Linköping University | Department of Electrical Engineering
Master thesis, 30 ECTS | Electronics
2022 | LiTH-ISY-EX–22/5480–SE

Comparator Design for High-Speed ADCs

Pelle Lund - johlu110@student.liu.se

Supervisors: **Erik Backenius**
Ericsson, Linköping
Alireza Saberkari
ISY, Linköping University

Examiner: **Mark Vesterbacka**
ISY, Linköping University

Linköping University
SE-581 83 Linköping
013-28 10 00, www.liu.se

Copyright

Upphovsrätt

Abstract

As wireless communication is ever-evolving, demanding higher data speeds, the requirements increase for the ADC, and the requirements for the comparator, which is one of the main building blocks, increase as well. The primary purpose of the comparator is to compare two voltage levels and provide a logic output. One significant advantage of dynamic comparators is that they are more power-efficient than traditional comparators. There exist many different architectures for dynamic comparators. In this thesis, the most promising designs are optimized and evaluated over various parameters, such as speed, noise, offset, and hysteresis, while minimizing power consumption. The thesis includes a traditional StrongARM-latch, a double tail, and four triple tail comparators. The StrongARM-latch was the most power-efficient design while all the parameters were within the requirements, which was unexpected.

Acknowledgements

Firstly I would like to say thanks to Ericsson AB, who provided me with the opportunity to do this master thesis.

Secondly, a special thanks to Erik Backenius for all the help and guidance throughout my master thesis. It could not have been done without his support, patience, and endless answers to sometimes stupid questions through our many discussions.

I also want to thank Prakash Harikumar and Christer Jansson for their help with many technical discussions and support.

Lastly, I want to thank Alireza Saberkari for helping me proofread the thesis, and Mark Vesterbacka, my examiner at Linköpings University, for quick feedback and guidance.

Contents

List of Figures

List of Tables

Acronyms

ADC Analog-to-Digital converter.

CDF Cumulative Distribution Function.

CM Common-mode.

DAC Digital-to-Analog converter.

MOSFET Metal Oxide Semiconductor Field Effect Transistor.

NMOS N-type Metal Oxide Semiconductor.

PMOS P-type Metal Oxide Semiconductor.

PSS Periodic Steady-State.

RF Radio Frequency.

SAR Successive-Approximation-Register.

Nomenclature

V_{ss}	Ground voltage
V_{dd}	Supply voltage
V_{dac}	Voltage reference
V_{in}	Input voltage
V_{out}	Output voltage
V_{th}	Threshold voltage
ΔV	Differential voltage

Chapter 1

Introduction

An Analog-to-Digital converter (ADC) is a critical component in modern Radio Frequency (RF) systems. To handle the amount of data the users require, it needs to be fast and precise. The ADC's architecture can vary, but in this report, a Successive-Approximation-Register (SAR) ADC will be used as an example, which consists of a SAR block, a comparator, and a Digital-to-Analog converter (DAC) block.

1.1 Motivation

As wireless communication is ever-evolving, demanding higher data speeds, the requirements increase for the ADC, and the requirements for the comparator increase as well. When it comes to the performance of the ADC, the comparator is crucial in terms of speed, accuracy, and power consumption [1]. The parameters of the comparator this report will focus on are speed, input-referred noise, kickback noise, offset, hysteresis, and power consumption.

Traditionally, when data speeds were nothing like today, architectures like the StrongARM-latch were popular. Due to trade-offs between different parameters and limitations within the architecture in older technology, their performance was limited. Today, there are many different architectures, where the most promising will be evaluated in this report.

To reduce the problems with different trade-offs, architectures like the double-tail latch became popular in high-speed circuits. This architecture has a first stage acting as a pre-amplifier and a second stage acting as an amplifying latch. This architecture divides the steps within the circuit, giving the designer more room for optimization without compromising other parameters [2].

In later days, designers have been using the triple-tail latch to gain even more performance from the comparator. It divides the steps into another amplifying latch, letting the designer optimize the parameters even more independently. Of course, this comes with

trade-offs as well. The more transistors between input and output in the architecture, the longer the delay will be for the output. While it is possible to increase the speed, it will also consume more power.

1.2 Purpose

This thesis purpose is to investigate different high-speed comparator architectures. The most promising architectures should be:

- Implemented as transistor schematics.
- Simulated and initial sizing to verify basic operation.
- Introduce estimation of key performance parameters using expressions and different test setups.
- Optimize the selected architectures with the help of optimization tools and compare the results of the different architectures.

To get a fair comparison between the different architectures, they will be designed to meet the same speed and noise criteria while minimizing the power consumption. Since the optimization is done with a tool rather than manually by a designer, the comparison between the architectures probably will be fairer.

1.3 Delimitation

The Metal Oxide Semiconductor Field Effect Transistor (MOSFET) process technology used in this thesis work is protected under a non-disclosure agreement. A reference to a parameter instead of numbers will be presented in this thesis. An effect of this might be that the results will be valid for the technology used but hard to replicate in another technology.

The thesis work is limited to 20 weeks, 800 hours.

The design is limited to the schematic level, so the layout parasitics affecting the performance will not be considered.

The optimization tool used in the thesis is a tool within Cadence trying to solve the problem for a global minima/maxima. However, since the number of available test points is extremely big, there is no guarantee that the found solution is the optimal solution.

1.4 Requirements

The designs should be able to handle a set of requirements, for $-40°$ and $125°$, at both V_{dd} $\pm 5\%$ at a clock frequency at F_0, where the goal is to keep the parameters precisely withing the requirement and minimize for power consumption. The maximum allowed input-referred noise is set to $\overline{V_{n,in_0}^2}$ and the hysteresis should be maximum 6% of this value. The clock frequency can also be described as

$$F_0 = \frac{1}{T_0} \iff T_0 = \frac{1}{F_0} \tag{1.4.1}$$

and the requirement for decision time is set for two different cases, one where the differential input is equal to $\overline{V_{n,in_0}^2}$ and another with $10 \cdot \overline{V_{n,in_0}^2}$. The decision time should be less than 35% of T_0 in the first case and less than 28% of T_0 in the second case. In all speed tests, the reset time should be less than 18% of T_0.

The output load, C_{out}, is used as a capacitance to ground for the simulations, and the input equivalent load should be less than C_{in}. The maximum allowed energy per cycle is E_0, but the goal is to minimize this value. All the requirements can be seen in 1.4.1.

Parameter	Requirements
Temperature	-40° *to* 125°
Supply Voltage	Vdd+- 5%
Clock Frequenzy	$F_0 = \frac{1}{T_0}$
Input referred noise	$< \overline{V_{n,in_0}^2}$
Decision time @ $\overline{V_{n,in_0}^2}$	$< 35\%\ T_0$
Decision time @ $10 \cdot \overline{V_{n,in_0}^2}$	$< 28\%\ T_0$
Reset time	$< 18\%\ T_0$
Energy/cycle	$< E_0$
Input equivalent load	$< C_{in}$
Output load	C_{out}
Hysteresis	$< 6\%\ \overline{V_{n,in_0}^2} = V_h$

Table 1.4.1: Requirements of the comparators.

Chapter 2

Theory

This chapter will discuss the most important parameters of the ADC and comparator.

2.1 ADC Theory

As its name, Analog-to-Digital converter, it is used to convert an analog signal into a digital one. In figure 2.1.1 a 3-bit SAR ADC can be seen. It consists of three blocks, SAR logic, a DAC, and a comparator.

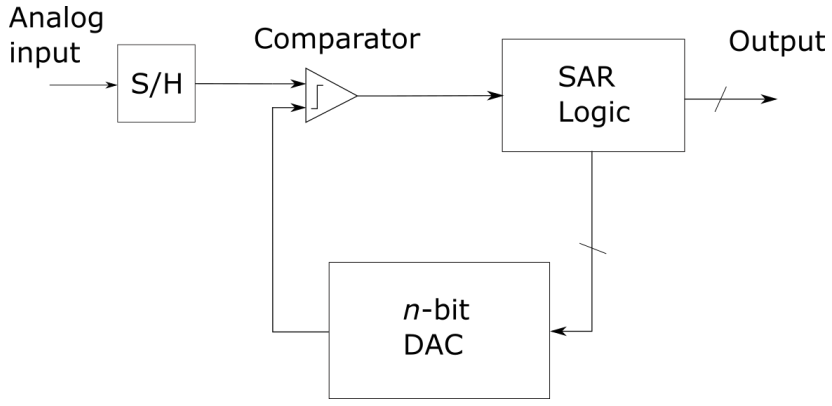


Figure 2.1.1: Overview of an SAR ADC.

The ADCs task is, as the name tells, to convert an analog signal into a digital bit pattern [3]. The signal V_{in} is the analog signal which should be converted, and V_{dac} is the reference signal from the DAC to the comparator. The binary search pattern is similar to figure 2.1.2 where the first comparison is between V_{in} and $\frac{1}{2}V_{dac}$. Depending on if V_{in} is larger or smaller than V_{dac}, the next comparison will be $\frac{1}{4}V_{dac}$ or $\frac{3}{4}V_{dac}$. V_{dac} gets cut in half every cycle in the direction of V_{in} to try to get an as precise result as possible, with the more bits, the better accuracy.

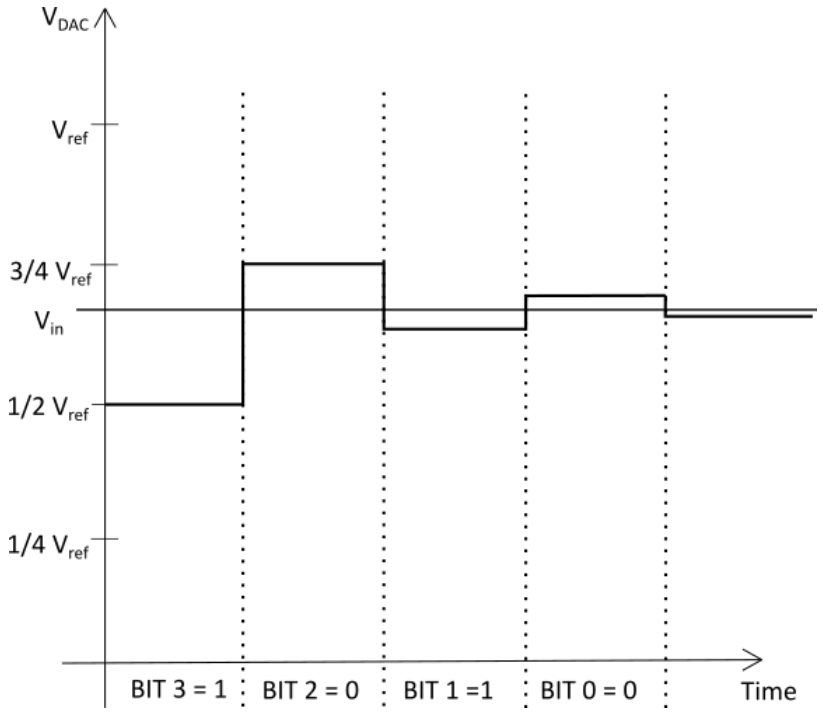


Figure 2.1.2: Searchpattern in an ADC.

Several factors limit the ADCs speed, resolution, and accuracy, but the focus will be the comparator in this thesis.

2.2 Comparator Theory

The comparator is a vital part of the ADC, and to have a good performance from the ADC, the comparator design is essential. The function of the comparator is to compare two analog signals and generate a logic output, so basically, it is operating as a 1-bit ADC [4]. Different kinds of comparators exist, but this report will focus on dynamic comparators for ADCs.

2.2.1 Basics

The analog inputs to the comparator are the reference signal from the DAC and the input to be measured, from here on, called V_{in_p} and V_{in_n}. In figure 2.2.1 the behavior of an ideal comparator is shown, comparing the analog signal V_{in_p} towards the reference signal V_{in_n}, providing the logic level for V_{out}. A general symbol for the comparator is also shown.

As components are never ideal, the behavior of the comparator is more similar to figure 2.2.2, where rise times will result in a delay at the output.

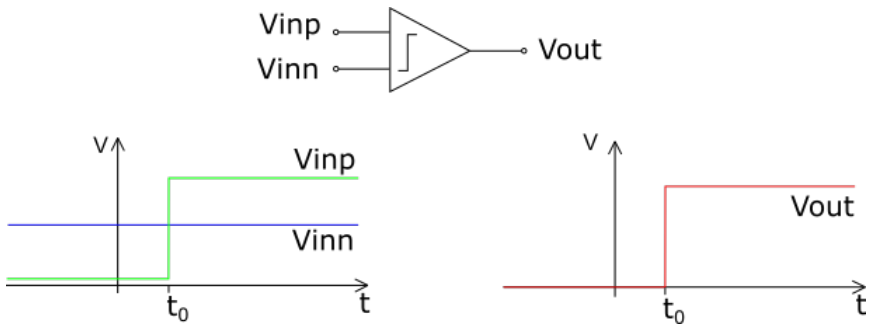


Figure 2.2.1: A simple illustration of the behaviour in an ideal comparator.

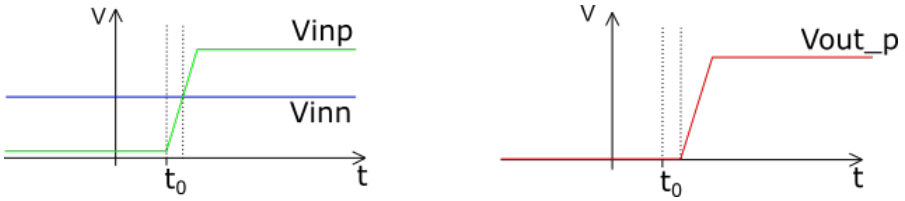


Figure 2.2.2: A simple illustration of the behaviour in a non-ideal comparator.

In dynamic comparators, the clock is used to evaluate on one edge and to reset on the other. Two outputs are also used, where one corresponds to the logic output, and the other corresponds to the inverse of the first one, as shown in figure 2.2.3.

Figure 2.2.3: A simple illustration of the behaviour in a comparator with two outputs.

2.2.2 Phases

A dynamic latched comparator has two different phases: the reset and the evaluation phases. During the evaluation phase, the comparator will compare the inputs, and generate the corresponding logic output depending which input voltage is larger, while during the reset phase, the comparator will try to reset all the internal voltage levels. If the comparator does not have a sufficient long reset, the voltage levels will act as memory, affecting the following comparison. A few cycles of a comparator's behavior are shown in figure 2.2.4.

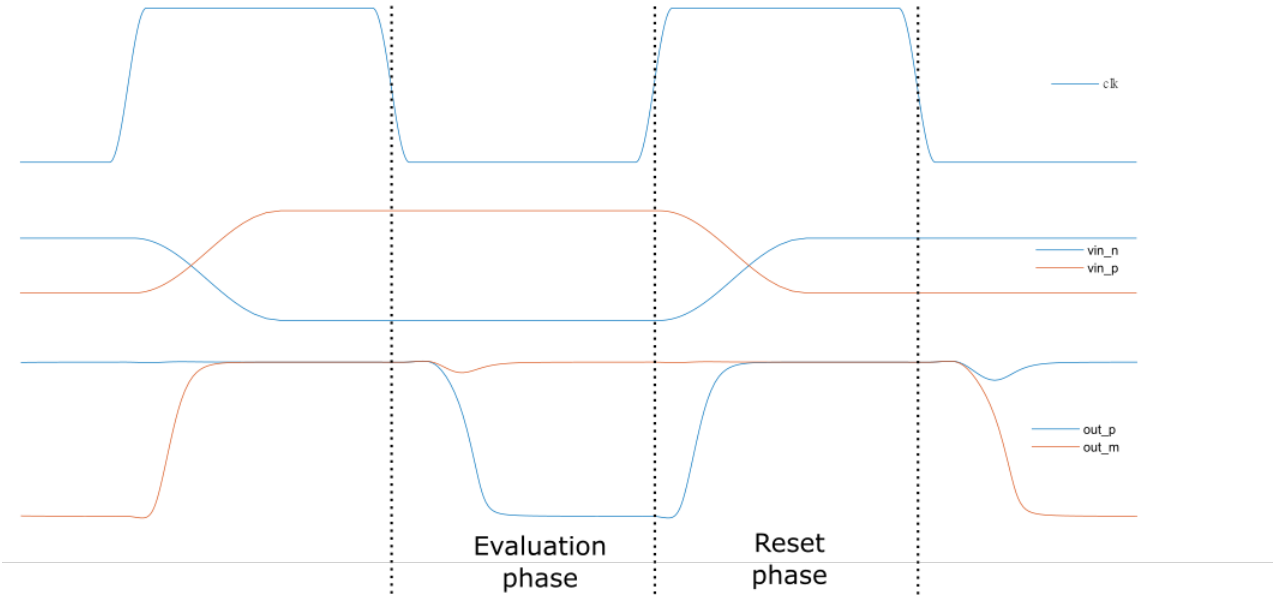


Figure 2.2.4: Illustration of the reset and evaluation phases.

2.2.3 Schematic

In figure 2.2.5 the StrongArm-latch comparator is shown as a reference to the rest of this chapter.

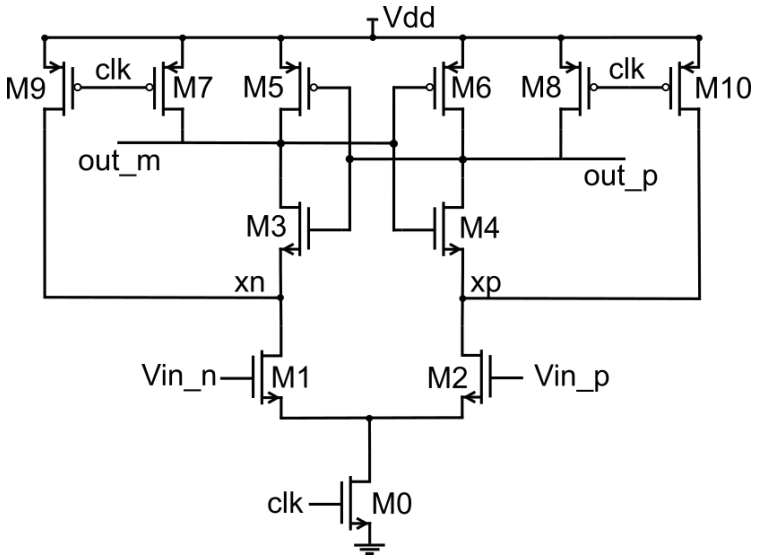


Figure 2.2.5: Schematic of a StrongARM-latch.

The operation starts with CLK going low, and the nodes xn, xp, out_m and out_p are precharged to V_{dd}. When CLK goes high, M1 and M2 will start to discharge nodes xn and xp

until the nodes drop to roughly $V_{dd} - V_{th,3,4}$. When M3 and M4 turn on, they will activate M5 or M6, where one output is pulled back to V_{dd} and the other output is discharged, creating a logic output. One significant advantage of this kind of dynamic comparator is that it does not consume any static power due to the tail transistor, M0, being switched on/off by the clock input signal instead of biased to a constant current [5].

2.3 Important parameters

There are several parameters to take in account when designing a comparator, as presented here.

2.3.1 Speed

Speed is one of, if not the most, important parameter when designing a comparator. The overall speed must the ADC must be matched, therefore, it is critical that a comparison can be made during one evaluation phase.

The speed of the comparator, $t_{decision}$, is usually measured from 50% of the evaluating edge on the clock until the difference between V_{out_p} and V_{out_m} is 50% of V_{dd}. However, in this report, it is measured when the difference is 90% of V_{dd}, to make sure that the decision has been taken. The reset time, t_{reset} is measured similarly, but at the reset edge on the clock, as shown in figure 2.3.1.

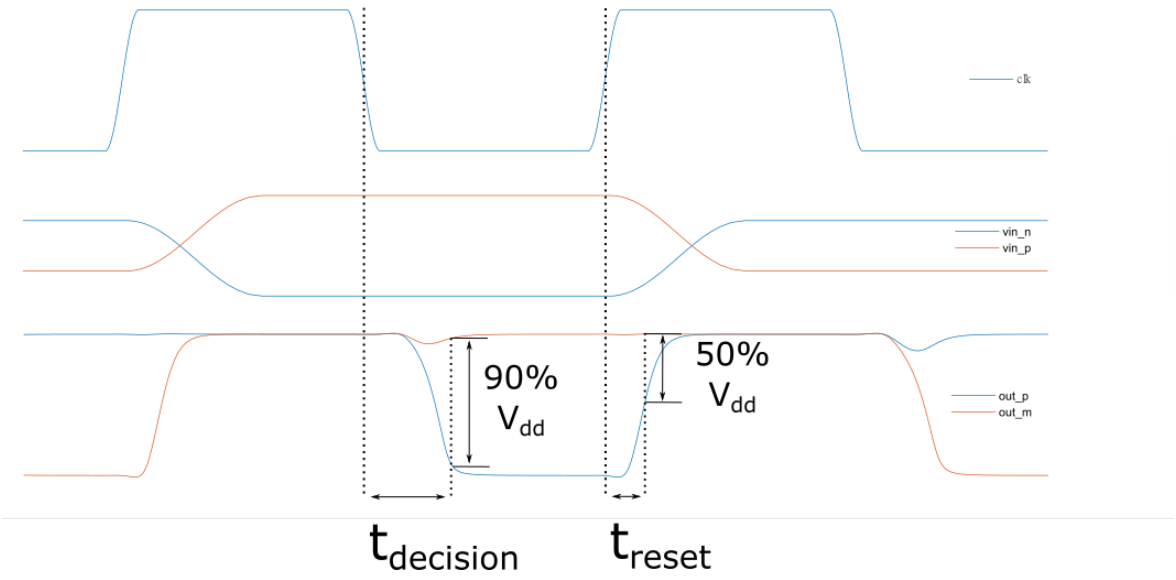


Figure 2.3.1: Comparator outputs plotted, with indications for speed calculation.

The comparator must compare the signal during one clock cycle and provide a stable output before the next cycle. As the difference in V_{in_p} and V_{in_n} gets smaller, the slower the comparison will be [6].

2.3.2 Power consumption

As in almost every electronic design, low power consumption is desired. One significant advantage of the dynamic latched comparator compared to the pre-amplified based comparator is that it does not consume any static power, if leakage currents are neglected.

The power consumption is calculated by

$$P_{avg} = I_{Vdd,avg} \cdot V_{dd}, \tag{2.3.1}$$

where the more interesting parameter might be energy per cycle, which is calculated by

$$E_{cmp} = \frac{I_{Vdd,avg} \cdot V_{dd}}{F_{clk}} \tag{2.3.2}$$

To have a fair comparison between the circuits, the power consumption from the clock path is also included and calculated by

$$E_{clk} = \int \left| \frac{I_{clk} \cdot V_{dd}}{2 \cdot N_{cycles}} \right| dt, \tag{2.3.3}$$

where I_{clk} is the current of the clock input. The resulting total energy per cycle is calculated by

$$E_{cycle} = E_{cmp} + E_{clk} \tag{2.3.4}$$

2.3.3 Offset

Due to mismatch in the manufacturing process of the transistors, the input pairs are not entirely identical and will therefore have slightly different threshold voltages. This will mean that their behavior is not exactly the same, which will result in an offset in the input pairs. When the voltage is exactly the same on the gates of the input transistors, one will draw a somewhat larger current and the corresponding drain node will move faster and give an output that might not be correct.

Other transistors than the input pairs will also contribute to the offset, but in general, the input pairs are the most dominant source to offset due to a large gain from the input pairs before other transistors activate [7].

2.3.4 Input-referred noise

Similarly, as to the offset, the input-referred noise will have its main component from the input pair transistors [8]. In many circuits, the input-referred noise can be approximated by measuring the output noise and divide that by the gain,

$$\overline{V_{n,in}^2} = \frac{\overline{V_{n,out}^2}}{A_v} \tag{2.3.5}$$

However, due to the comparator's behavior, which produces a digital output, this formula can not be used with a transient simulation. Another way to estimate the input-referred noise is to run long transient simulations with steady input. This method might be more accurate but is very time-consuming and not preferred in simulations during optimization.

A more time-saving approach is to run a Periodic Steady-State (PSS) simulation to estimate the gain and output-referred noise and, from there, calculate the input-referred noise with formula 2.3.6 [9].

2.3.5 Kickback noise

The big voltage swings in the drains of the input pairs cause something called kickback noise. Suppose the kickback noise levels are too high at the input during the evaluation phase. In that case, it might change the comparator output and provide an incorrect output when the differential input is small, which will degrade the ADCs accuracy [10].

2.3.6 Hysteresis

Hysteresis is when the comparator output goes from 0 to 1 at one input but will switch from 1 to 0 at a slightly different input. This effect can cause the comparator to behave as storing the old input and tends to keep it when there are small differences in the input or in the opposite direction. In some applications an hysteresis is favorable, for example in noisy circuits [11].

2.3.7 Input equivalent load

The equivalent input load, the capacitance over the input pairs, can be calculated by

$$C_{diff} = \frac{dQ}{dV}, \tag{2.3.6}$$

during the reset phase of the comparator. The differential current at the gate of the input

transistors is integrated, divided by the difference in voltage and calculated with the following formulas

$$dQ = \int \left| \frac{I_{vin\_p} - I_{vin\_n}}{2} \right| dt \tag{2.3.7}$$

$$E_{clk} = \int \left| \frac{I_{clk} \cdot V_{dd}}{2 \cdot N_{cycles}} \right| dt, \tag{2.3.8}$$

$$dV = \left| vin\_p - vin\_n \right| \tag{2.3.9}$$

C_{diff} is the equivalent load between vin_p and vin_m, and the relevant measurement is the single-ended load. Capacitors in series can be calculated with the following formula

$$\frac{1}{C_{tot}} = \frac{1}{C_{serie}} + \frac{1}{C_{serie}} \tag{2.3.10}$$

and, therefore, $C_{load,in}$ is calculated by

$$C_{load,in} = 2 \cdot \frac{dQ}{dV}. \tag{2.3.11}$$

2.4 Process Corners

The fabrication of MOSFET is not perfect, which leads to parameter variations from wafer to wafer, over the same wafer and also over the same chip. Because of this, performance will differ between different chips, and this must be taken into consideration in the design to make sure that the yield is acceptable [12].

In figure 2.4.1, the speed of N-type Metal Oxide Semiconductor (NMOS) and P-type Metal Oxide Semiconductor (PMOS) is put into a rectangle, where the corners of the rectangle are the process corners, SS (slow-slow), FF(fast-fast), SF(slow-fast) and FS(fast-slow). The middle point is the typical corner.

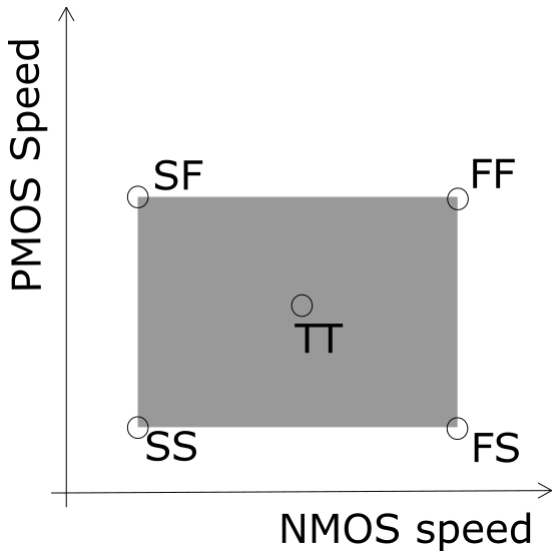


Figure 2.4.1: Process corner rectangle based on the speed of PMOS and NMOS.

The designer can be confident that the produced chip will have transistors somewhere within the rectangle, so during the design process, tests are made over different corners for different cases to make sure the chip will handle the fabrication variations.

The temperature and supply voltage variations is also considered in this thesis work. Here, every corner is simulated at $V_{dd} \pm 5\%$, T_{min} and T_{max}.

2.5 Circuit optimization

In a normal sweep in cadence, the tool will brute force the possible combinations without any mathematical insight. With the optimization tools, to find an optimal solution in the design space, the tool will use an iterative mathematical process. The tool itself does not have any knowledge of the circuit, but will work towards the optimal solution based on the designers mathematical expressions and the output from the design. The tool will try to minimize the cost, where a solution that meets the requirements stated in the expressions have a low cost, and a solution that does not meet the requirements have a higher cost [13].

Before starting to use the optimizing tools, it is important that the tests and expressions are carefully set by the designer. Also, setting the parameters for the transistors, the designer should match the relationship between matching pars, for example differential pairs. The workflow with the optimizer tools is described further in chapter 3.2.1.

2.5.1 Global Optimizer

The global optimizer is trying to find the global minima for the design space, and does not require a starting point. This is typically the first step during the optimization, to ensure that

the whole design space is considered. It is possible to run both optimizers in two different evaluation modes, full and conditional. Full evaluation means that it will evaluate the whole point for all the test points, while in conditional mode, it will only evaluate expression which is known to be far from the requirement. In the latter mode, if those points is better than the reference point, the rest of the expressions is also evaluated for that test point.

Typically the global optimizer is used with a stopping criteria, which can be if the requirement is met, a time limit, number of points limits, if there is no improvement within a number of points, or a certain number of points after the requirement is met.

2.5.2 Local Optimizer

The local optimizer requires a starting point, searching around that point until it will take the right direction towards the local minima. This is usually done after the global optimizer have found a good solution to start with, to ensure that the starting point is close to the global minima.

The local optimizer also has stopping criteria, which is if the requirements is met, time limit or number of points. To ensure that the tool is running through the whole local minima, all the stopping criterias can be unchecked, meaning that the tool will run freely until it is satisfied within the design space. In this phase of the optimization it is favorably to weight the most important criterias, for example to minimize power consumption, forcing the tool to prioritize a solution depending on the weighing. Weighing criterias too early in the optimization process can cause the tool to prioritize to minimize one parameter rather than meet the requirements over all parameters.

Chapter 3

Method

After a literature study, a few architectures were selected to be of interest. Almost every comparator has the input on nmos pairs, but in this thesis, all the architectures of interest will be modified to have the input on pmos pairs. Basically, the whole circuit is flipped upside down where nmos became pmos and vice versa. The clock signals are the same as with nmos input pairs to have the evaluation phase on the falling clock edge and the reset phase on the rising edge.

3.1 Comparators

The comparators of interest will be presented here, and the rejected comparators will not be presented. There are inverters on the outputs on all comparators, which is not seen in the schematics. On the outputs, there are capacitors with the value C_{out} connected to ground.

3.1.1 Strong-ARM Latch

The Strong-ARM latch is used mainly to reference older architectures and how their performance is compared to newer solutions. The architecture can be seen in figure 3.1.1

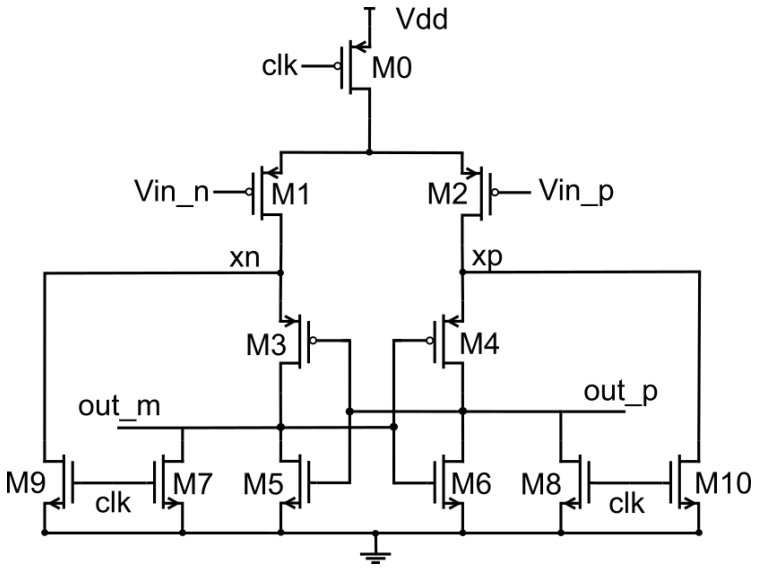


Figure 3.1.1: Schematic of comparator in [8], StrongARM-latch.

3.1.2 Double tail

The double-tail comparator of interest can be found in [14], and the schematic in figure 3.1.2. The main difference in comparing a double tail comparator and a conventional StrongARM-latch comparator is its two stages. The first stage acts as a pre-amplifier and the second stage as an amplifying latch.

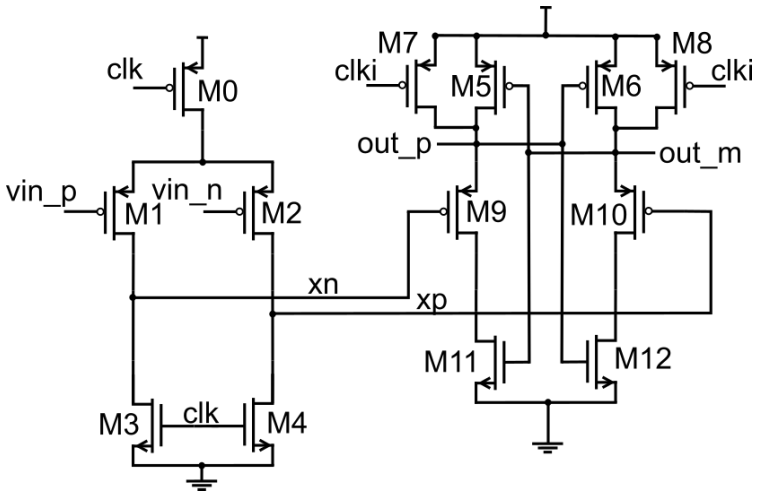


Figure 3.1.2: Schematic of comparator in [14].

3.1.3 Triple tail

Four different triple-tail comparators were chosen to be implemented in this project. There are some differences and focus in the architectures, which are interesting to compare. The

main difference between a double tail comparator and a triple tail comparator is the extra stage, acting as another amplifying latch stage.

3.1.3.1 Triple tail 1

The first triple tail architecture can be found in [15], and figure 3.1.3. Comparing the double-tail and this architecture, they are identical except for the extra stage found in the triple tail. The second and third stage is similar, whereas the third stage is basically the same as the second but flipped upside down and does not have a tail transistor. The clocked transistors in the second stage precharge the output nodes to V_{dd}, while in the third stage, they discharge the output nodes to V_{ss}.

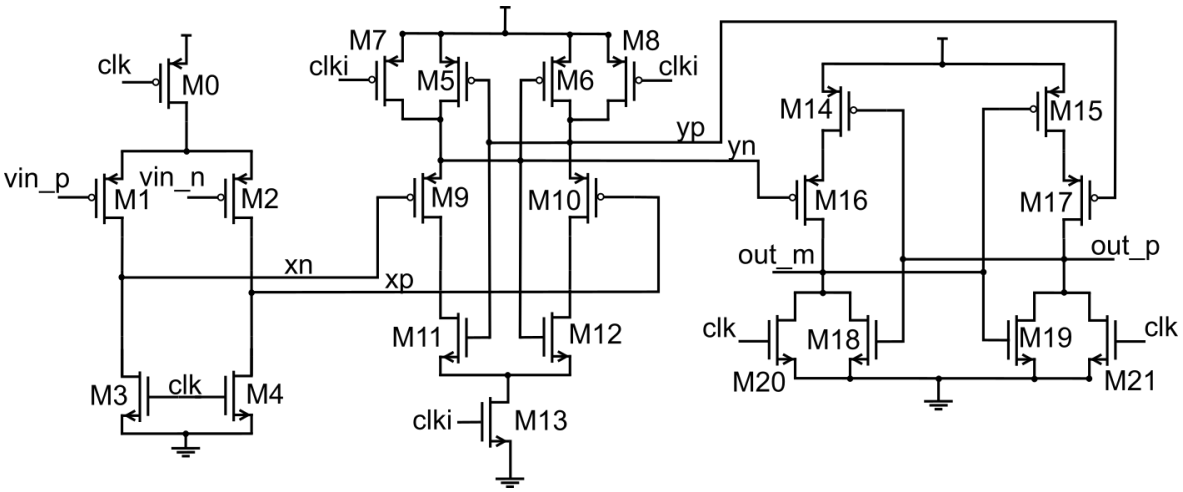


Figure 3.1.3: Schematic of comparator in [15].

3.1.3.2 Triple tail 2

The second architecture is found in [6], and figure 3.1.4.

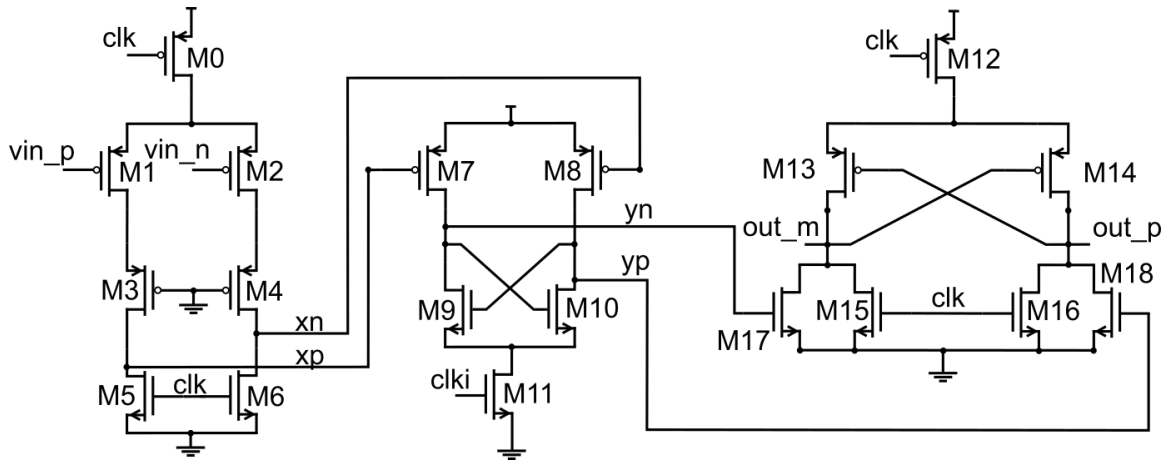


Figure 3.1.4: Schematic of comparator in [6].

3.1.3.3 Triple tail 3

The third architecture is found in [2], and figure 3.1.5. Comparing this architecture with the previous architecture, they are very similar. They both have an extra differential pair between the input pairs and the output from the first stage, but the gates are connected to Vss in triple tail 2 and cross-coupled in this architecture. The other difference is the extra transistors M15 and M16 in this architecture, providing a feed-forward path from the output of the first stage directly to the third stage.

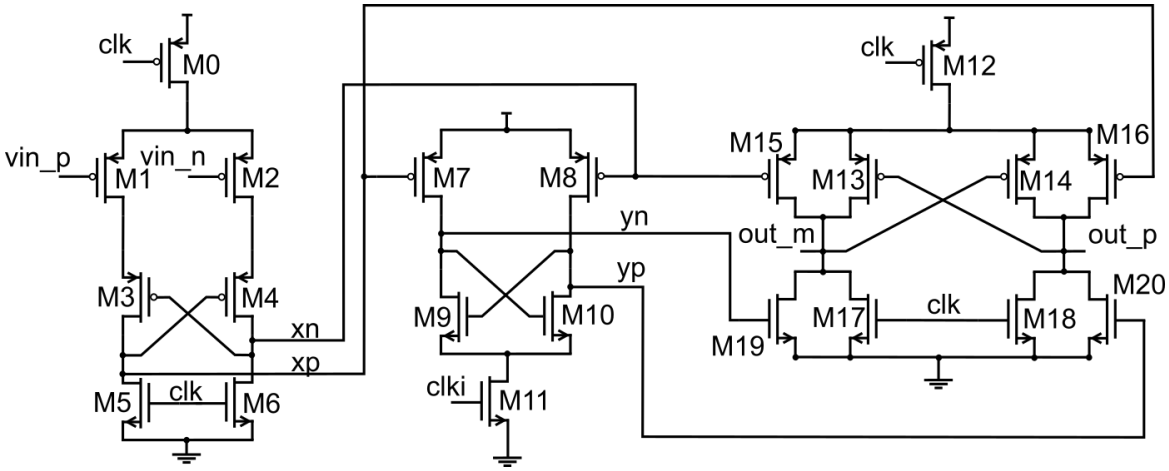


Figure 3.1.5: Schematic of comparator in [2].

3.1.3.4 Triple tail 4

The fourth and last triple tail is slightly different from the other architectures, and the actual implementation is confidential. However, the results from this comparator version are still included in this report.

3.2 Implementation

The comparators are tested for speed and power consumption in the initial phase to see the initial performance. From there, the most promising comparators are chosen for further optimization with additional tests.

3.2.1 Optimizing

For the first phase of the workflow, which can be seen in figure 3.2.1, the comparators are implemented as schematics with initial sizing. To make the comparators meet the requirement, the Global Optimizer tool is used. Initially, the transistor parameters are swept in large ranges with large steps, for example, from 2 to 30, with steps of 4, over the typical corner with nominal V_{dd} and temperature. The total number of sweep points is often larger than 10^{20} in all phases, but the tool handles such big numbers surprisingly well.

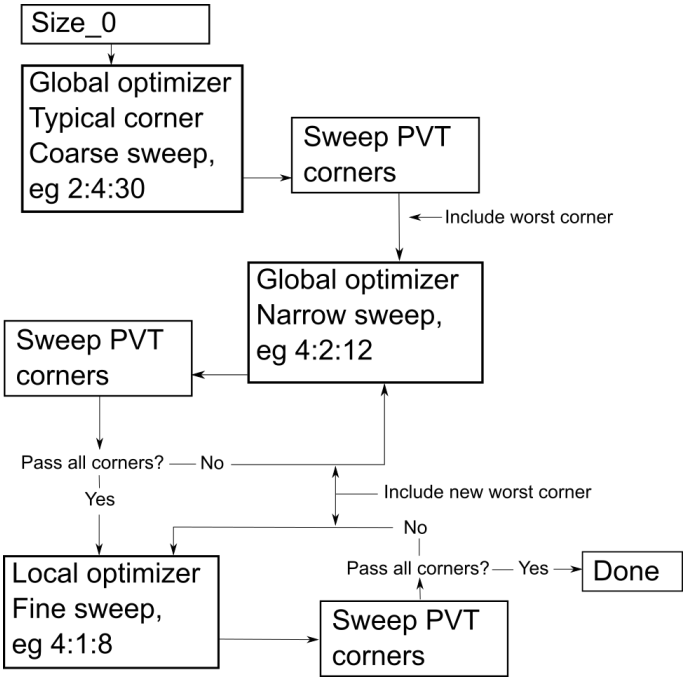


Figure 3.2.1: The optimization workflow.

Whenever the optimizer finds a solution that meets all the requirements over the typical corner, the solution is swept over all corners to see which corners perform the worst is in the various tests, typically ss, -40°, -5% V_{dd} for speed and ff, +125°, +5% V_{dd} for power consumption. Those worst corners are then added to the next phase.

In the next phase, still with Global Optimizer, the sweep range is narrower, for example, from 4 to 12, with steps of 2, with the solution from the first phase as the starting

point. Whenever the optimizer is at the bottom or top of the sweep range within any parameter, the optimizer is stopped, and the sweep range is changed to let the optimizer run freely.

When the optimizer finds a solution within requirements, all corners are swept again. However, if any corner fails this time, it is included, and the second phase is started over. If all the corners pass, phase two is done.

In the third and last phase of the optimization, the sweep is even narrower, for example, from 4 to 8 with steps of 1, and again with the solution from the previous phase as the starting point. This time the Local Optimizer is used to try to find the local minima rather than the global minima. Some parameters within the tests also get weighted to ensure the optimizer prioritizes, such as minimizing power consumption.

The Local Optimizer is run without any stopping criteria to let it decide by itself when it can not find any better solution. Similar to the previous phase, the final solution is swept over all the corners when the optimizer is done. If everything passes the tests, the design is considered done and optimized. If it fails any corner, that corner is included, and phase three is started over.

3.3 Tests

Here, the tests used on the comparators to determine the performance will be explained. During the optimization phase, the following parameters are tested:

- Speed
- Power consumption
- Hysteresis
- Input equivalent load

while the other parameters are tested on the final design. During the optimization, the common-mode voltage is regulated to nominally $\frac{V_{dd}}{2}$.

The speed is evaluated for two different cases, with inputs toggling between $\overline{V_{n,in_0}^2}$ and $119 \cdot \overline{V_{n,in_0}^2}$ for the first case, and inputs toggling between $10 \cdot \overline{V_{n,in_0}^2}$ and $20 \cdot \overline{V_{n,in_0}^2}$ for the second case. The requirement for speed is $< 35\% \, T_0$ for the first test and $< 28\% \, T_0$ for the second test. The reset time is also measured during this tests, with a requirement of $< 18\% \, T_0$ in both tests.

The power consumption is calculated from the speed tests, with the formulas described in chapter 2.3.2 where $< E_0$ is the requirement.

3.3.1 Input Referred Noise

To determine the input-referred noise levels, there is a separate test. Usually, the designer would run a very long transient noise simulation to estimate the noise levels from the gaussian distribution. However, that is not possible with such an extensive sweep range with the optimizer since it is very time-consuming. Instead, the circuit is tested with a PSS simulation, which is about 220x faster in simulation time [9].

In the later stages, to verify that the noise levels are within the requirements, $< \overline{V_{n,in_0}^2}$, the optimized circuit is simulated with a transient noise simulation. This is done in a few steps.

Firstly running the design for a large number of iterations with the expected input-referred noise, $\overline{V_{n,in_0}^2}$, as ΔV_{in}. When the simulations are done, the number of 1's and 0's are counted, and from there, get the probability for an output of 1. The next step is to calculate the inverse of the Cumulative Distribution Function (CDF) by

$$x = norminv(p) \tag{3.3.1}$$

in MatLab. Then the input-referred noise is calculated by

$$Sigma = \frac{\overline{V_{n,in_0}^2}}{x} \tag{3.3.2}$$

where sigma is the standard deviation, and the value for the input-referred noise.

3.3.2 Hysteresis

The hysteresis test is made similarly to the speed test. The main difference is that the clock is run at half the required frequency, but the reset period is kept the same. That means the comparator has three times as long time to make a decision, but the reset time is the same. Here one evaluation phase is triggered just after a recently changed input and the second evaluation phase is on the same input voltage as the previous evaluation. In this way, it can be made sure that the comparator takes the right decision for both cases.

In the early test, the input was altering every clock cycle and the optimized comparator to correct decision for this case, but not for a constant input, where it wanted to change output every clock cycle for small ΔVin. Figure 3.3.1 shows the signals for the hysteresis tests.

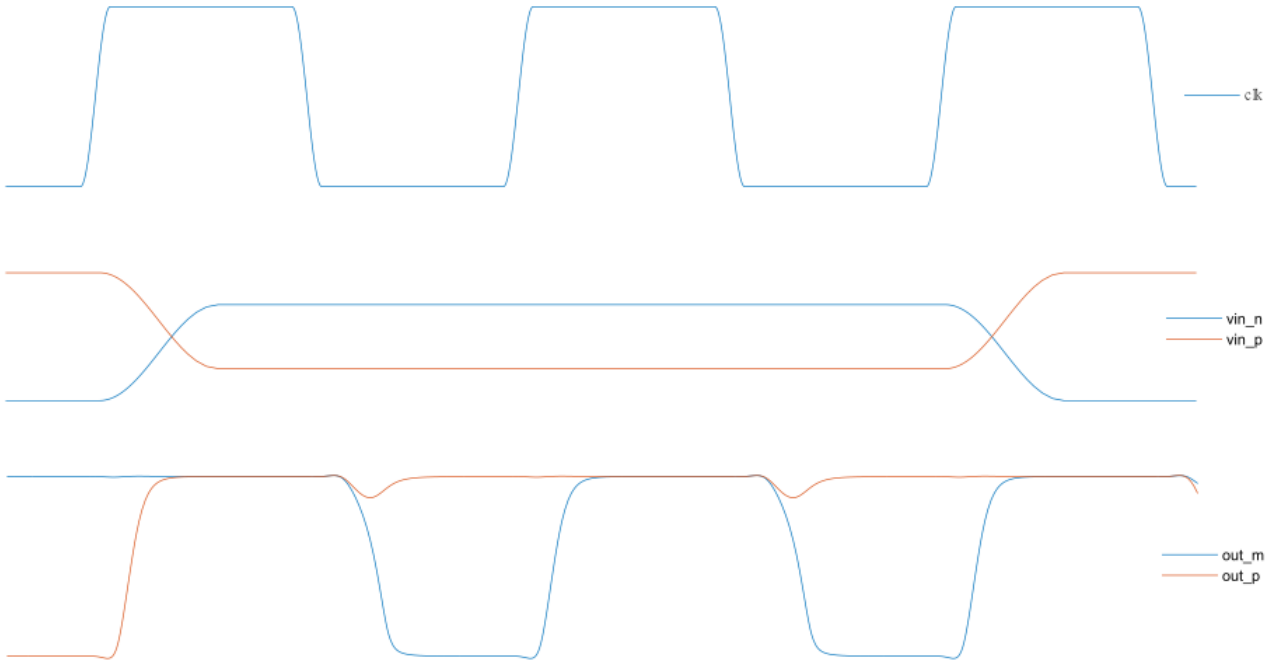


Figure 3.3.1: Inputs and output for hysteresis test.

The input voltages are 6% $\overline{V_{n,in_0}^2}$ and 60% $\overline{V_{n,in_0}^2}$ so that if the comparator fulfills the expressions for the test, the hysteresis will meet the requirement. To determine the exact hysteresis of the comparator, the input is swept at a triangular wave with small steps as figure 3.3.2 and the outputs are checked in the plot as seen in figure 3.3.3. There will be one value for V_{in} where the comparator switches output from 0 to 1, and another value when it switches from 1 back to 0, and from those inputs, the hysteresis can be determined.

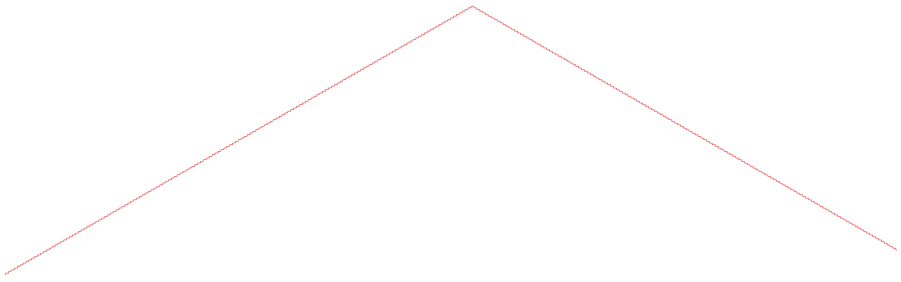


Figure 3.3.2: Plot of a stepping triangular wave used for hysteresis test.

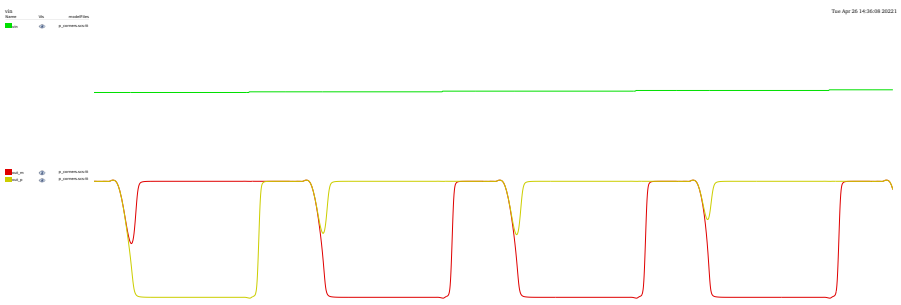


Figure 3.3.3: Output switching from 0 to 1.

3.3.3 Input equivalent load

The input equivalent load is tested separately due to interfering signal levels during the normal running of the comparator. Therefore it is tested during constant reset, with inputs toggling as normal. The load is calculated as described in chapter 2.3.7 where the requirement is $< C_{in}$.

3.3.4 Offset

Since we need to run Monte Carlo simulations to estimate offset levels, there is no test for offset during the optimization phase. The offset simulations are done in a separate testbench, with a VerilogA block as the input to the comparator and the output from the comparator fed back as input to the VerilogA block. To estimate the offset, the circuit is run in a Monte Carlo simulation over a large number of iterations, and from there, the offset can be estimated by Gaussian distribution.

A binary search using VerilogA is performed to increase the simulation speed instead of a stepping signal. The starting point is $V_{in} = 0$, and the output is fed back to the VerilogA-block. Depending of the output from the comparator, the next input will be either $V_{in} = \frac{-V_{search}}{2}$ or $V_{in} = \frac{+V_{search}}{2}$. It will continue to halve the input in the desired direction until the number of steps is reached, and the last used V_{in} is the offset [9].

In this test the search space is set to $-20\,\mathrm{mV}$ to $+20\,\mathrm{mV}$ with 16 steps, giving a resulotion of

$$\frac{40\,\mathrm{mV}}{2^{16}} = 610\,\mathrm{nV} \tag{3.3.3}$$

over 2000 test points per corner for a high accuracy.

3.3.5 Kickback Noise

The simulations for kickback noise are also done in a separate testbench, with a stepping input and a switch, separating the comparator from the voltage source with a capacitor

connected to ground on both inputs. In this test it's checked that the comparator is still taking the correct decision with the presence of the kickback. The input is stepping from -200 mV to 200 mV. The differential kickback noise is measured as

$$V_{max,p} = comp\_in_p - input_p \tag{3.3.4}$$

and

$$V_{max,n} = comp\_in_n - input_n \tag{3.3.5}$$

resulting in

$$V_{kb,diff} = V_{max,p} - V_{max,n} \tag{3.3.6}$$

and the common-mode kickback noise is measured as

$$V_{kb,CM} = \frac{V_{max,p} + V_{max,n}}{2} \tag{3.3.7}$$

3.4 Limits

As discussed in earlier chapters, the offset and kickback noise is one big problem with comparators. It is tough to include the tests for these parameters in the optimization and keep it time-efficient since they need different kinds of testbenches. If those would be possible to include, the requirement for offset and kickback could be set in the beginning. With the current setup, other parameters are known, and these parameters are simulated in the end without much room to optimize.

Chapter 4

Results

In this chapter, the results are presented. The general idea is that the comparator should just meet the requirements and the one with the lowest power consumption is the best one. However, the truth is not so simple, as every design has some weakness even if it meets the requirement.

All the designs were optimized until the optimizing tool in cadence was satisfied. Therefore it should be the best possible solution, passing all the tests over all the process corners with a minimized power consumption.

As can be seen in table 4.0.1, the StrongARM-latch draws by far the lowest amount of power, while triple tail 2 is consuming the most.

Energy/cycle ff corner, [% E_0]	StrongARM-latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
ΔV_{in} @ $\overline{V_{n,in_0}^2}$	23	42.3	43.7	85.3	56.1	49.5

Table 4.0.1: Results for energy/cycle

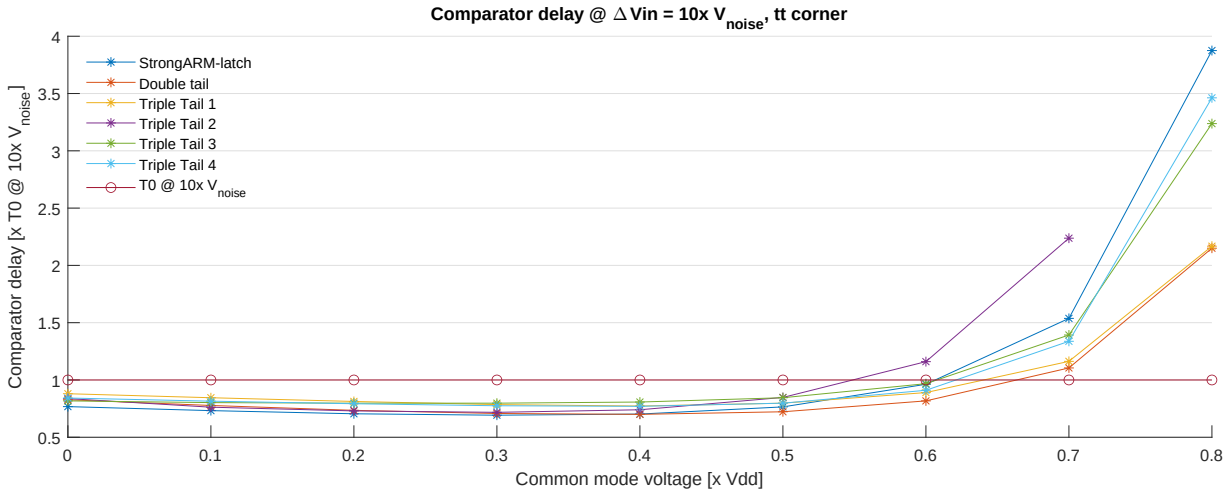
4.1 Common-mode sweep

Since the designs passed the tests during the optimization, it is known that they meet the requirements over all the different corners, but it is also interesting to see how they behave in different operation points.

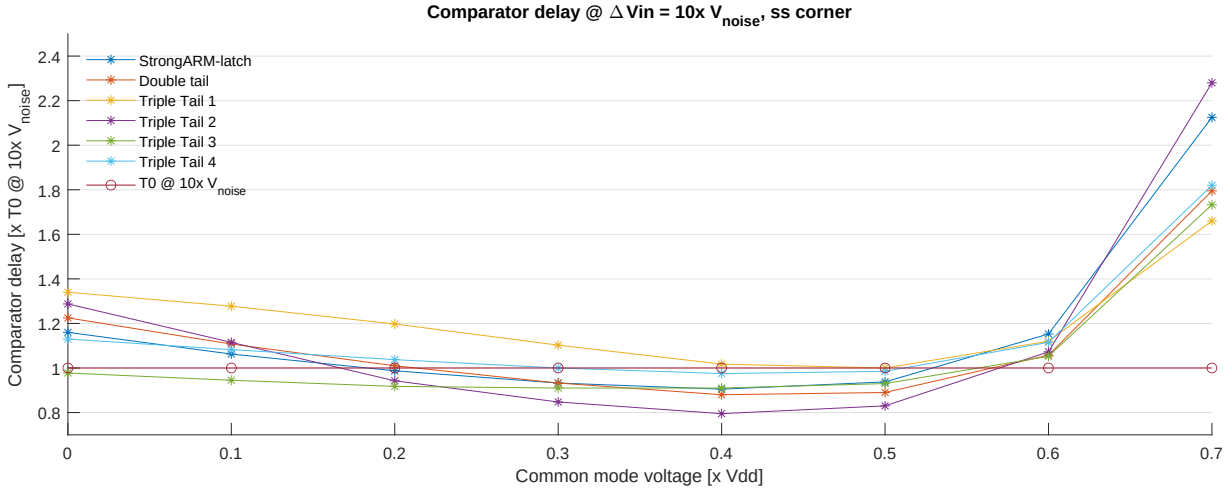
4.1.1 Speed

The Common-mode (CM) voltage is swept to see how the designs behave over different inputs. As is expected, ss corner is the worst corner for speed. Figure 4.1.1b refers to the simulations during optimization, where it is clear that all the architectures pass the requirement at $0.5 \cdot V_{dd}$, where triple tail 1 is the worst performer, only passing the requirement over a small common-mode range. It fails in both $0.4 \cdot V_{dd}$ and $0,6 \cdot V_{dd}$. The best architecture in this test is triple tail 3, passing the requirement from $0 \cdot V_{dd}$ to $0.5 \cdot V_{dd}$. Triple tail 2 is the most sensitive architecture to common-mode voltage.

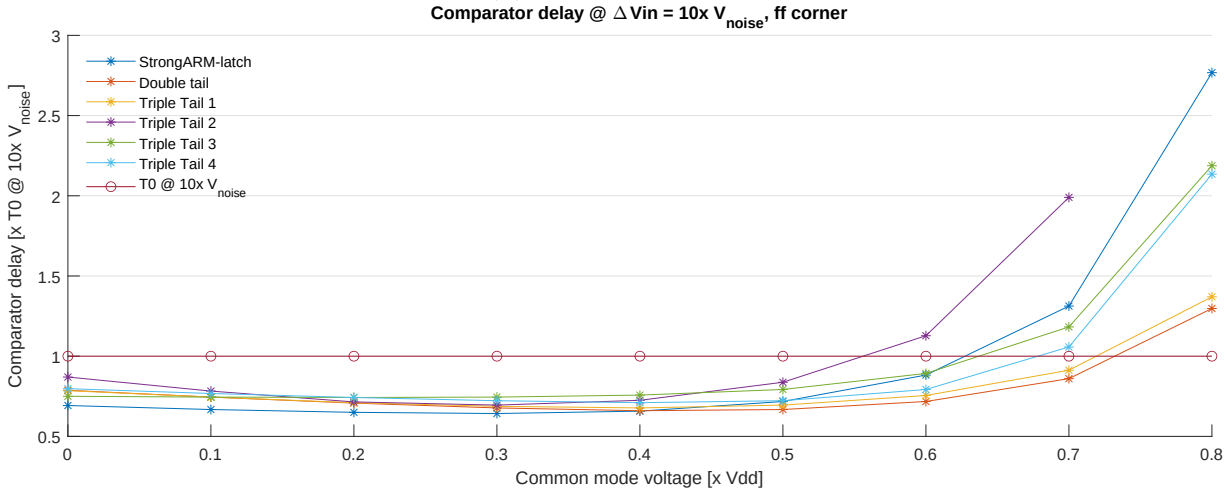
As seen in figure 4.1.1a and 4.1.1c, the architectures perform well within the requirements at both tt and ff corner, where the tendencies are the same as ss corner over the sweep range.



(a) tt corner.



(b) ss corner



(c) ff corner.

Figure 4.1.1: Common-mode sweep over different corners.

4.1.2 Energy

The power consumption is compared over ff corner since it is the corner with the highest power consumption and therefore is most interesting regarding the requirement. As seen in figure 4.1.2, the energy per cycle is stable over common-mode for most of the architectures. Triple tail 2 draws the most power in all the simulations and does not even converge at $0.9 \cdot V_{dd}$. All the other comparators could converge at $0.9 \cdot V_{dd}$, where triple tail 3 did not pass the requirement.

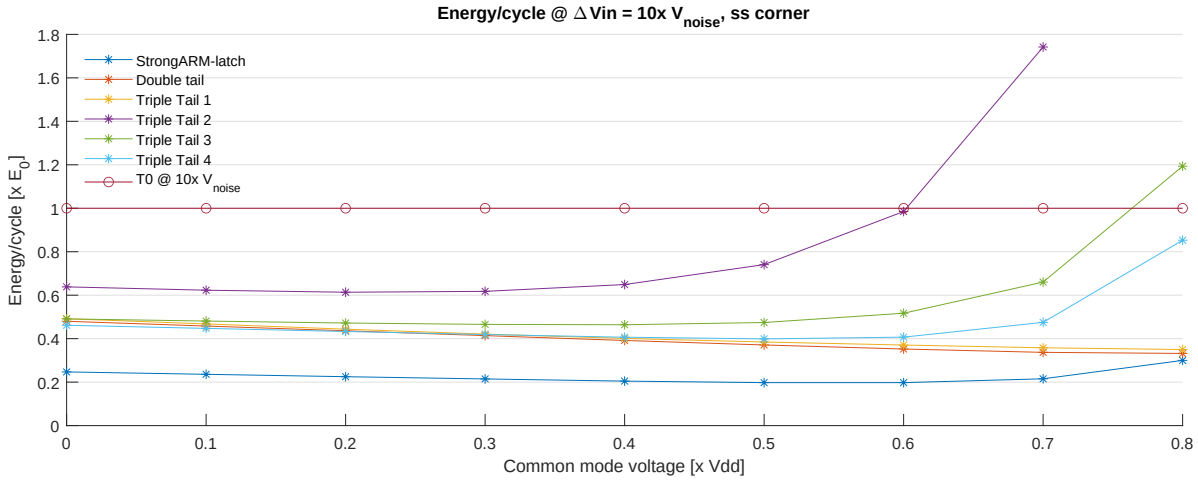


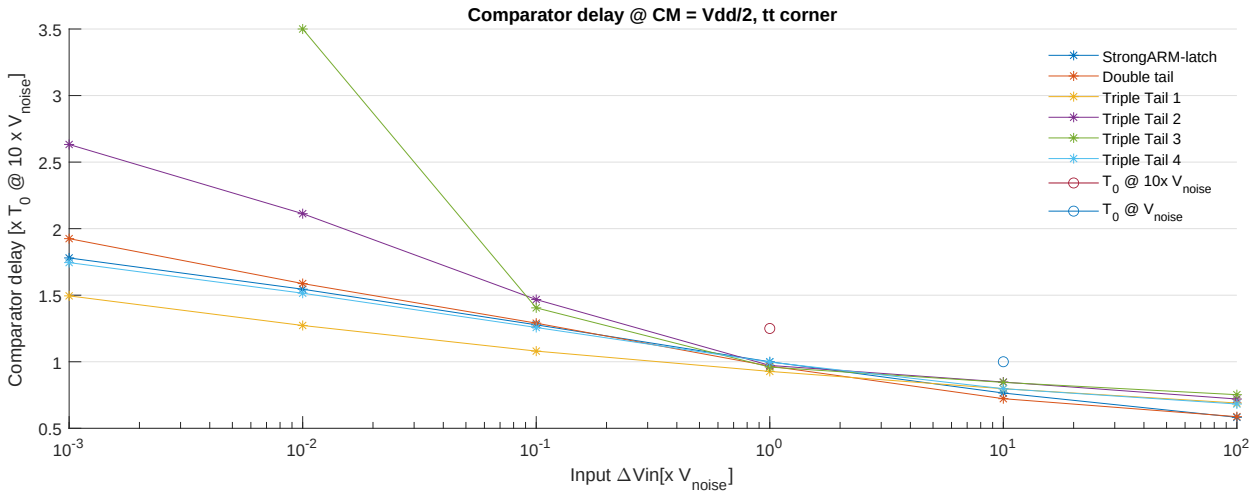
Figure 4.1.2: Common-mode sweep over ff corner.

4.2 Differential sweep

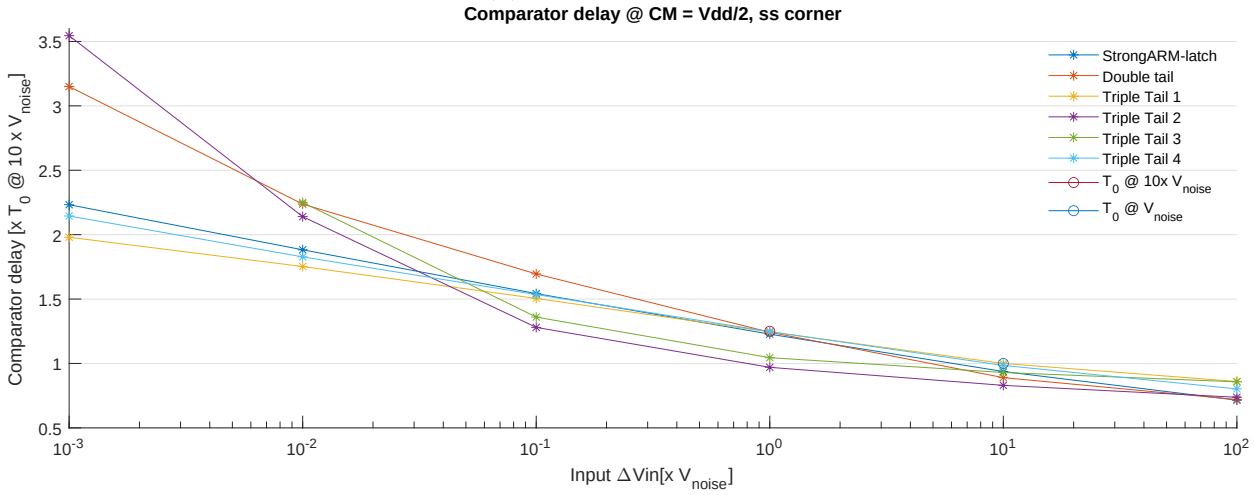
Sweeping over ΔV_{in} will show the performance from very small ΔV_{in} to big ΔV_{in}.

4.2.1 Speed

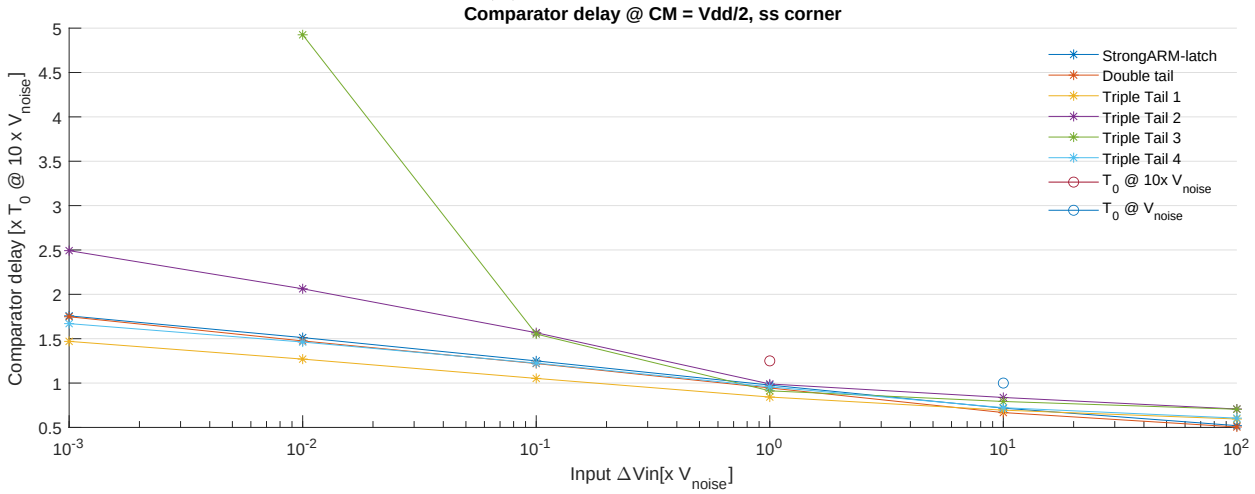
As can be seen in figure 4.2.1, all the comparators is within the requirements at $\Delta V_{in} = \overline{V^2_{n,in_0}}$ and $\Delta V_{in} = 10 \cdot \overline{V^2_{n,in_0}}$. As expected, the decision time increases when ΔV_{in} decreases. The most stable architectures over the test are StrongARM-latch, triple tail 1, and triple tail 4, while triple tail 3 had big trouble converging on small inputs. The Double tail performs well for both ff and tt corner but is significantly slower at ss corner.



(a) tt corner.



(b) ss corner.



(c) ff corner.

Figure 4.2.1: ΔV_{in} sweep over different corners.

To visualize the difference in differential input, figure 4.2.2 shows the delay in the outputs from one of the designs for inputs ranging from $0.001 \cdot \overline{V_{n,in_0}^2}$ to $100 \cdot \overline{V_{n,in_0}^2}$

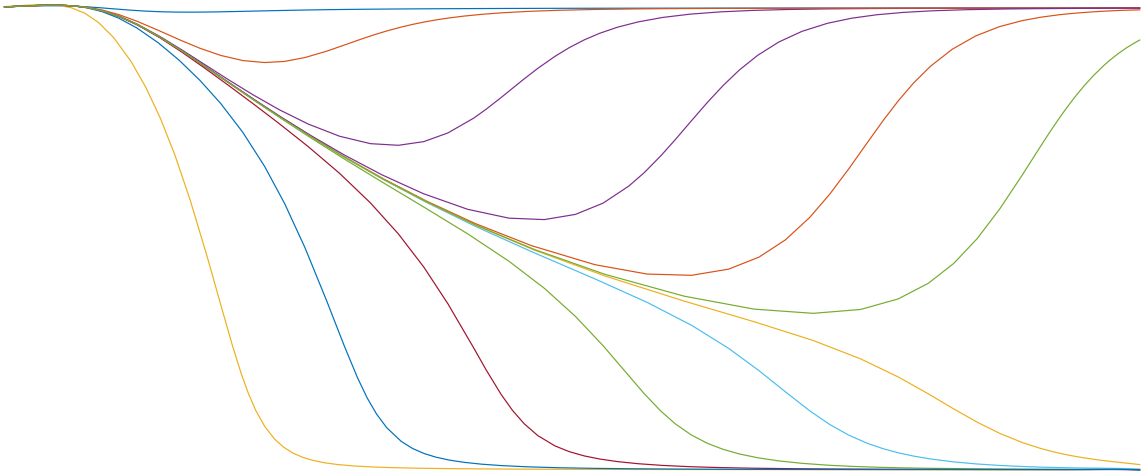


Figure 4.2.2: Outputs for different ΔV_{in}.

4.2.2 Energy

The power consumption is similar to the common-mode sweep,seen in figure 4.2.3, where triple tail 2 draws more power than the rest. Triple tail 3 performs similar to the other architectures in the common-mode sweep, but in the ΔV_{in} sweep, it is consuming more power than the majority. The triple tail 4 has the most stable power consumption over the whole input range, while the StrongARM-latch, double tail, and triple tail 1 have a similar slope.

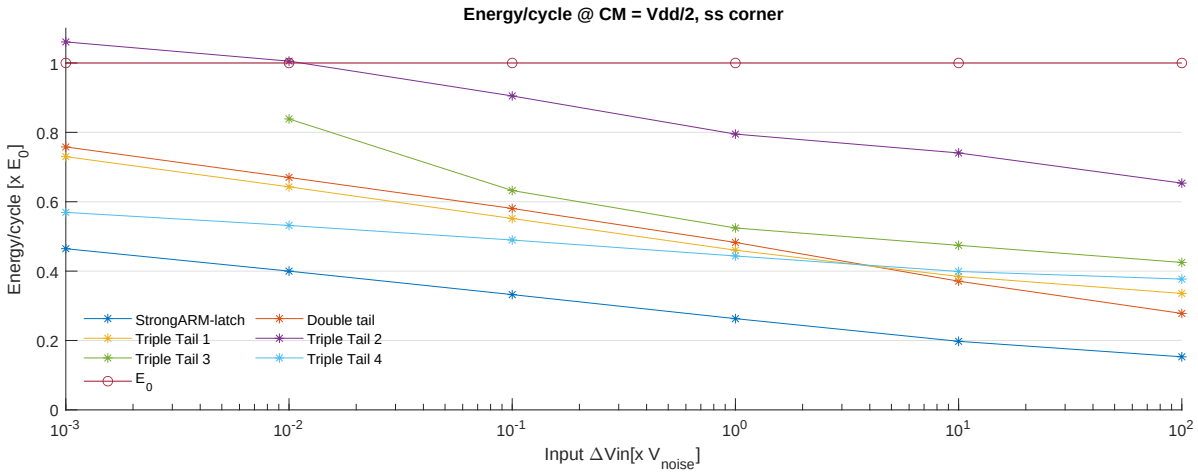


Figure 4.2.3: ΔV_{in} sweep over ff corner.

4.3 Input referred noise

As can be seen in figure 4.3.1, all the designs passes the requirement with a common-mode voltage at $0.5 \cdot V_{dd}$. The lower the common-mode voltage, the more noise there is, with triple tail 1 and triple tail 3 the most sensitive designs, while the double tail and triple tail 3 are the least sensitive.

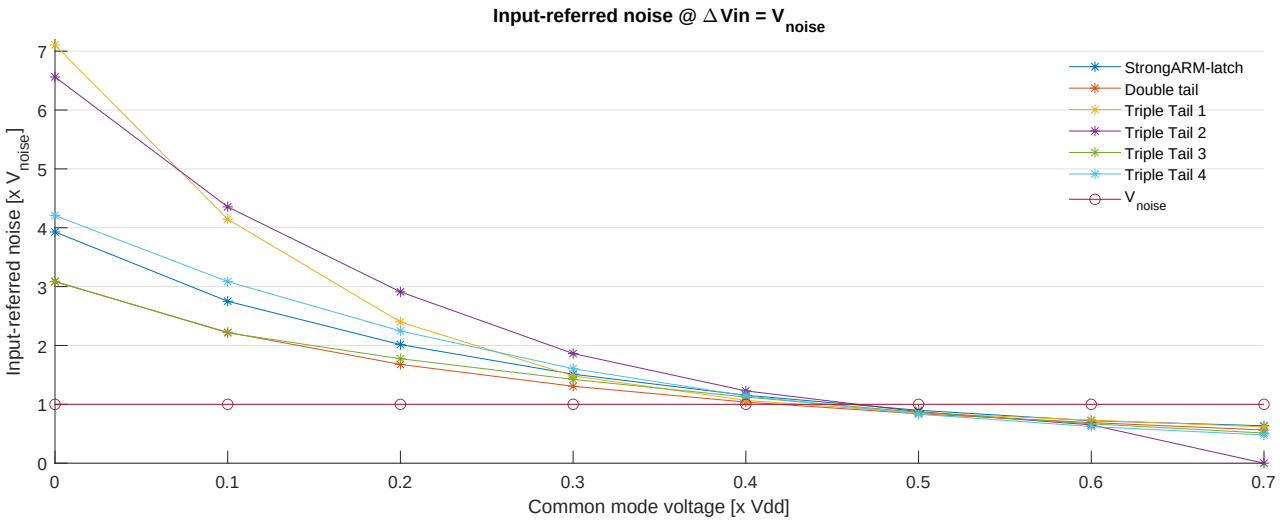


Figure 4.3.1: Common-mode sweep over tt corner.

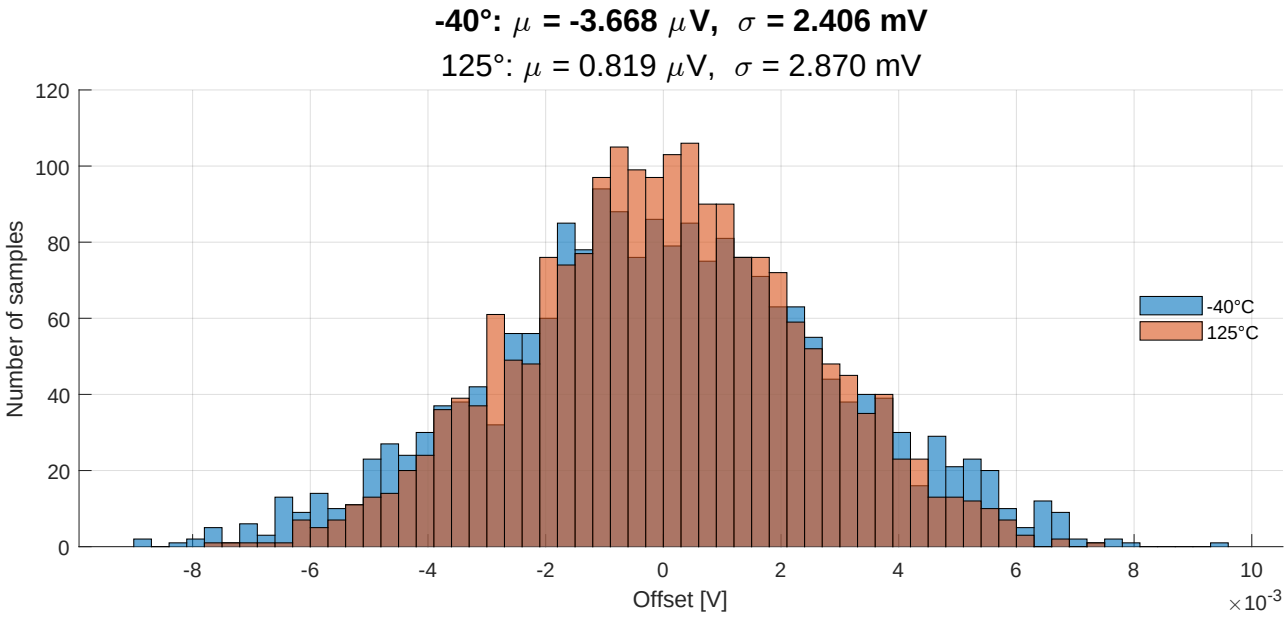
To verify that the results from PSS simulations align with the results from transient simulations, long simulations over 8000 points per design are made. Table 4.3.1 shows the results from the different simulations. As expected, the PSS results for the worst corner are just within the requirement, and the results from the transient simulations are similar but somewhat lower in all cases, indicating that the PSS simulations are trustworthy.

Input-referred noise [% $\overline{V_{n,in_0}^2}$]	StrongARM-latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
PSS tt corner	89	84	86	87	86	82
Transient tt corner	85	79	85	83	87	76
PSS worst corner	99	100	100	100	95	97
Transient worst corner	95	93	97	94	94	89

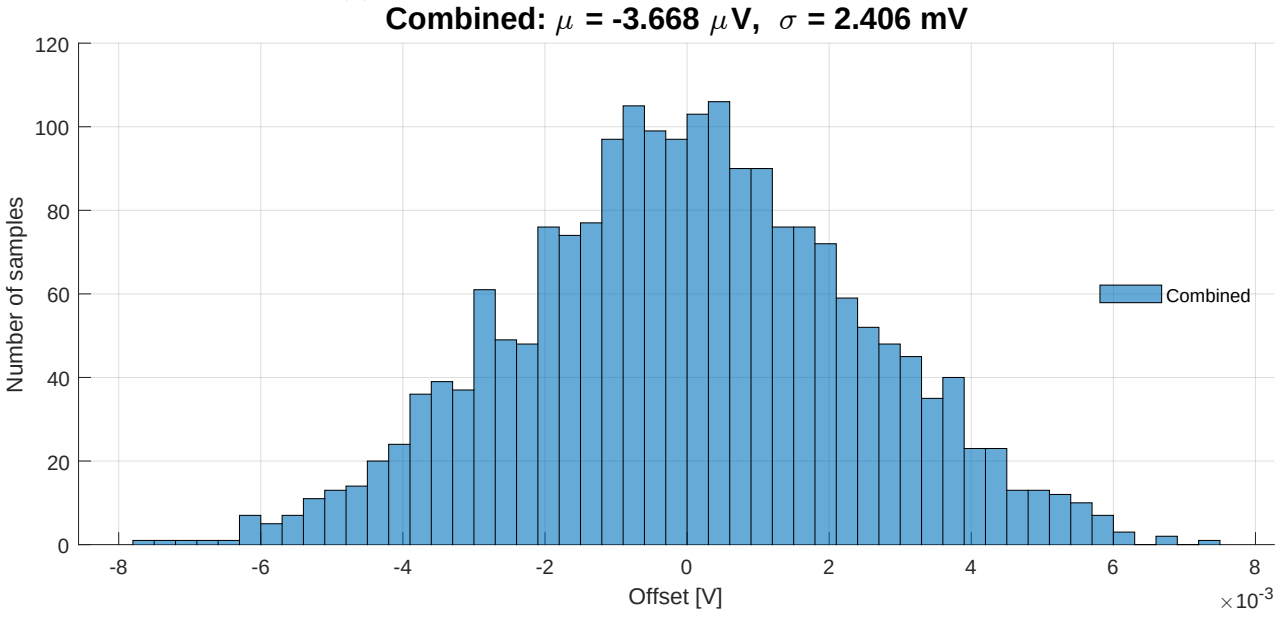
Table 4.3.1: Results from input-referred noise simulations.

4.4 Offset

The offset is tested with Monte Carlo, both at -40°and +125°, over 2000 samples per corner, where the spread can be seen in figure 4.4.1. The results is also presented in table 4.4.1 where the standard deviation, σ , refers to the offset.



(a) StrongARM-latch, -40°and 125°.



(b) StrongARM-latch, combined.

Figure 4.4.1: Histogram over Monte Carlo simulations.

Offset	StrongARM-latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
-40, mean $[\mu V]$	-3.67	-6.01	3.76	-7.99	-7.55	3.9
125, mean $[\mu V]$	0.82	-6.21	4.47	14.27	26.07	4.12
Combined, mean $[\mu V]$	-1.42	-6.11	4.11	3.13	9.26	4.01
-40°, σ $[mV]$	2.4	2.78	2.5	3.28	2.32	2.31
125°, σ $[mV]$	2.87	3.02	2.73	3.62	2.69	2.65
Combined, σ $[mV]$	**2.64**	**2.91**	**2.62**	**3.45**	**2.51**	**2.48**

Table 4.4.1: Results from Monte Carlo simulations for offset.

There is small variations in offset between the different architectures, where triple tail 2 having a sligthly larger offset then the rest, and the triple tail 4 the lowest. It is also interesting that all the comparators shows a bigger offset at 125° than at -40°.

4.5 Kickback noise

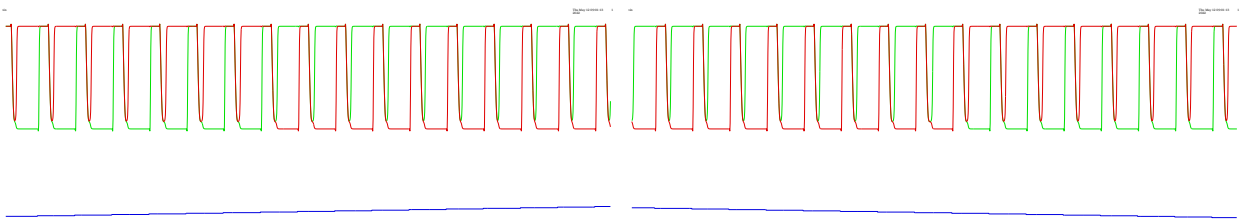
This test was done with two cases with different source capacitances, $3.75 \cdot C_{in}$ and $25 \cdot C_{in}$. From table 4.5.1, it is clear that triple tail 2 provides the lowest amount of kickback noise, except differential kickback noise at $3.75 \cdot C_{in}$. StrongARM-latch has been a strong performer through all the tests, it clearly shows its weakness in differential kickback noise.

Kickback noise $[x\,\overline{V^2_{n,in_0}}]$	StrongARM-latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
CM @ 3.75 $\cdot C_{in}$	52.7	45.8	44.9	23.2	52.74	57.5
CM @ 25 $\cdot C_{in}$	10.4	9.1	9.1	4.5	11.3	11.8
ΔV@ 3.75 $\cdot C_{in}$	6.3	1.3	1.2	2.7	5.4	1.5
ΔV @ 25 $\cdot C_{in}$	1.7	0.5	0.6	0.3	0.4	0.7

Table 4.5.1: Results from kickback noise simulations.

4.6 Hysteresis

The hysteresis is measured as the difference between when the output start to switch from 0 to 1 and again when it start to switch from 1 to 0, as seen in figure 4.6.1.



(a) Output switch from 0 to 1. (b) Output switch from 1 to 0.

Figure 4.6.1: Plots over hysteresis test.

The hysteresis is tested over the typical corner in two cases, with an input capacitance of 3.75 $\cdot$ C$_{in}$ and 25 $\cdot$ C$_{in}$ and the results is seen in table 4.6.1. The requirement for the hysteresis is V_h, which is 6% $\overline{V^2_{n,in_0}}$ and the resolution for the simulation is close to 0.5% V_h. All the designs passes the requirement in this test.

Hysteresis [% V_h]	StrongARM-latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
3.75 $\cdot$ C$_{in}$	0	4.2	0.5	0.5	35.6	0.5
25 $\cdot$ C$_{in}$	0	3.69	0	0.4	30.1	0.5

Table 4.6.1: Results from hysteresis simulations.

It can be seen that triple tail 3 has the largest hysteresis, the double tail has a small hysteresis, and the others have zero or close to zero hysteresis.

4.7 Summary

Table 4.7.1 shows a summary of results in the tests, with a common-mode at 0.5 $\cdot$ V_{dd} where a lower value is desirable across the table.

Parameter	StrongARM -latch[8]	DT[14]	TT1 [15]	TT2[6]	TT3 [2]	TT4
Energy/cycle @ $\overline{V_{n,in_0}^2}$ [% E_0]	23%	42.3%	43.7%	85.3%	56.1%	49.5%
Input equivalent load [% C_{in}]	43.5%	49.5%	46%	30.5%	74.5%	63.5%
Hysteresis @ $3.75 \cdot C_{in}$ [% V_h]	0	4.2	0.5	0.5	35.6	0.5
Hysteresis @ $25 \cdot C_{in}$ [% V_h]	0	3.69	0	0.4	30.1	0.5
Offset [mV]	2.64	2.91	2.62	3.45	2.51	2.48
Kickback noise CM @ $3.75 \cdot C_{in}$ [x $\overline{V_{n,in_0}^2}$]	52.7	45.8	44.9	23.2	52.74	57.5
Kickback noise CM @ $25 \cdot C_{in}$ [x $\overline{V_{n,in_0}^2}$]	10.4	9.1	9.1	4.5	11.3	11.8
Kickback noise ΔV @ $3.75 \cdot C_{in}$ [x $\overline{V_{n,in_0}^2}$]	6.3	1.3	1.2	2.7	5.4	1.5
Kickback noise ΔV @ $25 \cdot C_{in}$ [x $\overline{V_{n,in_0}^2}$]	1.7	0.5	0.6	0.3	0.4	0.7

Table 4.7.1: Summary of from simulations.

Chapter 5

Discussion

This chapter will discuss the results and methodology of the thesis work.

5.1 Results

The aim for the comparators was to meet the requirements stated in 1.4.1 and let the optimizer run until the best possible solution was found. This is, of course, not the whole truth since there are other parameters to consider and different operating points which is not suitable to include in the optimizer.

The StrongARM-latch seemed to be the clear winner from the initial tests, passing all the requirements and by far the smallest power consumption. However, with the additional test, which was not included in the optimizer, it had large kickback noise, and if this is a problem in the ADC design, the StrongARM-latch might not be a suitable choice. There are techniques to reduce the kickback noise in comparators, but none of those is used in this thesis.

The double tail, triple tail 1, and 4 had similar performance in the tests, always somewhere in the middle and never in the extremes. In contrast, triple tail 2 and 3 showed poor performance at some operation points, for example, common-mode sweep.

As expected, all the designs have one or multiple weak spots. For example, the triple tail 4 is the worst in kickback noise at common-mode. The double tail is sensitive to common-mode voltage at speed tests over ss corner. The triple tail 1 is sensitive to common-mode voltage when sweeping input-referred noise. However, it is clear that triple tail 2 and 3 are the worst designs in most tests.

Choosing one design over another would come down to which parameters are the most important in the overall design, and either choose the comparator which is best suited or use techniques to limit these problems.

5.2 Method

The use of an optimizer speeded up the work significantly, providing a good solution in such time that it would be impossible to do manually. It was crucial to figure out the tests and discover eventual problems early to avoid false results. One example of false results is the hysteresis test. When it was simulated with the inputs switching with the clock, it could not be seen if the comparator would hold the output for the same input or optimize it into switch output every time with a small input. The solution for this was to double the period for the inputs, keeping the same value for two clock cycles, and measure the time four times instead of two to make sure it kept the output when it should and switched when it should.

As already mentioned, it was not possible to include all the tests in the optimizer, which led to some designs performing poorly in the other tests. If these could be included in the optimizer, those designs would probably be better overall, with a possibly slightly higher power consumption as a result. It could be possible to include different common-mode voltages in the tests, for example, but it is always a balance between proper tests and time.

While the optimizer decreased the design time significantly, it might not be suitable for every design. A comparator is a relatively small design with few transistors, allowing the optimizer to work through thousands of points in a relatively small amount of time. With larger designs, the time will increase significantly, doing the groundwork with calculations and previous tests more critical. There was a significant difference in time in finding a suitable solution by only comparing the StrongARM-latch, containing 17 transistors including inverters, and a triple tail, containing up to 28 transistors including inverters.

5.3 Different VT-transistors

Both the StrongARM-latch and triple tail 4 were implemented with transistors with different threshold voltages, VT_1 and VT_2, where VT_1 has the lower threshold voltage of the two. In one case, all transistors were VT_1, and in the other case, the input transistors were VT_1 and the rest VT_2. The StrongARM-latch worked perfectly fine in the first case but could not meet the requirements for speed in the second case. With the VT_2 transistors, it could not reach the speed requirement no matter how freely the optimizer could run, with a large increase in power consumption as well. In the final optimized solution with VT_2 transistors, the comparator delay was 10% over the requirement, while the power consumption was about seven times larger than the optimized solution with VT_1 transistors.

The triple tail 4 did show the same tendencies with VT_2 transistors, being slow and power-hungry. This design was never optimized until the tool was satisfied, so exact values are not known, but they can be expected to be similar to the Strong-ARM latch.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The main purpose of this work was to compare different state-of-the-art comparator. Knowing that the StrongARM-latch would be the best performing comparator, there would have been more time looking for similar architectures rather than triple tail comparators, which performed poorly compared to the StrongARM-latch in most tests. During the research some different StrongARM-latch with kickback noise reduction techniques were found, but they were not interesting enough in that phase of the thesis work. The StrongARM-latch was not planned to be included in the first place, but the purpose was to include it as a reference point to an older architecture and see how well it holds up against newer architectures.

There are different trade-offs and weaknesses with different technologies, and the majority of the available work is done in 35-280 nm. Using a different technology might change the weaknesses of the designs and provide surprising results, as it can be called in this thesis.

6.2 Future Work

There are a few things that would be interesting to explore following this work:

- Implement as layout for further testing.
- Investigate techniques to reduce offset and kickback noise.

The extracted schematic from the layout will probably be worse than the test results in this report, but some architectures might differ more than others, which would be interesting and need to be considered before an eventual tape-out. Considering the offset and kickback noise levels, it could be beneficial to reduce these, even if the levels from the simulations can be considered as low.

Bibliography

[1] Shih-Hsing Wang and Chung-Chih Hung. "A 0.3V 10b 3MS/s SAR ADC With Comparator Calibration and Kickback Noise Reduction for Biomedical Applications". In: *IEEE Transactions on Biomedical Circuits and Systems* 14.3 (2020), pp. 558–569. DOI: 10.1109/TBCAS.2020.2982912.

[2] Athanasios T. Ramkaj, Michiel S. J. Steyaert, and Filip Tavernier. "A 13.5-Gb/s 5-mV-Sensitivity 26.8-ps-CLK–OUT Delay Triple-Latch Feedforward Dynamic Comparator in 28-nm CMOS". In: *ESSCIRC 2019 - IEEE 45th European Solid State Circuits Conference (ESSCIRC)*. 2019, pp. 167–170. DOI: 10.1109/ESSCIRC.2019.8902790.

[3] Behzad Razavi. *Principles of Data Conversion System Design*. IEEE Press, 1995. ISBN: 9780470545638.

[4] B. Razavi and B.A. Wooley. "Design techniques for high-speed, high-resolution comparators". In: *IEEE Journal of Solid-State Circuits* 27.12 (1992), pp. 1916–1926. DOI: 10.1109/4.173122.

[5] Behzad Razavi. "The Design of a Comparator [The Analog Mind]". In: *IEEE Solid-State Circuits Magazine* 12 (Nov. 2021), pp. 8–14. DOI: 10.1109/MSSC.2020.3021865.

[6] Athanasios T. Ramkaj et al. "A 1.25-GS/s 7-b SAR ADC With 36.4-dB SNDR at 5 GHz Using Switch-Bootstrapping, USPC DAC and Triple-Tail Comparator in 28-nm CMOS". In: *IEEE Journal of Solid-State Circuits* 53.7 (2018), pp. 1889–1901. DOI: 10.1109/JSSC.2018.2822823.

[7] Behzad Razavi. *Design of analog CMOS integrated circuits, second edition*. McGraw-Hill Education, 2017. ISBN: 9780072524932.

[8] Behzad Razavi. "The StrongARM Latch [A Circuit for All Seasons]". In: *IEEE Solid-State Circuits Magazine* 7.2 (2015), pp. 12–17. DOI: 10.1109/MSSC.2015.2418155.

[9] João Silva et al. "Methods for Fast Characterization of Noise and Offset in Dynamic Comparators". In: *2021 19th IEEE International New Circuits and Systems Conference (NEWCAS)*. 2021, pp. 1–4. DOI: 10.1109/NEWCAS50681.2021.9462744.

[10] Mansoure Yousefirad and Mohammad Yavari. "Kick-back Noise Reduction and Offset Cancellation Technique for Dynamic Latch Comparator". In: *2021 29th Iranian Conference on Electrical Engineering (ICEE)*. 2021, pp. 149–153. DOI: 10.1109/ICEE52715.2021.9544395.

[11] Aashita Raj et al. “Trade-off Characteristics of Hysteresis Comparator used in Noisy Systems”. In: *2019 Devices for Integrated Circuit (DevIC)*. 2019, pp. 413–417. DOI: 10.1109/DEVIC.2019.8783668.

[12] Behzad Razavi. *Design of analog CMOS integrated circuits, second edition*. McGraw-Hill Education, 2017. ISBN: 9780072524932.

[13] Cadence. *Circuit Optimization*. Aug. 2019. URL: https://support.cadence.com/apex/ArticleAttachmentPortal?id=a1O0V000009Mp5fUAC&pageName=ArticleContent (visited on 05/13/2021).

[14] Guan-Ying Huang et al. “A 10b 200MS/s 0.82mW SAR ADC in 40nm CMOS”. In: *2013 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. 2013, pp. 289–292. DOI: 10.1109/ASSCC.2013.6691039.

[15] Jingqi Wang, Fan Ye, and Junyan Ren. “A Three-Stage Comparator with High Speed and Low Power”. In: *2021 IEEE 14th International Conference on ASIC (ASICON)*. 2021, pp. 1–4. DOI: 10.1109/ASICON52560.2021.9620370.