];

integer falling_edge = -1;
integer rising_edge  =  1;
integer i;

analog begin

        @(initial_step or initial_step("dc","ac","tran","xf", "pss"))
begin
                VTH = VDD*0.9;
                VTL = VDD*0.1;
                CompIn = 0;
                i = 1;
                $info( ["Start␣binary␣offset␣search"]);
        end

        @(cross (V(clk)-VTH, falling_edge, slack, clk.potential.abstol)) begin
                $display("␣Vin␣is␣%g", V(in));
                if( V(in) > VTH ) begin
                        CompIn = CompIn + VDD/(1 << (i+1));  /*
                                                                Comparator output is high ->
                                                                test for larger Vin
                                                                until the decision flip
                                                             */
                end
                else if( V(in) < VTL ) begin
                        CompIn = CompIn - VDD/(1 << (i+1));  /*
                                                                Comparator output is low, ->
                                                                test for smaller Vin
                                                                until the decision flip
                                                             */
                end
                i = i + 1;
        end
        V(out) <+ CompIn;
end
endmodule
```

D Normal Distributed Random Input Generator Veriloga

```
// VerilogA for MSc, uniform_src, veriloga
// Written by : Ole-Georg Berg

`include "constants.vams"
`include "disciplines.vams"

module norm_src(input clk, input minus, output plus);

electrical clk, minus, plus;

parameter real vth = 0.6;
parameter real slack = 10n;

integer seed = 0;
parameter real mean = 0 from [-inf:inf];
parameter real std = 1m from [-inf:inf];
parameter real offset = 0 from [-inf:inf];

real dist;
integer falling_edge = -1;

analog begin

        @(initial_step or initial_step("dc","ac","tran","xf"))  begin
                seed = 1;
                dist = 0;
        end

        @(cross (V(clk)-vth, falling_edge, slack, clk.potential.abstol)) begin
                dist = $rdist_normal(seed, mean, std);
        end
        V(plus) <+ V(minus) + offset + dist;
end
endmodule
```

E Mismatch Analysis Based on [3]

```
%%%%%%%%%%%%%%%% mismatch analysis %%%%%%%%%%%%%%%%

Cu = 25.96e-15;              % Unit capacitance
sigma = 25.0e-18;
Cpar = 1000e-18;             %
Cpar_top = 1000e-18;         %
%stdc = 18.32e-18;           % Standard deviation
Cspec = Cu;                  % The spesific capactiance for the
                             % pelgrom mismatch coefficient

Kc = abs(Cu-sigma./Cu);      %  pelgrom mismatch coefficient
stdc = Kc*sqrt((Cspec*Cu)/2) ; % Standard deviation of the unit capacitance

% Create cap array with random numbers chosen from the
% Gaussian distribution with unit capacitor Cu as mean and a standard deviation stdc
for i=1:NUM_ADC_BITS
    for j=1:2^(i-1)
        CAR(i,j) = Cu + normrnd(0, stdc); % Temporary store the capacitor array
    end
end

% Sum the amount of Cu*normrnd for each capacitor weight
C = zeros(1,NUM_ADC_BITS);
for i = 1:NUM_ADC_BITS
    for j=1:2^(i-1)
        C(i) = C(i) + CAR(i,j);
    end
end

clear CAR;
C = fliplr(C)+Cpar;

% Input/output transition levels
DACout = zeros(1,2^(NUM_ADC_BITS)-1);
for i = 1:2^(NUM_ADC_BITS)-1
    D = fliplr(de2bi(i,NUM_ADC_BITS));  % digital word
    W = C.*D/(sum(C)+Cpar_top);         % DAC output weight
    DACout(i) = fs*sum(W);              % DAC output voltage
end

% Caclulate static nonlinearity metrics
DNL= diff(DACout)/mean(diff(DACout))-1;
INL = cumsum(DNL); % integration of the estiamted DNL

% plot
figure('units','normalized','outerposition',[0 0 1 1]);
figure(1);
h(1)=plot(1:2^(NUM_ADC_BITS)-1, DACout, 'LineWidth',4);
ylim([GND VREF]); xlim([0 2^(NUM_ADC_BITS)-1])
xlabel("Code"); ylabel("Fullscale Input"); title("Static Performance Evaluation")
h(2) = line(nan, nan, 'Linestyle', 'none', 'Marker', 'none', 'Color', 'none');
h(3) = line(nan, nan, 'Linestyle', 'none', 'Marker', 'none', 'Color', 'none');
legend_1 = 'V_{DACout}';
legend_2 =  "{\Delta}" + sprintf(' DNL: %.2f', max(DNL)-min(DNL));
legend_3 =  "{\Delta}" + sprintf(' INL: %.2f', max(INL)-min(INL));
leg=legend(h([1 2 3]), legend_1, legend_2, legend_3, 'location','best');
set(gca,'FontSize',20)
```

F Spectral Performance Evaluation

```
plotEstCurve = true;
resPath = "C:\Users\Ole-Georg\Documents\UIO\MSc_PROJECT\092022\09092022\run9\";
corners = unique(readtable(resPath+"corners.csv", 'Delimiter', ','));
params = unique(readtable(resPath+"param.csv", 'Delimiter', ','));

% table to store results
len=size(corners,1);
z=zeros(cast(len, 'int16'),1);
str=strings(cast(len, 'int16'),1);
snr_t = table(z,z,z,str, 'VariableNames', {'code' 'SNR' 'timeout' 'corner'});

for k = 1 : length(corners.Corner(1:end))

    DOUT = readtable(resPath + "\" + corners.Corner(k) + "\" + "VOUT.csv");
    EOC = readtable(resPath + "\" + corners.Corner(k) + "\" + "EOC.csv");
    DOUT_0 = readtable(resPath + "\" + corners.Corner(k) + "\" + "DOUT_0.csv");
    Pdissipated = readtable(resPath + "\" + corners.Corner(k) + "\" + "Pdissipated.csv");

    timeoutCTR =0;
    for lsb_i = 1 : length(DOUT_0.value)
        if(DOUT_0.value(lsb_i) > 0.6 )
            timeoutCTR = timeoutCTR +1;
        end
        DOUT_0.value(1:end);
    end
    timeout_rate=(timeoutCTR/length(DOUT_0.value))*100;

    for param_i = 1 : height(params)
        if (strcmpi(params.Test(param_i), unique(EOC.test(1:end))) &&
        strcmpi(params.Corner(param_i),unique(EOC.corner(1:end))) )
            break; % find index which params match to
        end
    end
    corner_i=char(corners.Corner(k));
    % ADC spec
    nbit=8;
    Tclk = 1/0.5e9; % 0.5 GHz

    Fs = (1./(Tclk*(nbit+3))); % sampling rate
    x=DOUT.value(1:end);       % samples
    N = length(x);             % Number of ADC samples

    n = [0:N-1];               % [0,1,2,4,5, ... ,N]
    t  = n*(N/Fs);
    T = N/Fs;
    freq = n/T;

    X = fft(x)/N*2;
    X=abs(X)/max(abs(X));                % normalize the fft data to 1.
    [V1, bx1] = max(X(2:N/2)); % find signal

    X_cpy = X(2:N/2);                    % Ignore DC, start at position 1
    freq_cpy =  freq(2:N/2);             % Ignore DC, start at position 1

    [V, bx] = max(X_cpy); % find signal
    spect_low_level = min(20*log10(X_cpy));
    spect_off_level = max(20*log10(X_cpy))

    spect_low_level = min(20*log10(X));
    %%Plot the spectrum:
    ax=figure('units','normalized','outerposition',[0 0 1 1]);
    figure(k);
    for i = 1 :(N/2)-1
        x = freq(i);
        y= 20*log10(X(i));
        line([x x],[y spect_low_level-1],'color',[0.36 0.54 0.66], 'LineWidth',2)
        hold on
    end

    Cont. at next page ...
```

F.1 Part Two

```
    ax=gca; ax.XAxis.Exponent = 6;
    xlabel('Frequency [Hz]','Interpreter','latex');
    ylabel('Magnitude [dBFS]','Interpreter','latex');
    xlim([min(freq(1:N/2)) max(freq(1:N/2))])
    ylim([spect_low_level-1 0])
    hold on;

    %%% SNR: Signal to Noise Ration %%%
    % Remove DC bin, Mask out the harmonics of the fundemental freq.
    % Then integrate noise power and relate it to the fundemental freq.

    X_cpy = X(2:N/2);                     % Ignore DC, start at position 1
    freq_cpy =  freq(2:N/2);              % Ignore DC, start at position 1

    % mask out harmoincs of the signal
    ki=1;
    while((bx+bx*ki)<N/2)
        X_cpy(bx+bx*ki)=0;
        if (bx-bx*ki>0) % harm. below the fundemental tone
            X_cpy(bx-bx*ki)=0;
        end
        ki=ki+1;
    end

    %Noise bins: all except signal bin
    As = 10*log10(X_cpy(bx).^2);
    X_cpy(bx)=0;
    An = 10*log10(sum(X_cpy.^2)); % fft**2 is the power. Power of Noise floor [dB]
    SNR = As - An;
    SNR = SNR + 10*log10(N/2);    % The DFT noise floor is 10log10(N/2)dB below the
                                  %actual noise floor (assuming white noise)

    %%% SFDR: spurious free dynamic range %%%
    % Find range were we are free from spurious
    [V_2nd, bx_2nd] = max(X_cpy); % find index were the largest spurious appears
    % take the difference between fundemetal tone and spurious
    SFDR = abs(10*log10(X_cpy(bx_2nd).^2)-10*log10(X(bx_2nd).^2));

    %%% THD: Total Harmonic Distortion %%%
    % Find power harmonic of the fundemental freq.
    X_cpy = X(2:N/2);                     % Ignore DC, start at position 1
    freq_cpy =  freq(2:N/2);              % Ignore DC, start at position 1

    % find harmoincs power of the fundemental signal, sum contributions
    ki=1;
    sq_sum=0.0;
    while((bx+bx*ki)<N/2)
        sq_sum = sq_sum + (X_cpy(bx+bx*ki)^2);
        if (bx-bx*ki>0) % harm. below the fundemental tone
            sq_sum = sq_sum + (X_cpy(bx-bx*ki)^.2);
        end
        ki=ki+1;
    end
    thd = sqrt(sq_sum)/X_cpy(bx);
    THD = 20*log10(thd);

    SINAD=-10*log10(10^(-SNR/10)+10^(THD/10));
    ENOB=(SINAD-1.76)./6.02;
    FoM=(Pdissipated.value)./(2^ENOB*(Fs/2));

    snr_t{k,1} = params.FINE_CODE(k,:);
    snr_t{k,2} = SNR;
    snr_t{k,3} = timeout_rate;
    snr_t{k,4} = string(corner_i);

    if (plotEstCurve)
        hold on;
        yline([-SNR+spect_off_level],'-',{'Noise Floor Relativ To Carrier'},'Interpreter','latex', 'LineWidth',2,'FontSize',20)
        cell_array=[];
        cell_array{1} = ["No. Samples"+': '+string(N)];
        cell_array{2} = ["Sampling Freq"+': '+string(num2eng(round(Fs,2)))+"Hz"];
        cell_array{3} = ["Carrier  Freq"+': '+string(num2eng(round(freq(bx),2)))+"Hz"];
        cell_array{4} = ["LSB Timeout Rate"+': '+string(round(timeout_rate,1))+" %"];
        cell_array{5} = ["SNR"+': '+string(SNR)+" dB"];
        cell_array{6} = ["SFDR"+': '+string(SFDR+abs(10*log10(X(bx_2nd).^2)))+" dBFS"];
        cell_array{7} = ["SFDR"+': '+string(SFDR)+" dBc"];
        cell_array{8} = ["SINAD"+': '+string(SINAD)+" dB"];
        cell_array{9} = ["THD"+': '+string(THD)+" dB"];
        cell_array{10} = ["ENOB"+': '+string(ENOB)+" bits"];
        cell_array{11} = ["FoM"+': '+string(num2eng(round(FoM,15)))+"J/conversionstep"];
        cell_array{12} = ["Avg Power Dissipation"+': '+string(num2eng(round(Pdissipated.value,5)))+"W"];

        dim = [0.15 0.5 0.3 0.3];
        annotation('textbox',dim,'String',cell_array,'FitBoxToText','on', 'Interpreter', 'none');
        set(gca,'FontSize',20) % Creates an axes and sets its FontSize to 18;

        saveas(figure(k), resPath + "\" + "fft_" + corners.Corner(k) + ".png");
    end
end

writetable(snr_t,resPath + 'snr.csv')
```