Contents

Comparator and DDNR* **3**

5.1 Comparator Overview 3

5.2 Comparator Topologies 5

5.2.1 Straight-forward amplification 5

5.2.2 Amplification with positive feedback path 6

5.3 Pre-amplification 7

5.4 Latch regenerative operation 9

5.5 Metastability and Bit error rate 12

5.6 Accuracy and comparator offset voltage 14

5.7 Noise in a comparator 19

5.8 Kickback noise 21

5.9 Hysteresis 23

5.10 Initial design step 26

5.11 Scouring comparator architectures for the best choice 29

5.11.1 FIA 35

5.11.2 FLL initial design 43

5.12 Noise analysis and separate testbenches 50

5.12.1 PSS+ Pnoise analysis + PXF 50

5.12.2 Thermal noise analysis 55

5.13 Offset analysis 59

5.13.1 Background offset calibration 69

5.13.2 Foreground offset calibration 72

5.14 Power analysis 75

5.14 Asynchronous loop design 77

5.15 Finalized comparator results 82

5.16 Layout 85
5.17 DDNR 95
5.18 Conclusion 108

Chapter 5

Comparator and DDNR*

In the previous chapters we described the major concept of SAR ADCs, delved into the general structure of SAR ADCs with its three major components as in SAR LOGIC, DAC switching and the Comparator, and described the whole switching algorithm of the proposed design with its inverted merged switching algorithm and the allocated LOGIC design approach to accommodate such pristine switching algorithm.

In this chapter we will go through the full design of the comparator along with its associated devices including the Asynchronous clock generation loop and the comparator enable logic which plays a vital role in conserving large amounts of energy, but first before we indulge ourselves with the design process, let's understand first what the comparator is and what its major function really is.

5.1 Comparator Overview

The comparator is the circuit location where nature turns into bits, meaning that, this is where the digital decision is made, the comparator does one of two things, it either detects the polarity of a differential analog input signal and generates a digital output either logic HIGH or logic Low

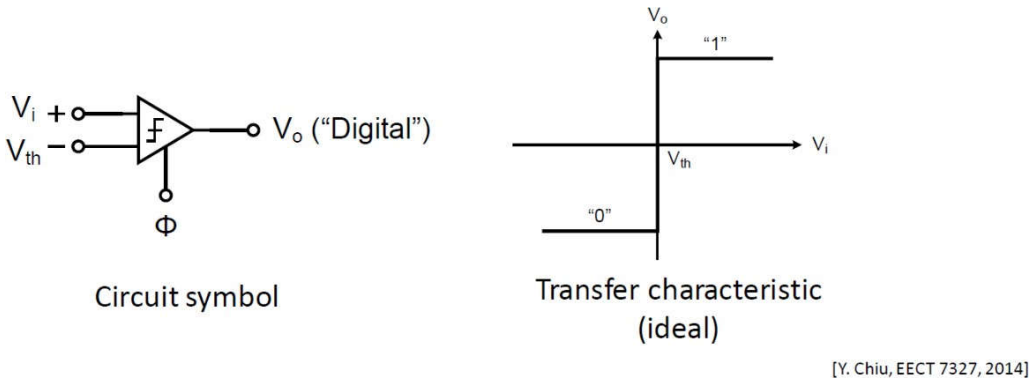


Figure 5-1 An overview of the comparator concept of operation, where it acts as a threshold crossing detector on the right, and a polarity detector on the left.

accordingly, or it compares a SE input to a threshold reference voltage, which makes it a threshold-crossing detector as you can observe in Figure 5-1.

Every analog-to-digital converter contains at least one comparator. Yet, there is little uniformity on comparator topologies. Many aspects need consideration when designing a comparator. These aspects may vary for every different application, for specific classes of signals, technology, etc. No universal "one design fits all converters" comparator exists. There are as many comparator circuit variants as there are analog-to-digital converter designers. Still some general remarks on the design of the comparator can be made. Accuracy and speed are the global design parameters that have to be balanced versus power consumption.

The fundamental task of a comparator is to amplify a big or small difference between its input terminals into a digital decision. In other words, to extract the sign of the differential input signal. A number of requirements can be specified for a comparator:

• A large amplification is imperative as miliVolt, or microVolt signal differences are translated in digital levels.

• A wide bandwidth for operating on high-frequency signals.

• Accuracy of various forms is required. In practical terms this means a low input offset, a low noise-figure, for 1/f noise as well as for thermal noise. Clocked comparators should not add uncertainty to the decision moment: a low timing jitter is required.

• A low power consumption, especially in analog-to-digital converters employing a lot of comparators.

• A wide input common mode voltage range, with a high common mode rejection, to overcome certain effects like common mode noise coming from events like cross talks between the wires.

• No metastability, any decision is better than no decision. No decision means to be stuck in between the Logic High and Logic Low, which would cause significant errors if the comparator preceded a finite state machine, aka. a conceptual model that describes systems with a finite number of states and transitions to represent their behavior and logic.

• The previous comparator decision should not affect the following, no memory effect, also referred to as: inter-symbol interference or Hysteresis window.

5.2 Comparator Topologies

There are three basic Comparator Topologies that we think of when we want to design a comparator,

5.2.1 Straight-forward amplification

The simple and relatively slow option is the straight-forward amplification option which is a free-running version of the comparator that doesn't require a triggering clock, as we can see in Figure 5-2.

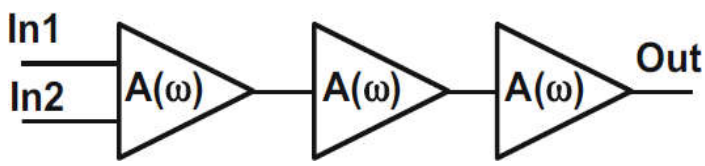


Figure 5-2 Shows a cascade of amplification stages [M.Pelgrom, 2017]

The straight-forward limiting amplifier consists of a cascade of amplification stages to obtain as much gain as possible. For a given current consumption, the unity-gain bandwidth (UGBW) per stage determines the overall speed of this chain of amplifiers. In amplifier theory the UGBW is that frequency where the gain equals unity.

A major flaw in this design concerning the speed of this comparator is the recovery from the previous decision. Now let's discuss the problem more thoroughly, when a large signal input is given to the comparator, it can drive the last stages of the amplifier to saturation, which means that the output of the amplifier reaches its maximum or minimum limit and cannot increase or decrease further despite an increase in the input signal.

The effect of saturation is that certain elements within the amplifier such as MOS transistors tend to lose their inversion charge, which is critical for proper functioning of MOS transistors, as it is related to the formation of the conductive channel that allows the current to flow.

After saturation, the transistors need to recover their inversion charge to return to the normal operation, this process takes time which introduces a delay in the next comparison cycle of the comparator, this is undesirable as it slows down the overall speed of the comparator impacting its performance in applications requiring rapid successive comparisons.

One common method of solving this is the use of limiters like diodes to mitigate the issue of saturation where they can limit the voltage swing, preventing the amplifier from reaching saturation and thus the inversion charge in the MOS transistors is preserved, which in turn reduces the recovery time between the comparisons allowing the comparator to operate at higher speeds.

Another concern is that if the comparator in this case is operating in open loop, it will require no compensation, it will provide high gain but at the cost of the bandwidth, so we will need to operate it in feedback closed loop formation to ensure high stability by pushing Gx (Frequency at unity gain) inwards, but this will decrease the gain and thus the efficiency of the comparator. Let's say we use a series of low gain high bandwidth Op-amps would that be a good thing ? It might not be a good thing, because if the assembly will give strong one this means that there will be large voltage transition which might cause slewing, and this will limit the speed with a non-linear settling behavior. But using a chain of inverters might work, as it can give decent gain with relatively good trans-conductance, and it will not be limited by slewing because they are push-pull circuits, and this will give decent speed.

5.2.2 Amplification with positive feedback path

There are two types of this topology, one that uses a small amplifier as a positive feedback element. As this amplifier is weaker than the feed-forward path it will only marginally contribute to the amplification. What it can do is add a threshold in the decision process. This so-called hysteresis is discussed in section 5.9 . Yet, in some circuit configurations this topology may arise, be-it unintentionally. In the last topology, the feedback path of the amplifier is of comparable strength to the forward path. The idea of this regenerative stage or latch is that the already build up difference in the forward stages feeds the positive feedback in order to reach a fast decision. This mode of amplification must be reset by a clock phase that clears the data after the decision, disables the feed forward, and allows preamplification. The block diagram of both can be shown in Figure 5-3.

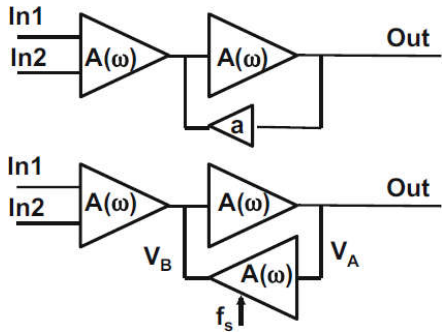


Figure 5-3 shows two positive feedback topologies used in comparators. [M.Pelgrom,2017]

This last topology of using clocked positive feedback will be the core of our design, presented in Figure 5-4 is one of the simplest regenerative comparator circuits. The operation has some sort of built-in S/H operation where it samples the input on the preamp outputs and then holds it in the Latch, it can be viewed in the same way as a sample-and-hold circuit. Or the device itself is activated by a clock at a certain instant, like we are taking a snapshot at a certain instant, inherent sample and hold operation.

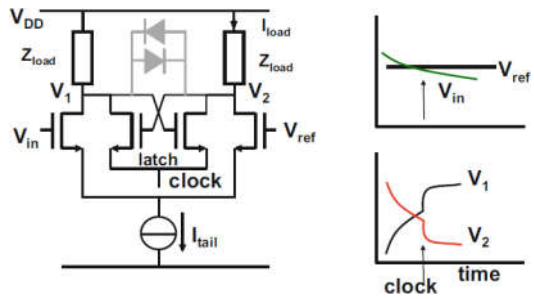


Figure 5-4 A simple regenerative comparator circuit [M.Pelgrom,2017]

5.3 Pre-amplification

Before the regenerative section of the comparator is activated, the signal is amplified. This amplification will reduce the influence of mismatch in the regenerative section and create a form of isolation between the regeneration process and the input sources. Most comparators use an input differential pair. Here the difference is formed between the input value and the reference value. The differential pair allows to create some tolerance for the common mode level of the input signal. The differential current is applied to a load. After some amplification a positive feedback latch activated by a clock pulse will turn the latch on and amplify the small input voltage difference to a large signal. A basic example circuit is shown in Fig. 5-4. The gain of the comparator is determined by the transconductance of the input differential pair and the load: $gm_{,in} * Z_{load}$

This gain must be sufficient to suppress mismatches, noise, etc. in the regenerative part of the circuit. On the other hand, there is no reason to boost the DC-gain to very high values usually preamp gain $\leq$ 10. Too much gain will cause saturation, slewing and unnecessary kick-back. The achievable speed performance is limited by the unity-gain bandwidth, which in turn is determined by the parasitic capacitive load C_{load} on the internal nodes and the available current drive. The bandwidth of the comparator must be analyzed both in small-signal mode and in large-signal mode.

In small-signal mode the input transconductance and the capacitive load of the nodes $V1$ and $V2$ determine the bandwidth:

$$\omega = {}^{gm,in}/_{C,\,load} \tag{5.1}$$

A large input transconductance is beneficial for a large small-signal bandwidth. Choosing a wide input transistor also helps in reducing the input referred mismatch but creates a larger capacitive load for the input terminal and the reference. Also, the parasitic coupling between the drain voltages and the input terminals will become stronger increasing the kick-back.

In order to achieve the overall performance also the large-signal bandwidth of the circuit must be considered. Two main issues are crucial for the large-signal bandwidth Slew rate and saturation.

First, slew rate, we can say that the fastest change of the signal that is amplified without distortion is limited by the slew rate :

$$\begin{aligned} \text{slew-rate} = \frac{dV}{dt} &= \frac{I_{tail}}{C_{load}} \\ C_{load}\frac{dV_1(t)}{dt} &\leq I_{tail} \\ 2\pi f_{in} V_{1,max} C_{load} &\leq I_{tail} \end{aligned} \tag{5.2}$$

The current is I_{tail} because we assume that full steering takes place due to the large input difference at the input terminals, as only with high differential inputs comes the slewing case, $V1,max$ is the amplitude of an equivalent sine wave. If the charging current during the transient of the signal exceeds the tail current, the charging of the capacitors in the circuit will be limited and distortion can follow.

The second large-signal effect in this comparator is saturation. The large signals that drive a comparator will force the input transistors and the internal components in a saturated “on” or “off” regime. Internal nodes will be (dis-)charged to the power supply levels. Saturation of the input transistors creates significant currents in the gate connection . In order to revitalize the comparator all saturated components will have to be brought back into their linear operation regimes. This process requires current, and the signal processing will experience a delay time. This will result in signal distortion. To prevent saturation effects, it is wise to limit the maximum swing on the internal nodes to 200–300mV. In the comparator schematics of Fig. 5.4 two diodes represent the signal swing limiting on the internal nodes from saturation. The diodes will make sure that the largest voltage difference between the output nodes is the diode ON voltage which is about 0.7v for silicon diodes, this will protect the system from the unwanted saturation effect. In normal CMOS technology these diodes are not available. Various other circuit techniques can be used to reduce the internal voltage swings (non-linear loads, cascode stages, etc.).

Figure 5.5 depicts a capacitive pre-amplifier. During a pre-charge phase, the capacitors are short-circuited and both internal nodes equal VDD .When the switches open the current controlled by the differential pair will discharge the capacitors and the differential current causes divergent voltages. The gain due to the integration action is increasing with time. This circuit will not compare fluctuating input signals but operates only on stable signals from, e.g., a T&H circuit as in a successive approximation converter. The major advantage lies in the inherent filtering properties of the capacitors that limit the noise bandwidth. The switching operation on the capacitors creates kT/C noise. Next to that there is the noise from the input pair. After the switches are released this noise is integrated into the capacitors and follows a random walk till after some time constants an equilibrium is reached.

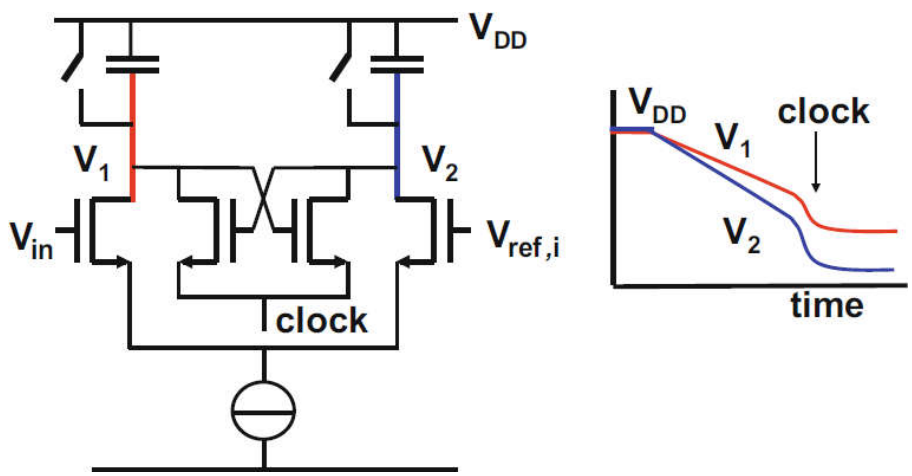


Figure 5-5 depicting a capacitive preamplifier and its charging and discharging output nodes [M.Pelgrom,2017]

5.4 Latch regenerative operation

A comparator in its regenerative phase is in the end a non-linear circuit: there is not a linear algebraic relation between input and output. However, in the early stages of regeneration the feed forward circuit can be described with linear equations. Figure 5.6 (left) shows the basic ingredients: the transconductance of the latch transistors form a positive feedback system controlled by the time constant $\tau = \frac{C}{gm,l}$, the starting point of the regeneration is the pre-amplified voltage $\alpha * VLSB$, where is $0 < \alpha < 1$ a multiplier for an LSB size voltage. Now a simple Laplace analysis shows that :

$$V(t) = \alpha V_{LSB} e^{+t/\tau} \tag{5.3}$$

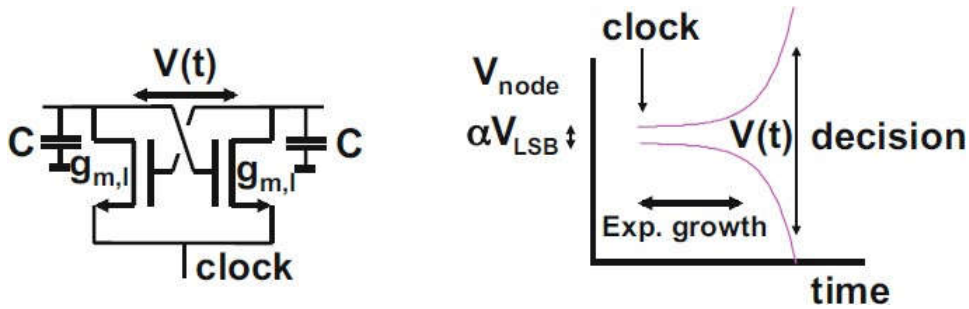


Figure 5.6 The latch behaves as a linear feed forward amplifier for most of the decision time [M.Pelgrom,2017]

The exponential signal growth will continue until a physical barrier is reached, e.g. one of the nodes reaches a power supply, or some non-linear limiting mechanism takes action.

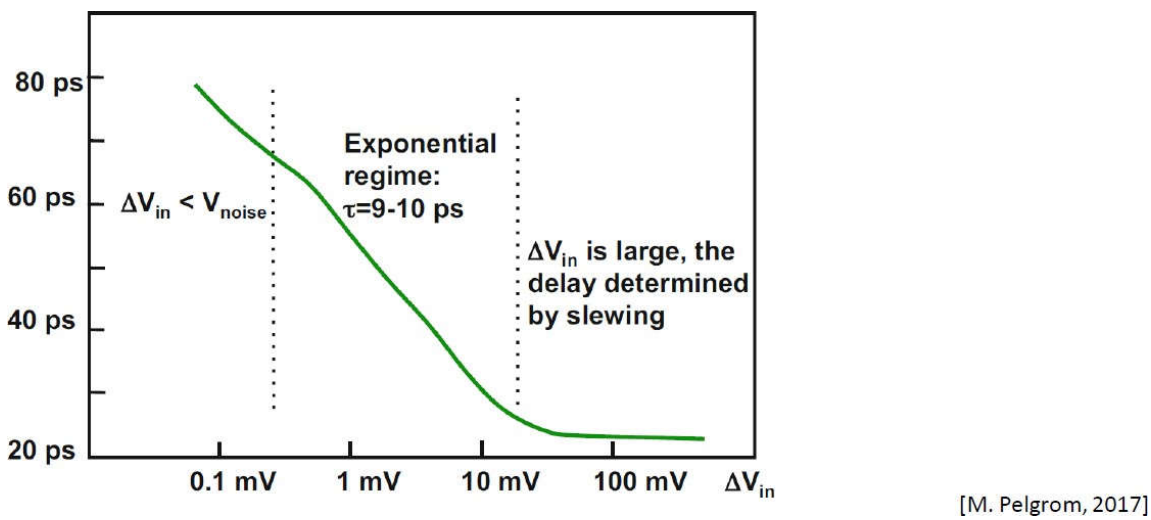


Figure 5.7 The delay of a comparator decision as a function of the input voltage difference. Simulation in 65-nm CMOS. Courtesy: Jianghai He, MCCI

Another view is presented in Fig. 5.7. A popular comparator schematic has been simulated in a 65 nm technology. The overall delay time from the activation by the clock until an arbitrary large amplitude has been reached is shown as a function of the differential input voltage. At high input voltages, the delay by the regeneration is not relevant as other circuits cause a minimum delay. Below 10mV the latch starts to create additional delay. As the input voltage is plotted on a logarithmic scale the delay shows a linear relation. Around 0.1mV the circuit reaches the noise levels. This comparator with lay-out parasitics has a regeneration time constant of 9–10 ps.

$$V(t) = \Delta V_{in} e^{+t/\tau} \rightarrow \; t_d = \tau \ln \frac{V_{FS}}{\Delta V_{in}} = \tau(\ln V_{FS} - \ln \Delta V_{in})$$

The delay of the comparator is determined by the differential input voltage.

We don't have a maximum value for the delay as it is dependent on the input voltage difference, but we do have minimum value for the delay set by the slewing of the Latch, which will convert it from exponential linear settling to non-linear settling (slewing).

The latch is a bistable circuit, it is not just a stable circuit, because assuming that it is lying in a metastable state (see next section), any small signal perturbation will push the circuit to either logic high or logic low, its mechanical equivalent is shown in Figure 5-8, we got state A and state B both resembling the high and low states while the top of the hill represents the Metastable state, any small signal perturbation will cause the circuit to fall into one of the two states that is why it is called bistable circuit.

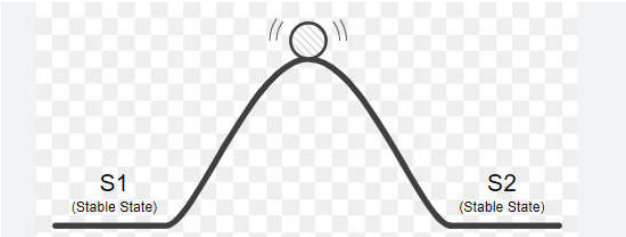
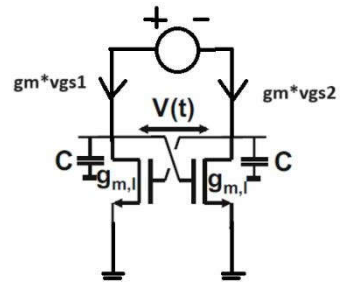


Figure 5-8 representing the mechanical equivalent of the metastable state.

During the reset state triggered by the inactive clock pulse, the output nodes of the latch are reset to either ground or VDD, depending on the architecture this is similar to breaking the positive feedback loop, the same way it happens in the D-flipflop latches where the loop is broken when a new input is being stored, and then in the next active clock pulse, the regeneration process can take place. The pre-amplifier part can be removed but the application must be of very low precision, because the latch itself has large offset and noise variations that will affect the precision of the device as it couples to the input terminals, the Preamplifier amplifies the differential input to make it robust against the noise and offset of the latch, it isolates the sensitive analog signal from the noisy digital latch.

Another important thing is that we don't need a very large gain for the Preamp, we have that in the Latch but what we do need in the Preamp is the high bandwidth.

Asserting on the positive feedback issue :



Assuming that the upper source is Vx giving current Ix then,

$gm * vgs1 = -gm * vgs2$ and assuming that the two transistors are identical.

Then, $vgs1 = -vgs2, \rightarrow Vx = vgs2 - vgs1 = -2 * vgs1$

$$Req = \frac{Vx}{Ix} = \frac{-2 * vgs1}{gm * vgs1} = -\frac{2}{gm}$$

This gives negative resistance which provides the positive feedback operation (bistable circuit), this gives an unbounded output, and thus the output keeps on increasing until the non-linearity of the system limits that increase (such as hitting the power rails).

$Ceq = \frac{C}{2}$ as they are in series, then the pole = $\frac{1}{R*C} = \frac{1}{\frac{2}{gm} * \frac{C}{2}} = \frac{gm}{c}$, this is not an oscillator as the poles do not lie on the jw-axis, this gives a Latch/bistable circuit, which makes the outputs grow exponentially in the time domain.

5.5 Metastability and Bit error rate

The regeneration delay in the comparator creates a fundamental problem in the form of metastability. Metastability is associated with any form of comparison and is also well known in, e.g., synchronization circuits. The crucial observation is that the time a comparator or latch needs to form a digital signal depends on the initial over-drive voltage. In other words: the input differential voltage determines the delay of the comparator.

For very small overdrive voltages, there is a fraction α of the LSB size where the comparator has insufficient voltage difference on its nodes to reach a decision in the limited time Ts of a clock pulse. Now the comparator will not generate a clear “zero” or “one” output level after time Ts. With ambiguous signals the succeeding logical circuitry can generate large errors, e.g. if this digital signal is crucial for determining the MSB.

For small signals the latch can be viewed as two single-pole amplifiers in a positive (regenerative) feedback mode. The voltages on the nodes of the latch develop exponentially in time with a time constant , see Eq. 5.3 This time constant is determined by the internal node capacitance and the transconductance of the latch transistors. Now the critical value of α below which there is ambiguity can be approximated by

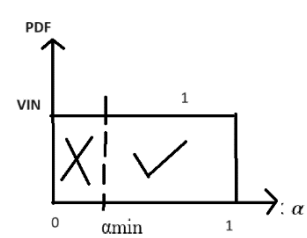
$$\alpha \approx \frac{V_{latch}}{V_{LSB}} e^{-T_s/\tau} \quad \Rightarrow \quad \text{BER} \approx 2^N e^{-T_s/\tau} \tag{5.4}$$

where the ratio between the maximum latch swing $Vlatch$ and $VLSB$ is estimated as 2^N. The signal has a uniform probability arriving between 0 and $VLSB$. If the range between 0 and $\alpha VLSB$ causes errors, then the bit-error rate equals α. The bit error rate is an important parameter in the design of fast converters with a high accuracy. In CMOS the time constant τ is formed by the parasitic and gate capacitances and the achievable transconductance.

A method to get an impression of the bit error rate is to apply a slow sine wave to the device with an amplitude that guarantees that no more than one LSB change occurs at the digital output. Now the bit error rate can be estimated by recording output codes that differ more than one bit from the preceding code.

From a fundamental point of view the BER cannot be avoided completely, however decreasing the time constant (by lower capacitance and higher transconductance), a BER of 10^{-13} is possible. Measures to improve the BER include improving the latch speed with more current and smaller capacitances, additional gain stages or even a second latch, and a special decoding scheme avoiding large code errors due to a metastable state.

Now let's break down that BER into some graphs and equations, Assuming that there will be an error when the input is less than $\Delta Vin = \alpha VLSB$ where $0 < \alpha < 1$, since the differential input voltage makes the comparator delay large, so we are under the possibility that Ts comes to take the resorted data before the comparator actually reaches a steady state either zero or one, in that case $V(Ts) < VFS$, and from eq 5.3, $V(Ts) = \alpha * VLSB\, e^{\frac{-Ts}{\tau}} < VFS$ then $\alpha \min = \frac{VFS}{VLSB}\, e^{\frac{-Ts}{\tau}} = 2^N\, e^{\frac{-Ts}{\tau}}$

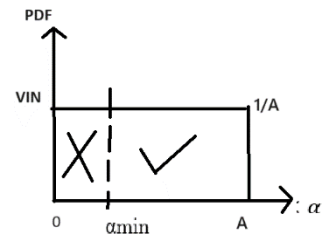


Assuming Vin has uniformly distributed probability density function, and the comparator will produce ambiguous states when $Vin < \alpha \min * VLSB$ then BER = αmin = $2^N e^{\frac{-Ts}{\tau}}$, where the BER can never be zero ($e^x \neq 0$)

no matter how fast the latch becomes. From the PDF we can see that the BER is the area up until αmin.

In order to improve the BER we can either improve the latch speed by decreasing the capacitive loading or by increasing the current and thus increasing the transconductance, or we can increase ΔVin by using multiple gain stages where $0 < \alpha < A$ where A is more than one.

Where here BER = $\frac{\alpha min}{A}$. Another method to improve the BER is use a second latch, the same idea as the synchronizer, where there is an asynchronous system and using synchronizing clocks and d-flipflops or gates we can synchronize it to the system, in our case the latch can give another extra Ts for the comparator latch to resolve its output and thus limiting errors, as we take the output from the second circuit not the first, in the second latch we can install some special decoding schemes which forces either zero or one when it detects metastability in order to recover, but as we found BER can never be reduced to zero.



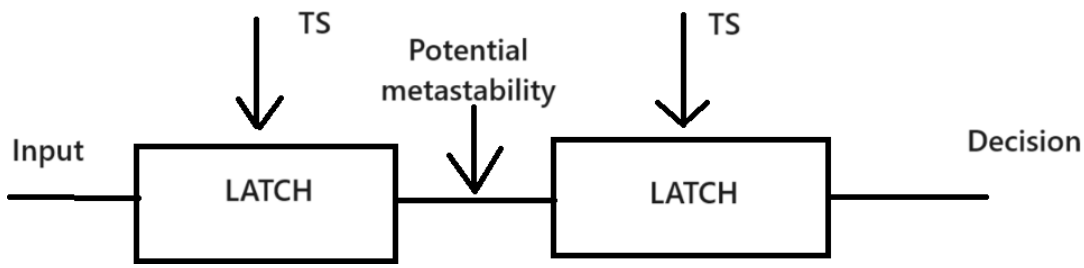


Figure 5-9 indicating a case where potential metastability and high BER can be mitigated by using two latches and sampling at 2*TS instead of a single TS.

5.6 Accuracy and comparator offset voltage

The required accuracy of a comparator depends on the required specification and the architecture of the analog-to-digital converter. In dual-slope, successive approximation and pipeline designs, the static random offset is a second-order effect and generates a (random) DC-shift of the entire signal. In a flash converter the input referred random offset is crucial because it will shift the individual reference levels and impacts both integral and differential non-linearities. Also, in time-interleaved analog-to-digital converter, the random offset returns as a spurious tone.

The input referred random offset σ_{Vin} must be designed to a value a factor 5 to 10 lower than the size of the LSB at the comparator as an initial target for e.g. flash converters. Fine-tuning

of this requirement depends on, e.g., the tolerance for conversion errors on system level. In some other cases a spec of 0.5LSB<3 $\sigma_{in,os}$. (meaning that 99.7% of the units have offset less than 0.5LSB).

Accuracy has static and dynamic components. Just as in every analog circuit the static accuracy is (in a carefully laid-out circuit) determined by the random mismatch of a transistor pair, Of course, the input differential pair is not the only contributor, also the mismatch of the load and the latch must be taken into account. We can see this in a classification of offset voltage components:

Offset voltage has static and dynamic components :

Static offset which is dominant in preamplifiers:

1. Systematic offset (due to systematic mismatch): can be minimized by good design and matched layout (Common centroid, inter-digitation and placement of dummies).
2. Random offset (due to random mismatch): reduced by using large devices but cannot be eliminated (Pelgrom s model)

Dynamic offset:

- Due to imbalanced switching effects and capacitive load mismatch
- Dynamic offset is dominant in latches

In the schematic diagram of Fig.5.10 current differences caused by the input pair, the load and the latch all come together on the drains of the transistors. All contributions can be referred back to an equivalent input referred random mismatch σVin. With the help of Eq.5.5, the input referred random offset is described as:

$$\sigma_{Vin}^2 = \sigma_{VT,12}^2 + \frac{g_{m,34}^2}{g_{m,12}^2}\sigma_{VT,34}^2 + \frac{I_{load}^2}{g_{m,12}^2}\frac{\sigma_R^2}{R_{1,2}^2} \qquad (5.5)$$

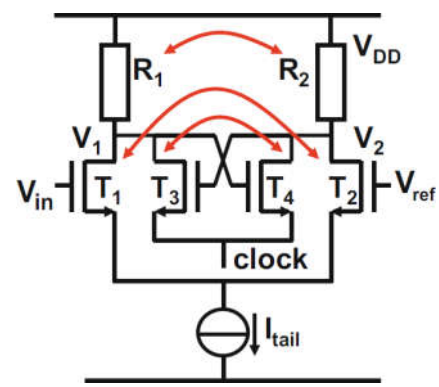
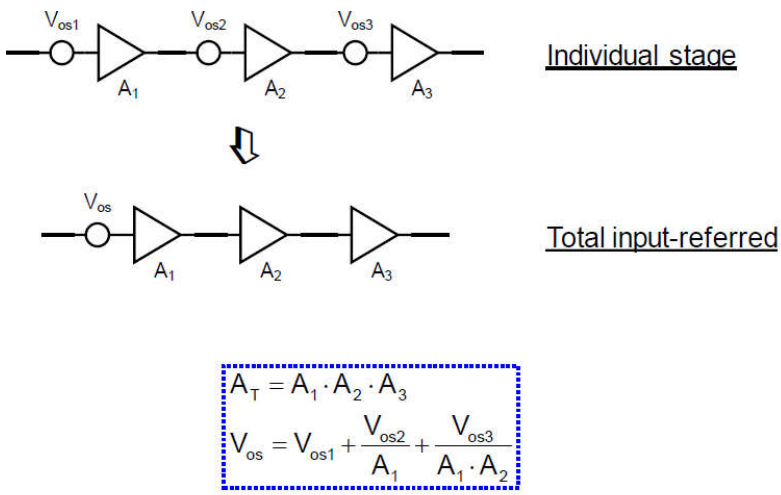


Figure 5-10 Mismatch sources in comparator

This is based on the concept Multistage amplifier offset as can be seen in Figure 5-11.



[Y. Chiu, EECT 7327, 2014]

Figure 5-11 Describing how to refer offset values to the input of a multistage amplifier, of course the most pronounced offset is the one generated from the first stage that is why it has to be the one with the best offset spec.

In eq. 5.5, The equation assumes that the random mismatch can be calculated in a linearized model of the circuit. Unfortunately, in this equation the transconductance of the latch does not behave in a linear fashion. In one extreme there is no current running in the latch and its contribution to the input-referred random offset is zero. Directly after the clock is activated and a current flows in the latch transistors, a step function on the nodes $V1$ and $V2$ will occur. If the gates are not equal or in the presence of mismatches due to differences in capacitive loading, different charges are drawn from the local nodes and will create random differences in these voltage steps. If, on the other hand, the latch is constantly fed with a small tail current that is not sufficient to activate the latch, the dynamic errors can largely settle before the clock is activated. However, the non-zero transconductance will contribute an additional component to the input-referred random offset.

About dynamic offset coming from a latch let's discuss it on Figure 5-12 :

- ❑ M5-6: input pair
- ❑ M7-8: cross-coupled load (regenerative)
- ❑ M9: reset switch
- ❑ Dynamic offset due to:

- ▪ Imbalanced CI and CF
- ▪ Imbalanced load capacitance
- ▪ Mismatch between M5 and M6
- ▪ Mismatch between M7 and M8
- ▪ Clock routing

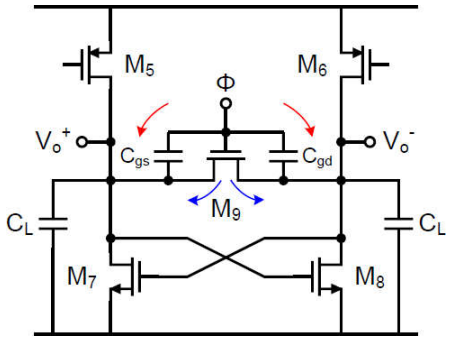


Figure 5-12 discussing the Dynamic offset sources coming from the Latch. [Y. Chiu, EECT 7327, 2014]

Now, let's discuss methods for offset cancellation, two of the most common methods are input and output offset cancellation.

First, input offset cancellation, observe the circuit in Figure 5-13, as we can see it is a two-phase operation, Ø1 is amplification phase and Ø2 is offset storage phase where the offset voltage is stored on the capacitor, now let's discuss it analytically :

At Ø2 :

The circuit is a unity gain closed loop op amp, and so the gain is destroyed but there is closed loop stability that we require and if we tried to do compensation, GBW will decrease, some tricks can be done like miller compensation which is only operated in the offset storage phase.

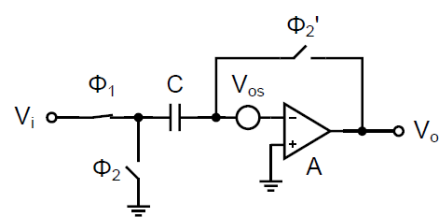


Figure 5-13 Input offset cancellation example circuit.

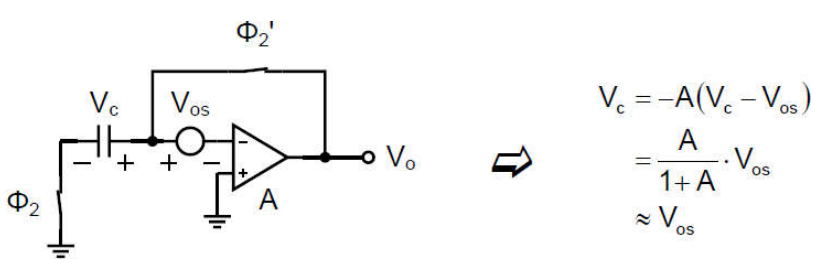


Figure 5-14 Ref: J. L. McCreary and P. R. Gray, "All-MOS charge redistribution analog-to-digital conversion techniques. I," JSSC, vol. 10, pp. 371-379, issue 6, 1975.

At Ø1, the amplification phase, the offset cancellation is incomplete if A is finite, also input AC coupling attenuates the signal gain due to the capacitance at the input terminals of the OTA which divides the voltage with the offset storage capacitor and causes input signal gain attenuation and thus C must be much greater than Cin :

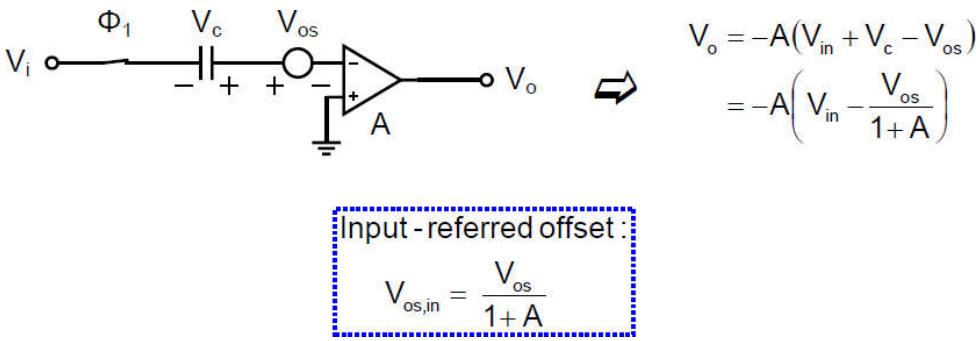


Figure 5-15 [Y. Chiu, EECT 7327, 2014]

As we can see, the offset is still there, and thus fully differential operation is a must in order to remove the common mode switching errors which is necessary for high precision switched capacitor integrated circuits.

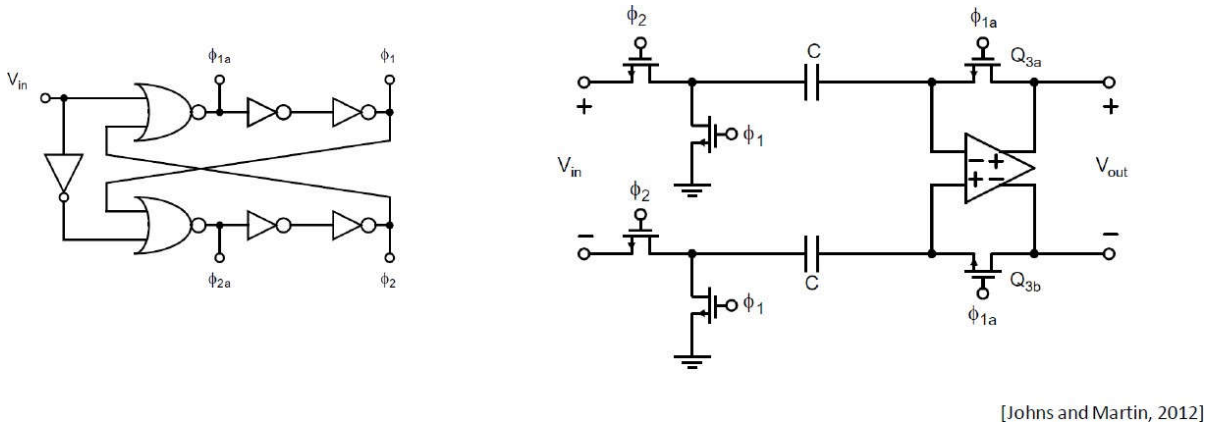


Figure 5-16 Fully differential operation

Second, the output offset cancellation, the problem of -ve feedback (the closed loop stability) is not present here

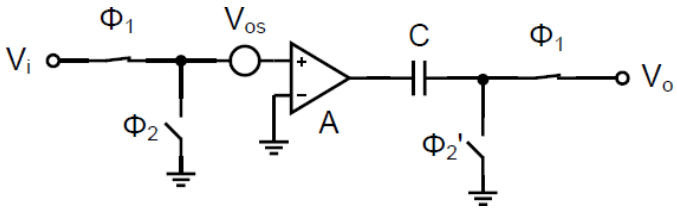


Figure 5-17 Output offset cancellation circuit [Y. Chiu, EECT 7327, 2014]

At Ø2 , the offset storage phase :

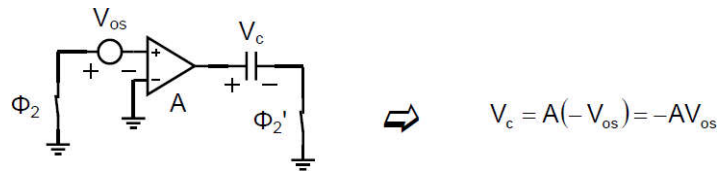


Figure 5-18 Ref: R. Poujois and J. Borel, "A low drift fully integrated MOSFET operational amplifier," JSSC, vol. 13, pp. 499-503, issue 4, 1978. [Y. Chiu, EECT 7327, 2014]

- AC coupling at output with offset stored in C
- The gain (A) must be small and stable (independent of Vo), so it must be a linear amplifier so that the gain seen by the offset remains the same in both phases.
- Does not work for high-gain amplifier, as it causes the amplifier output to saturate at the ± rails.
- CF (clock feedthrough) and CI (charge injection) of Φ2' gets divided by A when referred to input.

At Ø1, the amplification phase :

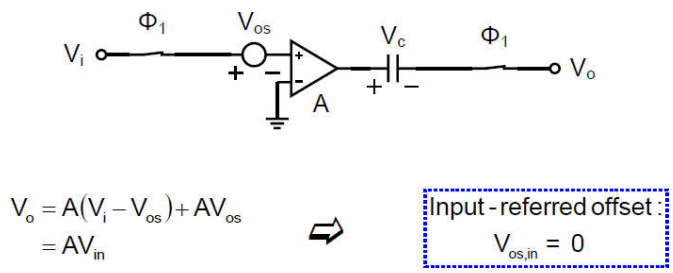


Figure 5-19 Amplification phase [Y. Chiu, EECT 7327, 2014]

- Cancellation is complete if A is constant (independent of Vo)
- AC coupling at output attenuates signal gain

5.7 Noise in a comparator

Thermal noise and 1/f noise create additional accuracy limitations. Because these contributions are time varying, also offset compensated analog-to-digital converter architectures suffer from this limitation. The contributions to the noise of the individual components are referred back to an equivalent input noise source in a similar manner as for mismatch. If the input transconductance dominates the noise, the effective value of the input-referred noise is:

$$v_{in,noise} = \sqrt{4kT\frac{BW}{g_m}} \tag{5.6}$$

with a bandwidth of 10 GHz and an input transconductance of 1 mA/V the value for the input referred rms noise is Vin, $noise$ = 400uV. The thermal noise amplitude distribution creates a cumulative Gaussian probability distribution around the trip level, Fig.5.20 In order not to exceed the accuracy specifications the bandwidth has to be reduced or more power has to be spent on improving the impedance levels.

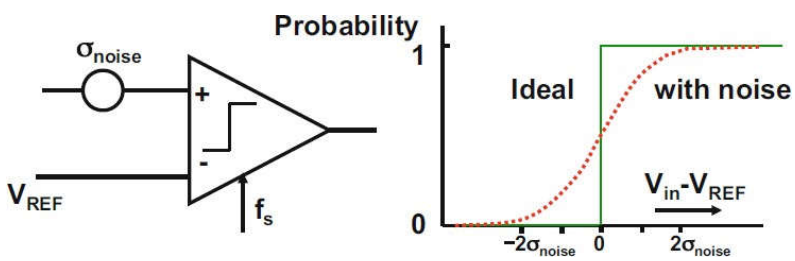


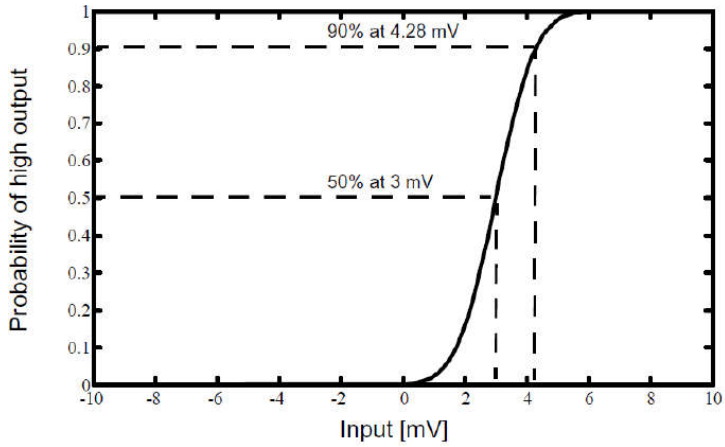
Figure 5-20 Thermal noise will create a probabilistic behavior of the comparator decision (dotted line) [M.Pelgrom 2017]

The input referred noise acts as time varying input offset, there is a trade-off between noise, speed and power consumption, so if the thermal noise dominates we get the following equation :

$$v_{nout,rms}^2 = v_{nin}^2(f) \times A_v^2 \times B_N = \frac{4kT\gamma_{eff}}{g_m} \times A_v^2 \times B_N$$

$$v_{nin,rms} = \sqrt{\frac{4kT\gamma_{eff}}{g_m} \times B_N} \tag{5.7}$$

For the input referred noise to decrease, power consumption must increase, and speed must decrease, BN refers to the equivalent noise bandwidth.



[Johns and Martin, 2012]

Figure 5-21 This is a simple example to calculate input referred offset and noise from the curve.

The offset simply is 3mv which is the center of the curve at the point of 50%, and since from the properties of the gaussian distribution is that 1.28 σrms from the mean is proportional to 90 %, the same way 3 σrms from the mean is proportional to 99.7%, then 1.28 σrms is equal to 4.28mv – 3mv, then 1.28 σrms = 1.28mv, then σrms = 1mv.

5.8 Kickback noise

The comparator is a non-linear element: the only relation between the input signal and the output is the sign. The decision process in the comparator is often implemented with transistors that pass through all operation modes of the devices. The charges controlling the devices will show considerable variation. These charges have to be provided and will be dumped preferably in the power supply lines, however also an exchange to the input and reference signals will occur.

Figure 5.22 (left) shows a differential input pair loaded with a latch. The input voltage difference will lead in the succeeding circuits to a decision marked by sharp voltage transitions. The drain-gate capacitors of the MOS devices forming the input pair will pass these fast edges to the input and reference nodes thereby generating "kick-back." In some comparator designs these transitions can be as large as the power supply voltage. In this topology the drain of one input device will be pulled to *VDD* and remain in inversion, while the current of the other device will go to zero. Its inversion charge will flow either into the terminal, either the reference or input. This will cause an even larger kick-back effect.

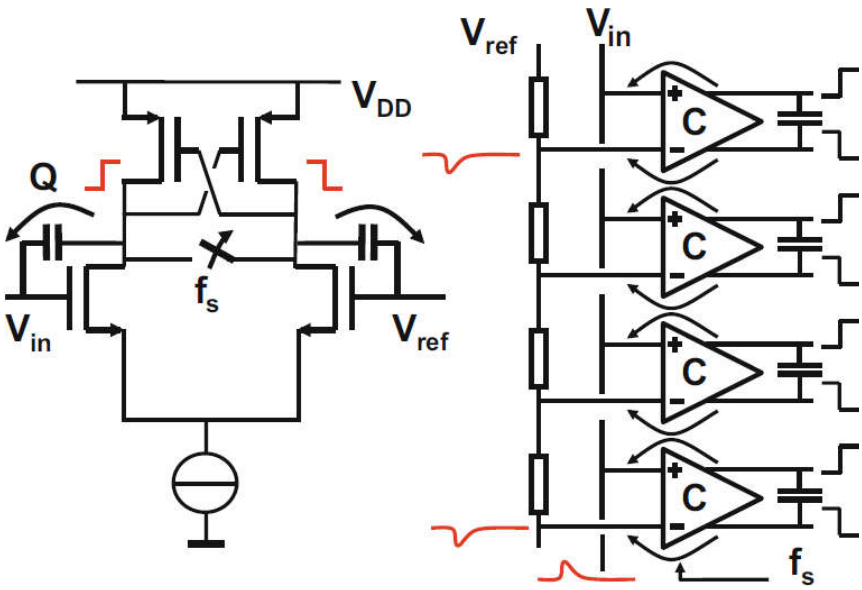


Figure 5-22 Kick-back in a comparator circuit causes currents in the input and reference structure [M.Pelgrom 2017]

Figure 5.22 (right) shows a flash circuit. In this architecture the kick-back effect is strong as the contributions of many comparators add up. During the simulation of the comparator ideal voltage sources instantly drain the charges from saturation effects and therefore will result in an optimistic performance prediction. Realistic impedances at the input terminals of the comparator allow a better simulation prediction of the performance in the presence of kick-back.

Reducing the kick-back signal requires to decrease the coupling capacitances by reducing the dimensions of the input related circuit elements with a penalty of higher random offset. Inside the comparator the swing of the signals must be sufficient to drive the next stages but can be kept as low as a few tenths of a Volt by swing limiting measures. The additional advantage is that the transistors in the comparator remain in inversion, thereby avoiding time loss due to recharging. A rigorous method applies an additional preamplifier that would isolate all that from the input terminals . Unfortunately, an additional stage often compromises the (speed, power) performance and will create additional delay.

To summarize :

- The output has sharp voltage transitions, we can try to limit these output transitions by limiting circuits like diodes, because it is these transitions that couple through Cgd to the input of the comparator.

- Cgd will pass these fast edges to $Vin\ and\ Vref$ causing kickback i.e. Reference voltage disturbance must settle before next sample

- Reducing Cgd by reducing sizing will make mismatch worse

5.9 Hysteresis

A comparator is designed to discriminate between positive and negative levels at its input terminal, irrespective of how small these levels are and what the previous decision was. In many comparator designs this ideal situation is not achieved. Either intentionally or as an unwanted consequence of the topology, the comparator remembers its previous state. Figure 5.23 shows an elementary comparator consisting of two inverters. The output of the second inverter is partly fed back in positive phase to the input. The comparator input has now a preference to keep its present

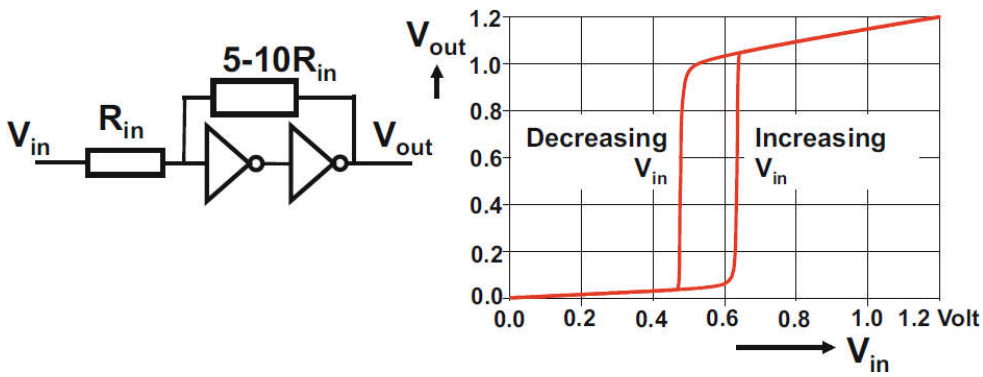


Figure 5-23 Hysteresis created by positive feedback in a two-stage buffer [M.Pelgrom 2017]

state and a small input value around the trip level of the inverter will not cause the output to change. Depending on the resistor ratio, the input will see a trip level that depends on the present state of the output. Going from negative input to positive or vice-versa results in different switching characteristics, see Fig. 5.23 (right). This effect is in analogy with magnetic materials called "hysteresis". In this example the hysteresis creates an additional threshold for change. The example circuit in Fig. 5.23 belongs to the class of "Schmitt-trigger" circuits. These comparators use the hysteresis threshold to avoid unwanted transitions in case of noise signals. In other words, a Schmitt-trigger circuit decides only if a clear input signal is present, and the hysteresis avoids that noise generates false outputs.

The resolution of the comparator can be no less than the hysteresis, large values of hysteresis are generally BAD for a high-resolution comparator as they will define the resolution of the comparator.

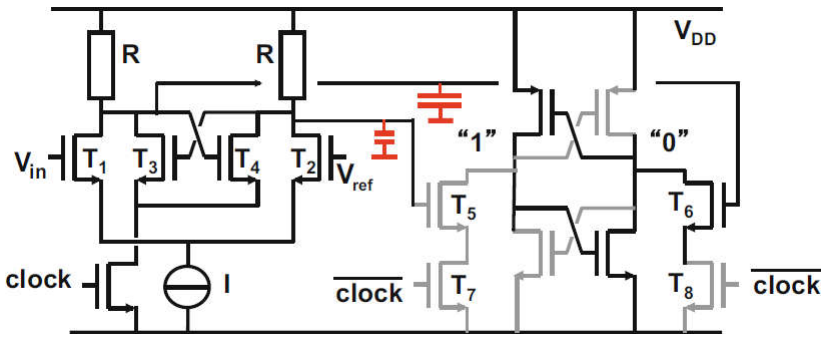


Figure 5-24 Hysteresis created by the signal-dependent loading of the input latch by the second latch. The *grey-colored* transistors are not conductive and form only a small load capacitor [M.Pelgrom 2017]

A practical example of unwanted hysteresis is a pre-latch stage that is connected to a second stage consisting of a full digital latch activated by the inverse clock. In Fig. 5.24 the signal is fed from the amplifier to the full latch via two transistors $T5$ and $T6$ that are in series with the inverse-clocked transistors $T7$, $T8$. The left-hand transistors $T5$, $T7$ will be out of inversion when that side of the latch stores a "1" signal. The connection to the gate of $T5$ sees only some parasitic loading. The righthand transistor $T6$ sees a "0" signal and its channel is in full inversion. The inversion layer is connected to ground level creating a full gate-capacitor load. Thereby the two coupling transistors $T5$ and $T6$ create an unequal capacitive loading of the first stage. This loading favors a similar signal in the first latch when the latch sees only a small input signal. The consequence is a comparator with hysteresis.

In other configurations the opposite effect may be the case: the hysteresis favors change and the comparator toggles at every clock cycle for an input signal close to the trip level.

The previous state can be removed by resetting the latch in the comparator. Some designs simply disconnect the latch from the supply. This method assumes that there is sufficient time to discharge the latch nodes. A more active approach consists of adding a specific clear or reset mechanism to the latch. Figure 5.25 shows two configurations. The left-hand circuit connects the latch nodes to the power supply and actively removes all state information. The low-ohmic switches change the DC-levels in the amplification branch. The DC-level must be restored and during that process any inequality in capacitive loading will show up as inaccuracy.

The construction on the right-hand side circumvents the restoration problem by pinching the two latch nodes together. This will keep the average DC-level of the nodes intact and set the latch to its trip point. A proper choice of the impedance of the switch allows us to use it as a

differential load for the current of the input pair. The reset transistor must be bootstrapped in case of low power supply.

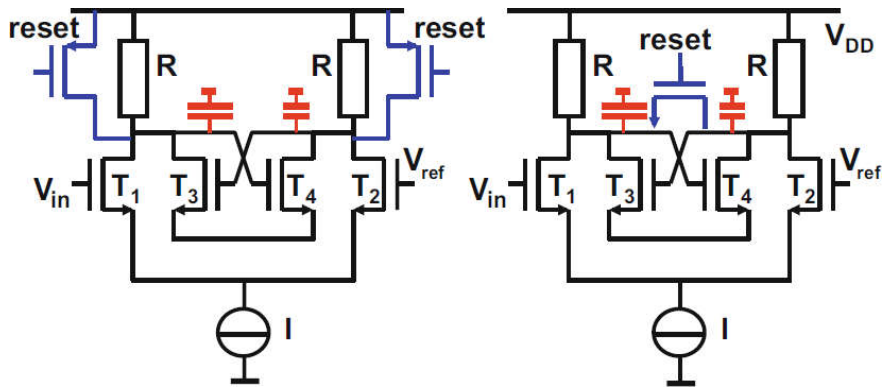
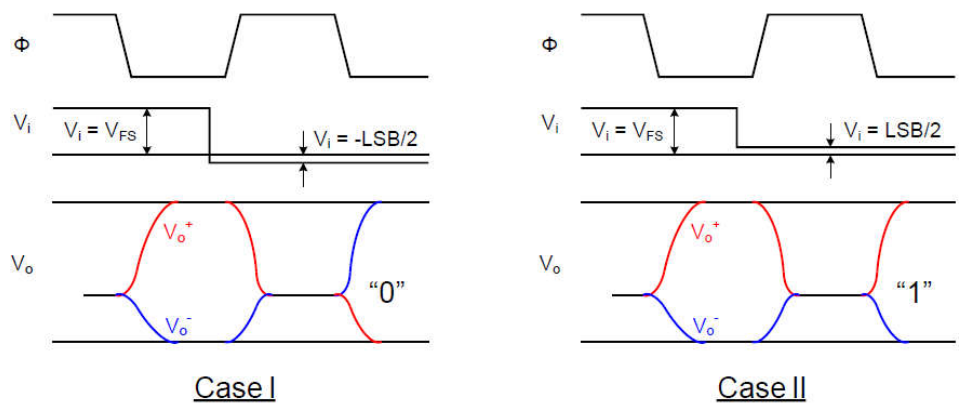


Figure 5-25 Two methods to clear the state in comparator circuit [M.Pelgrom 2017]

One of the tests that we will use later is the overdrive recovery test, it is a test that determines the hysteresis of the comparator, A small input (±0.5 LSB) is applied to the comparator input in a cycle right after a full-scale input is applied :

- The comparator should be able to resolve to the right output in either case → memoryless



[Y. Chiu, EECT 7327, 2014]

Figure 5-26 Observe the case here where (±0.5 LSB) input is applied right after a full-scale input,, in both cases the comparator resolved correctly which means that the hysteresis is less than 0.5LSB

5.10 Initial design step

First of all, before we even start with the schematic architectures of the comparator what we need to know is the flow of design, at first we need to understand the comparator's operation and we did, then we should create a behavioral model that would perform that operation, the behavioral model would be in the behavioral modelling phase of the system where all components are VerilogA models, the used VerilogA code for the comparator is as the following :

```
1  `include "discipline.h"
2  `include "constants.h"
3  // model CompLatched - Comparator
4  //==========================================================================
5  module lab_07_comp (VINP, VINN, CLK,CLKN, VOP, VON, VGND, VDD) ;
6  // VGND and VDD are included to be pin-accurate with the schematic view
7  // PINS
8  input VINP, VINN, CLK,CLKN;
9  output VOP, VON;
10 inout VGND, VDD;
11 electrical VINP, VINN, CLK,CLKN, VOP, VON, VGND, VDD;
12 // INSTANCE PARAMETERS:1
13 // vhigh = Analog value for a Digital 1 on output [V]
14 // p_off = Offset voltage added to VINP pin [V]
15 // slack = The smallest time interval considered negligible
16 // (abstol for clock) [S]
17 // td = intrinsic delay from threshold cross to output 50%
18 // change [S]. H: That's why I use td - trf/2
19 // trf = Transistion time for output rise/fall [S]
20 // vth = CLK pin's threshold voltage [V]
21 // vlow = Analog value for a Digital 0 on output [V]
22 parameter real p_off = 10.0u;
23 parameter real trf = 1n from (0:inf);
24 parameter real td = 1000n from (trf/2:inf);
25 parameter real vhigh = 2.0;
26 parameter real vlow = 0.0;
27 parameter real vth = 1.0;
28 // LOCAL VARIABLES:
29 real vin, outstateP,outstateN,flag=0;
30 //==========================================================================
31 // Room for development -> make the COMP RESET time smaller than the comparison time
32 analog begin
33 @(initial_step) begin
34
35 // We will set both outstateP and outstateN initially to zero to make sure that the
36 // CLOCK oscillates and avoid the conversion issue
37
38 //if(V(CLK) == vhigh) begin
39 //vin =V(VINP,VINN) + p_off;
40 //outstateP = (vin > 0.0) ? vhigh : vlow ;
41 //outstateN = (vin < 0.0) ? vhigh : vlow ;
42 //outstateP = 0 ;
43 //outstateN = 0 ;
44 //end
45 //else if(V(CLK) == vlow) begin
46 outstateP = 0 ;
47 outstateN = 0 ;
48 //end
49 end
50 // when re-CLKd need to determine if condition has changed
51 @(cross(V(CLK)-vth,+1)) begin // on RISING edge
52 flag=1;
53 vin =V(VINP,VINN) + p_off;
54 outstateP = (vin > 0.0) ? vhigh : vlow ;
55 outstateN = (vin < 0.0) ? vhigh : vlow ;
56 end
57 // ADD THE FOLLWONG CONDITION TO RESET PHASE
58 @(cross(V(CLK)-vth,-1)) begin // on FALLING edge
59 outstateP = 0;
60 outstateN = 0;
61 end
62 // TRYING to avoid convergence issues
63 @(timer(10n,0)) begin
64 outstateP = vhigh;
65 end
66 if(V(CLK) == vlow) begin
67 outstateP = 0 ;
68 outstateN = 0 ;
69 end
70 // normal and complement outputs in the case of the HIGH clock, and ensure
71 // that both outputs are RESET when the clock is LOW
72 V(VOP) <+ transition (outstateP, td-trf/2, trf, trf );
73 V(VON) <+ transition (outstateN, td-trf/2, trf, trf );
74 end
75 endmodule
76
```

Figure 5-27 VerilogA code that was created by Dr.Hesham Omran initially then we did multiple changes to obtain the desired operation

This VerilogA code models the operation of a comparator, that does the comparison part when the clock pulse arrives and resets both of its outputs when the clock is LOW, there are auxiliary circuits that are added to make the operation asynchronous as we will see later. From this point we tested the comparator and saw that it gives the required results as you can see in the following graph :

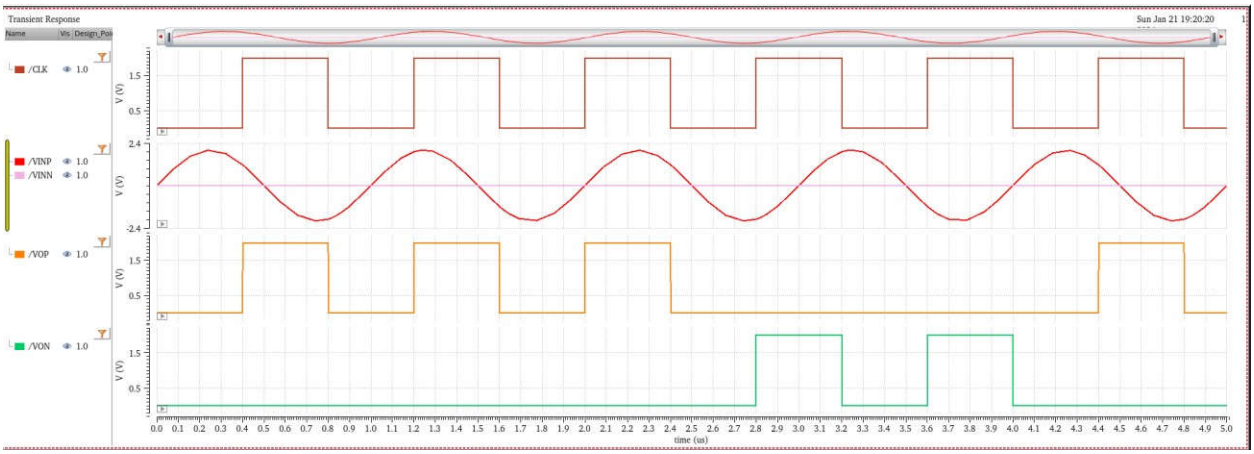


Figure 5-28 Viva graph showing the operation of the VerilogA code of the comparator

Now what we want is to create a complete asynchronous loop that would generate the clock for the rest of the system, so we had two proposals for an asynchronous loop, the first one is:

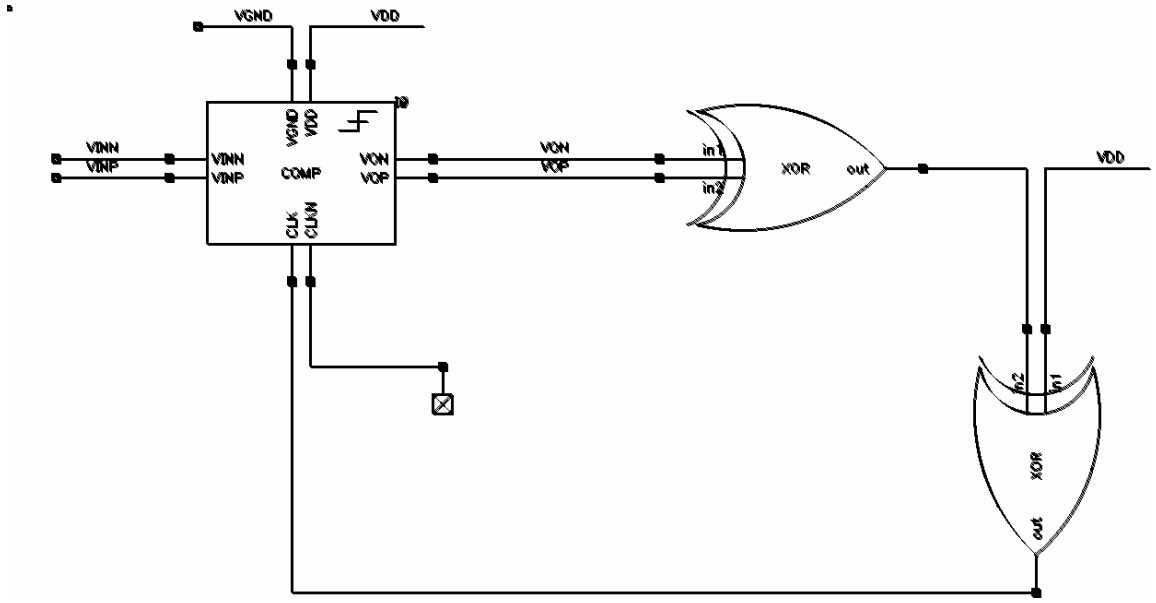


Figure 5-29 First proposal of the asynchronous loop

```
// VerilogA for GradProject, XOR, veriloga

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

// 2-input Exclusive Or gate
//

module axor (out, in1, in2);

output out; voltage out;
input in1, in2; voltage in1, in2;
parameter real vh = 2;            // output voltage in high state
parameter real vl = 0;            // output voltage in low state
parameter real vth = 1;  // threshold voltage at inputs
parameter real td = 10n from [0:inf); // delay to start of output transition The range [0:inf) indicates that td can take any non-negative real value. This will estimate the Comparison time
parameter real tt = 100p from [0:inf);// transition time of output signals The range [0:inf) indicates that tt can take any non-negative real value.
real outstate,flag;

// The goal of using the flag is that we want the XOR to start at ZERO state first and then respond to the changes at the input

analog begin
    // Input signals can only be sensed, they cannot be affected, while output signals can be affected but not sensed.
    @(initial_step) begin
    outstate = 0;
    flag = 0;
    end
    @(timer(100n,0)) begin
    flag=1;
    end
    if(flag==1) begin
    outstate = ((V(in1) > vth) ^ (V(in2) > vth)) ? vh : vl;
    end
    V(out) <+ transition( outstate , td, tt );
end
endmodule
```

Figure 5-30 VerilogA code for the XOR gate, we added the timer function to initiate the READY signal in order to avoid convergence issues.

The second proposal is the loop created by P.Harpe in his paper which is as the following:

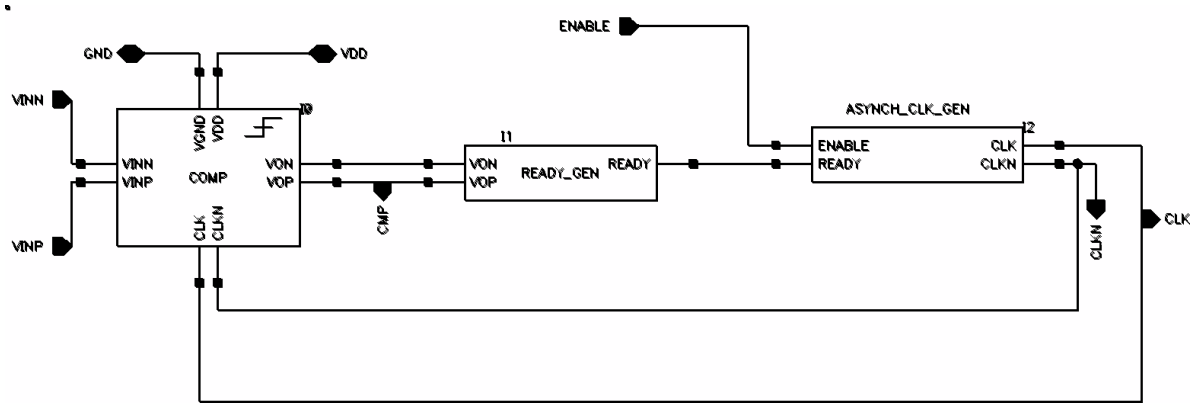


Figure 5-31 Second proposed loop .

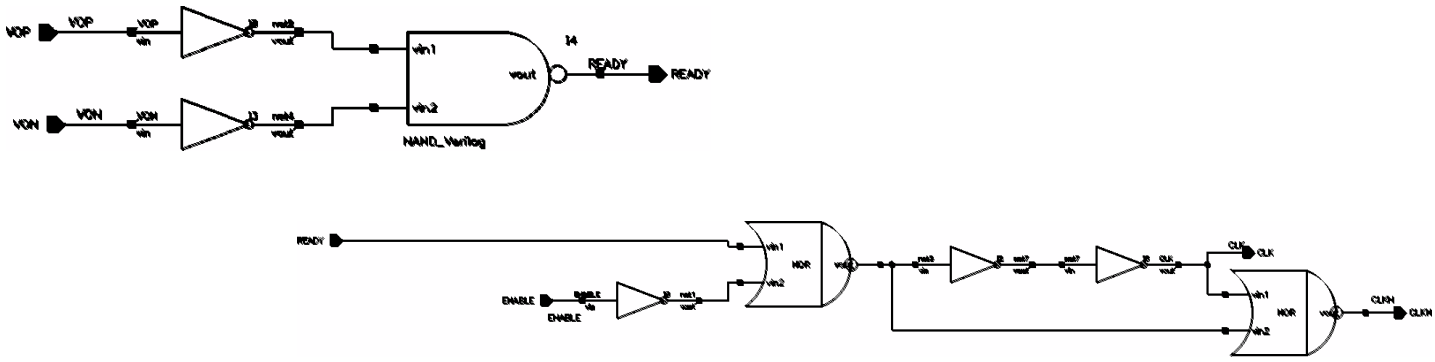


Figure 5-32 Schematics of Ready Gen. on the left and Asnych. CLK Gen. on the right.

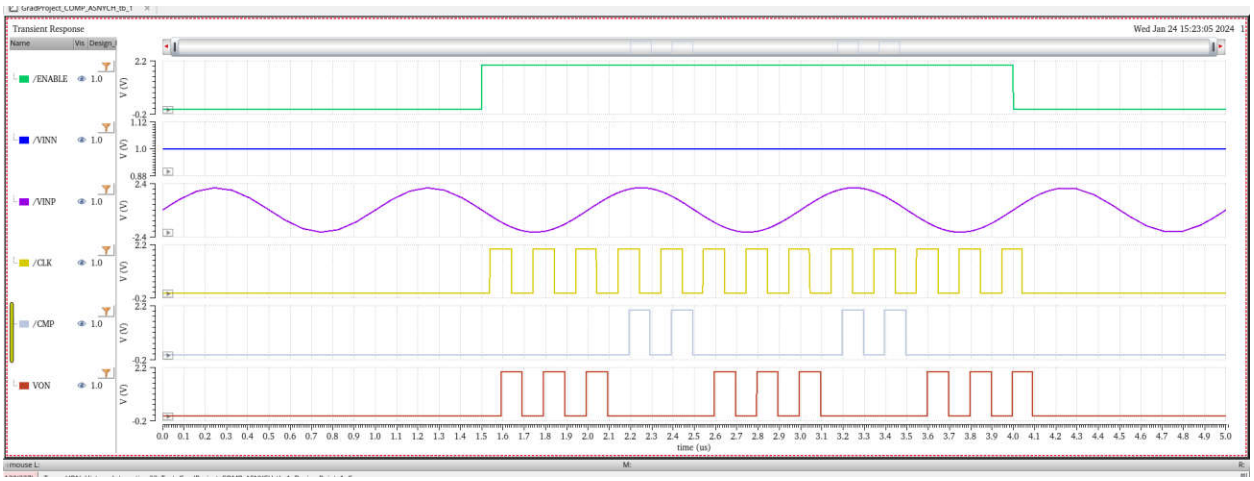


Figure 5-33 Viva graph showing the complete operation, at first started off at 2v VDD but then we lowered it to 0.6v

It is important to understand that clock of the comparator depends on the delays set by the VerilogA models, and later on the by the delays that are intentionally placed.

We decided to move forward with the second proposal because it offers more features like the Enable option, which is crucial for our operation in order to save power mainly, and other features as well.

5.11 Scouring comparator architectures for the best choice

Now we will study different types of the comparator architectures and choose amongst these designs what best fits our application. This section first reviews low-power design techniques for a standalone comparator. Then, it surveys techniques that take advantage of the architectural property of the SAR ADC to reduce the total comparator power consumption.

Standalone Low-Power Comparator Design Techniques

1) Low-Power Voltage-Domain Comparators: The Strong-Arm latch, in Figure 5-34, can be considered the baseline comparator design. It is simple and widely used, but it has limitations in the performance, including energy efficiency and speed. Over the past decades, researchers have developed several advanced dynamic comparators that outperform the Strong-Arm latch in various aspects.

One such design is the two-stage dynamic comparator proposed by Schinkel *et al.* [26]. It separates the latch stage from the pre-amplifier stage. As a result, it greatly accelerates the latch regeneration speed, reducing comparator delay. Nevertheless, the drawback is that the input transistor of the latch starts from the linear region, which reduces the effective interstage gain. This results in increased noise and offset coming from the latch. To overcome this issue, Miyahara *et al.* [27] and Van Elzakker *et al.* [28] modified the topology of the latch to ensure the latch input transistors start from the saturation region, leading to a higher gain and reduced comparator noise and offset. Another effective technique to boost the gain, as presented by Hsieh and Hsieh [29], is to use cascaded input pairs. By vertically stacking three input pairs, it equivalently increases the pre-amplification time by threefold, achieving a 3 times gain. Overall, its comparator energy efficiency is improved by 2 times thanks to the further attenuated latch noise.

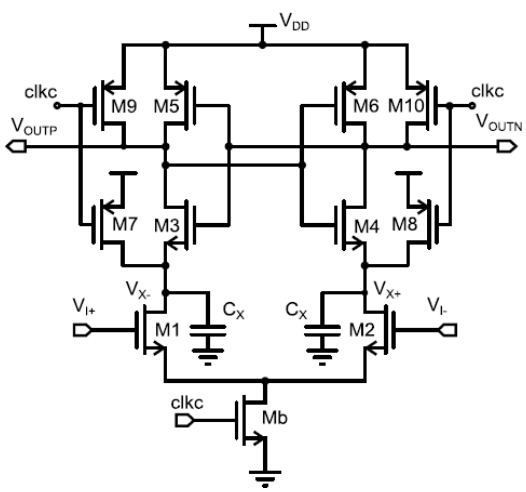


Figure 5-34 Strong-arm latch comparator

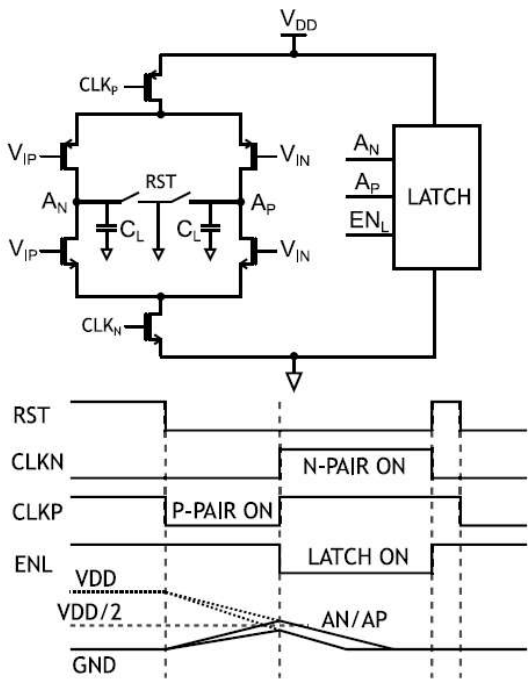
$$\overline{v_{n,pre}^2} \approx \frac{I_D}{g_m} \cdot \frac{4kT\gamma}{V_{thn}C_x} \qquad (5.8)$$

Input referred noise of the SA latch

The aforementioned techniques achieve higher energy efficiency by increasing the effective pre amplification gain. However, the pre-amplifier load Cx discharges completely to ground during comparison, consuming a fixed energy of $2 * CX * VDD^2$. Since large Cx is required to suppress the noise, as indicated in (5.8), the pre-amplifier consumes the majority of the power in a low-noise comparator design, e.g., approximately 80% in [26]. Hence, there is a strong need to improve the energy efficiency of the pre-amplifier itself.

To save the pre-amplifier reset power, Liu *et al.* [30] presented a bi-directional dynamic comparator. As shown in Fig. 5.35 , the pre-amplifier stage consists of both NMOS and PMOS input pairs. During the first half of the pre-amplification phase, the PMOS pair switches on so that the pre-amplifier loads are charged up from ground. Once they cross half VDD, the PMOS side turnsoff, and NMOS pair turns on to perform the second half of the pre-amplification. As a result, the

pre-amplifier loads return to their initial condition, which avoids the reset power. It only consumes an energy of $CX * VDD^2$while realizing the same gain and noise performance as the conventional design. Theoretically, it can improve the pre-amplifier energy efficiency by 2 times. Limited by the extra circuits needed for operation control, the overall comparator energy efficiency improvement is 1.5 times compared to that of a Strong-Arm latch. To further improve this, Tang *et al.* [31] combines bi-directional pre-amplifier with a common-gate stage of [32]. On top of the reset power saving, it extends the pre-amplification time and realizes 3 times gain boost. Yet, the overall efficiency improvement is still limited by extra circuits.



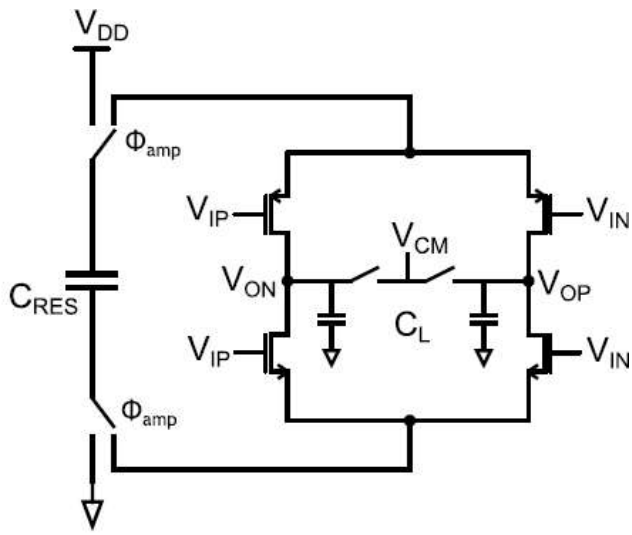
. The bi-directional dynamic comparator [30].

Figure 5-35 Bi-directional dynamic comparator

In the conventional dynamic pre-amplifier design, the loading capacitors are fully discharged despite the fact that only the initial portion contributes to the noise reduction [13], [14]. Bindra *et al.* [33] proposed a dynamically-biased pre-amplifier to address this issue. By providing a degeneration capacitor, VGS of the input pair keeps decreasing, and eventually, the input pair is cut off. It prevents fully discharging the load. In addition, the reduced VGS boosts the gm/ID during the pre-amplification phase, which increases gain and reduces noise. This work achieves over 2.5 times energy efficiency boost compared to its predecessor in [28].

Tang *et al.* [34] further improved the energy efficiency by introducing the floating inverter amplifier (FIA). As shown in Fig. 5.36, it adopts a CMOS input stage which naturally realizes current reuse. By powering the FIA with a floating reservoir capacitor, it provides a constant output

common-mode voltage, which elongates the pre-amplification time and increases the gain. It also avoids the common-mode discharge of the loading capacitor, thus saving considerable energy. In addition, the isolated power domain provided by the reservoir capacitor makes the amplification



The floating inverter amplifier [34].

Figure 5-36 FIA, the floating inverter preamplifier

insensitive of the input common-mode voltage and PVT variations. The reservoir capacitor also provides dynamic biasing similar to [33], which boosts the pre-amplifier gm/ID. Combining all these merits, it realizes a robust dynamic comparator with greater than 7 times energy efficiency improvement compared to that of the Strong-Arm latch.

COMPARATOR PRE-AMPLIFIER COMPARISON

Pre-amp Architecture	Speed	Area Efficiency	Pre-amp Gain	Energy Efficiency*	Sensitivity **
Conventional	Good	Good	Poor	Poor	High
Dynamic Biasing	Poor	Fair	Fair	Fair	High
Bi-directional	Fair	Fair	Poor	Fair	Medium
FIA	Poor	Poor	Good	Good	Low

* Pre-amp's thermal noise suppression in terms of power consumption
** Sensitivity against input CM/PVT variations

Table 5.1 [TANG *et al.*: LOW-POWER SAR ADC DESIGN: OVERVIEW AND SURVEY OF STATE-OF-ART TECHNIQUES]

Table 5.1 gives a qualitative evaluation of state-of-the-art comparator pre-amplifiers. The conventional pre-amplifier excels with great speed and simplicity and is still widely used. When

targeting energy efficient applications**, the FIA is a promising choice by offering high gain, great energy efficiency, and robustness simultaneously.**

To conclude, in the past years, various low-power dynamic comparators have been proposed. Here samples a few highlighted techniques:

• Pre-amplifier gain boosting techniques, such as double-tail comparators [27], [28] and cascaded input pairs [29].

• CMOS input pairs that double the pre-amplifier energy efficiency, such as bi-directional comparator [30] and FIA [34].

• Biasing techniques that boost gm/ID of the input pairs, such as dynamic biasing pre-amplifier [33] and FIA [34].

2) Low-Power Time-Domain Comparators:

In addition to the voltage-domain circuit optimizations, another promising direction is to perform the comparison in time-domain, which benefits from technology scaling with increased transistor speed and reduced power consumption. In the time-domain comparison, an input voltage signal is first converted to a time delay, which is then digitized using a 1-bit time-to-digital converter. The voltage to- time conversion can be implemented by amplifiers [36], voltage-controlled delay lines (VCDLs) [37], [39], [42], and voltage-controlled oscillators (VCOs) [38], [40]. But since we will not use any of these here, we will not delve into the details, go the suggested references for further knowledge about time domain comparators.

So, let's remain in the low-power voltage domain comparators, another promising comparator is the fast low leakage comparator proposed in [41], The comparator is shown in Fig. 5.37.

The first stage is a voltage amplification stage with INP and INN as differential input and FP and FN as differential output. The second stage contains both a simple voltage amplifier and a positive-feedback amplifier to obtain the rail-to-rail digital output SP and SN. To illustrate the functionality, Fig. 5.37 (right) contains a simulated comparison action. Prior to the comparison, the nodes have been pre-charged by a low clock voltage to the situation shown during the first 100 ps. A rising clock edge stops the pre-charging state and starts the amplification in the first stage. In the first stage, the differential output voltage increases while the common-mode output voltage decreases towards gnd. When the common-mode voltage approximates the threshold voltage of the input pair of the second stage, the voltage amplification in the second stage takes over. In the second stage, the differential output voltage increases while the common-mode output voltage increases. As the common-mode output voltage increases gradually the positive-feedback mechanism takes over and provides the rail-to-rail output.

The power dissipation in the first stage stops when its (parasitic) capacitances have been fully discharged. The power dissipation in the second stage stops when the positive-feedback amplifier has settled. When the clock signal becomes low again, all (parasitic) capacitances are pre-charged to their original values. Therefore, this comparator has virtually no power dissipation when the comparator is not active.

Especially during a critical comparison with small differential input signal, the positive-feedback amplifier in the second stage needs to provide a large gain for the required rail-to-rail output. To obtain this output within a reasonable time, it is helpful that the relevant devices are biased in strong inversion. This strong inversion is one reason that the positive-feedback amplifier is not energy efficient by itself. Another reason is that its noise performance is relatively constant during the amplification period, while the best noise performance is only required when the overall voltage gain is still low. The first stage of the two-stage comparator serves to improve the energy efficiency of the comparator. According to initial simulations based on initial sizing parameters, the (differential) voltage gain of the first stage at the moment the second stage takes over is approximately 5. This is a suitable gain because it both significantly reduces the input equivalent noise of the second stage, and it is a feasible gain for an energy-efficient amplifier. The factor of 5 in input-equivalent noise voltage corresponds to a factor of 25 in power and therefore the power dissipation in the second stage can be (up to) a factor 25 lower than without the first stage. The voltage amplification of 5 is also below the intrinsic voltage gain of a short-channel CMOS transistor and therefore no energy-inefficient measures to achieve a high gain are required. The first stage of the comparator can benefit from weak inversion operation of the input transistors to achieve a low equivalent input noise.

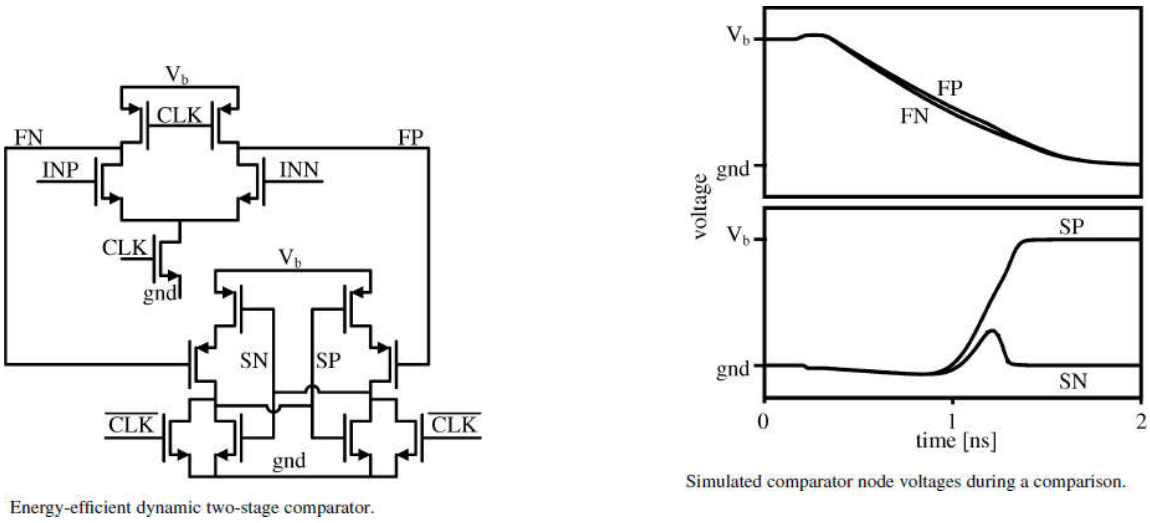


Figure 5-37 FLL, Fast-low-leakage comparator operation

So, considering all those prospects and the specs for our application, speed is not an issue, but power and capability of providing high resolution is key to out application, we chose two prospects that we will compare between, the FIA and the FLL, the FLL is the one used in the paper upon which we based this design so we will try the FIA, if it shows that it is incompetent we will go through with the FLL.

5.11.1 FIA

Three major specs will be addressed in this comparator and the following comparator, these specs are input referred noise less than LSB , offset due to mismatch must be as low as possible and power consumption less than 60 percent of the total power consumption. Speed is not the issue here because the application itself is slow paced.

As we have stated earlier, the FIA's best suit is reduced power consumption or maximum power efficiency, let's delve into some of the mathematical analysis for the comparator.

The pre-amplifier adopts an inverter-based input pair powered by a floating reservoir capacitor; **it realizes both current reuse and dynamic bias**, thereby significantly boosting gm/ID and reducing noise. Moreover, it greatly reduces the influence of the process corner and the input common-mode voltage on the comparator performance, including noise, offset, and delay.

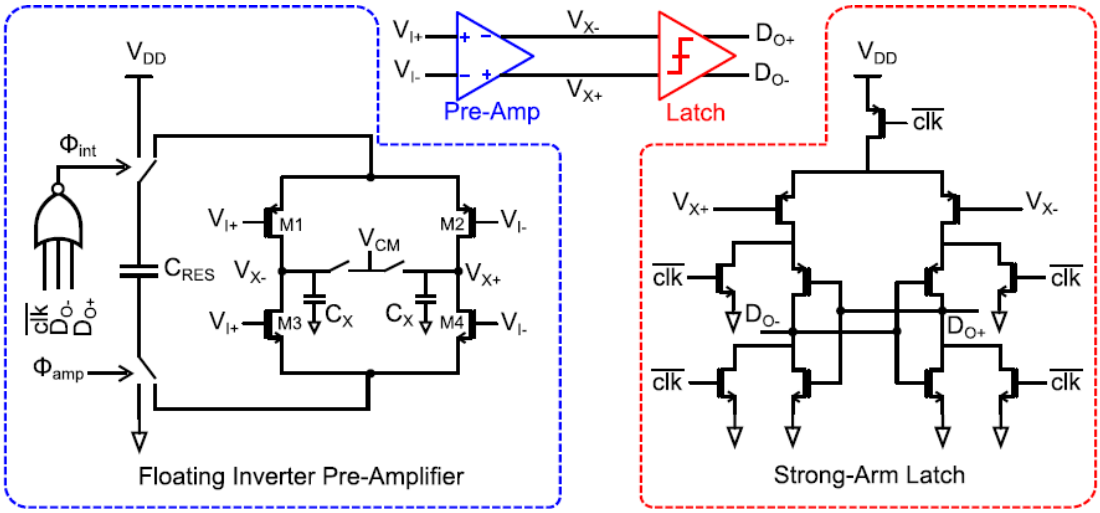


Figure 5-38 Schematic of the proposed FIA with SA latch
[TANG *et al.*: ENERGY-EFFICIENT COMPARATOR WITH DYNAMIC FIA]

As shown in Fig. 5.38 . The inverter-based input stage in the FIA naturally realizes current reuse. By powering the FIA with a floating reservoir capacitor, it provides an isolated power domain and makes the amplification independent of the input common-mode voltage. The FIA output common-mode voltage is kept constant, which elongates the integration time and increases the gain. In addition, the reservoir capacitor provides dynamic source degeneration that increases gm/ID and prevents the full discharge of the integration capacitors. Overall, the proposed comparator achieves over seven-time energy-efficiency improvement compared with the SA latch and provides input common-mode insensitivity.

Now we will tackle the assembly one at a time, starting the SA latch comparator, The conventional SA latch is shown in Fig. 5.34 . The operation can be divided into two phases, the integration phase and the latch regeneration phase, with the turn-on of the pMOS Cross coupled pair separating the two phases. During the integration phase, the comparator works as a dynamic integrator. The input signal is continuously integrated onto the integration capacitor CX. $M3/M4$ is gradually turned on during the integration process. With the output nodes dropping below (VDD$-V$THP), where VTHP is the threshold voltage of $M5/M6$, the comparator enters the latch phase. The positive feedback provides the exponentially growing gain and dominates the behavior of the comparator during this phase. Fig.5.39 shows a simplified model for a low-noise comparator, with large capacitors CX placed at the integration nodes. The CX integration process plays the dominant role in setting the comparator noise and power [2], [4], [7], [8] .

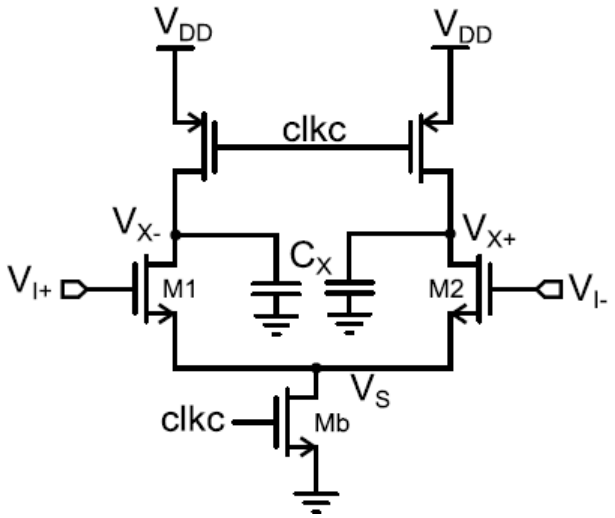


Figure 5-39

Now the FIA comparator will be discussed over two phases :

CMOS DB Integration Pre-Amplifier

To improve the energy efficiency of the pre-amplifier further, the CMOS DB integration is proposed in this section [42] . A differential-mode integration model, including a CMOS input pair powered by two tail capacitors, is shown in Fig. 5.40. During the integration phase, the bottom source node $VS-$ increases, while the upper one $VS+$ decreases. It results in the decreased overdrive voltage of both nMOS and pMOS input pairs. Let Gm = gm,n + gm,p represents the overall transconductance of the pre-amplifier, where gm,n and gm,p are the transconductance of the nMOS and pMOS input pairs, respectively. As shown in Fig. 5.40 (right), at the beginning of the dynamic integration phase Øint, the Gm/ID of the pre-amplifier is twice the one in the SA latch thanks to the current reuse. Throughout the integration, it boosts the average Gm/ID of the pre-amplifier by more than a staggering 2.5 times, leading to a large improvement in the power efficiency.

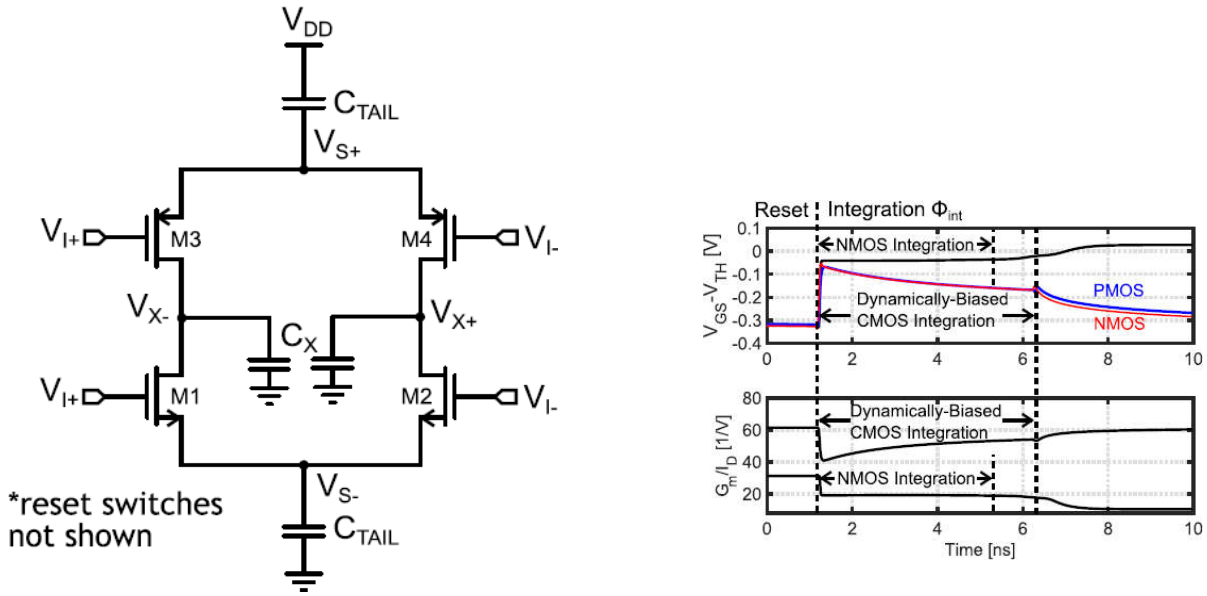


Figure 5-40

In addition, during the integration phase, only the differential charge is integrated on the loading capacitors, and the common-mode voltage stays constant, which is 0.6 V with a 1.2-V supply. It prevents the full discharge of CX and removes the bounded common-mode drop limitation for the pre-amplifier gain.

However, a caveat in this simple CMOS integration solution is the input common-mode and process corner sensitivity due to the lack of output common-mode feedback (CMFB). To ensure the CMOS integration functionality, the currents flowing through the pMOS and nMOS input pairs should be equal.

To solve this problem and build a robust dynamic integrator against process corner and input common-mode variations, the FIA architecture is proposed, as shown in Fig. 5.41 (a). By merging the two tail capacitors CTAIL into single floating reservoir capacitor CRES, it not only reduces the total capacitance size by 75% since CRES = CTAIL/2 but also provides an isolated voltage domain for the pre-amplifier. A common-mode equivalent is shown in Fig. 5.42(b) . Since the input and output currents from CRES must be equal (IAMP+ = IAMP−), the common-mode current flowing into the integration capacitor IX,CM is forced to be 0, thus achieving a constant output common-mode voltage without a dedicated CMFB circuit.

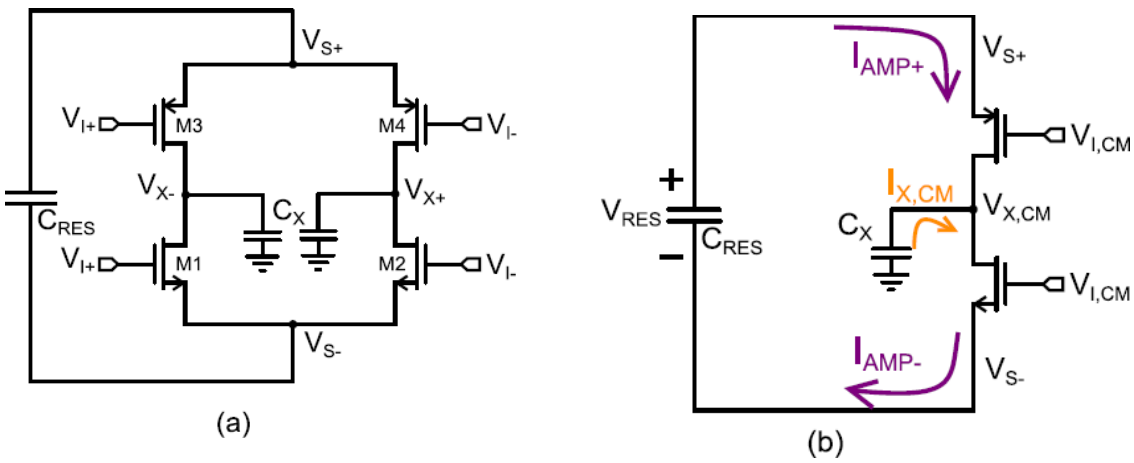


Figure 5-41

In Figure 5-42, we can see FIA behavioral simulation with (a) process corner variation and (b) input common-mode voltage variation, we can see how robust the system is against input common mode variations and process corners and variations, notice that the paper was operating in 1.2v supply voltage, in our case it is 0.6v.

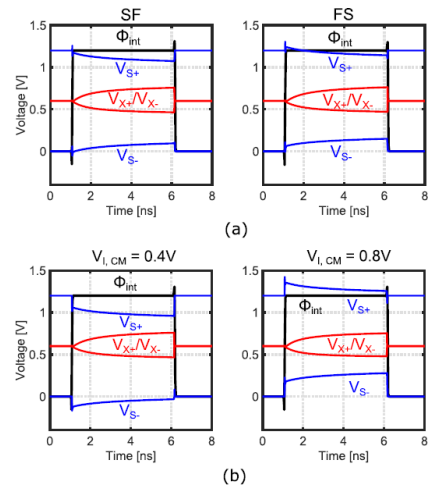


Figure 5-42

From the proposing paper [42], mathematical analysis was made for the input referred noise using the following equation,

$$\sigma^2_{in,\mathrm{FIA}}(T_{\mathrm{INT}}) = \frac{2nkT}{C_{\mathrm{RES}} \cdot \Delta V_S(T_{\mathrm{INT}})} \cdot \frac{I_D}{G_m}. \qquad (5.9)$$

Where n is the subthreshold slope factor $n = \frac{Cox+Cdep}{Cox} \approx 1.2 - 1.5$, choosing Gm/id initially to be 15 which is decent starting point, σin $\leq 1\ LSB \approx 300uv$, regarding the graphs from the paper we found that $\Delta VS(Tint) \approx 10\%\ of\ VS(intial) = 0.6 * 0.1 = 60mv$ which fits with our simulations

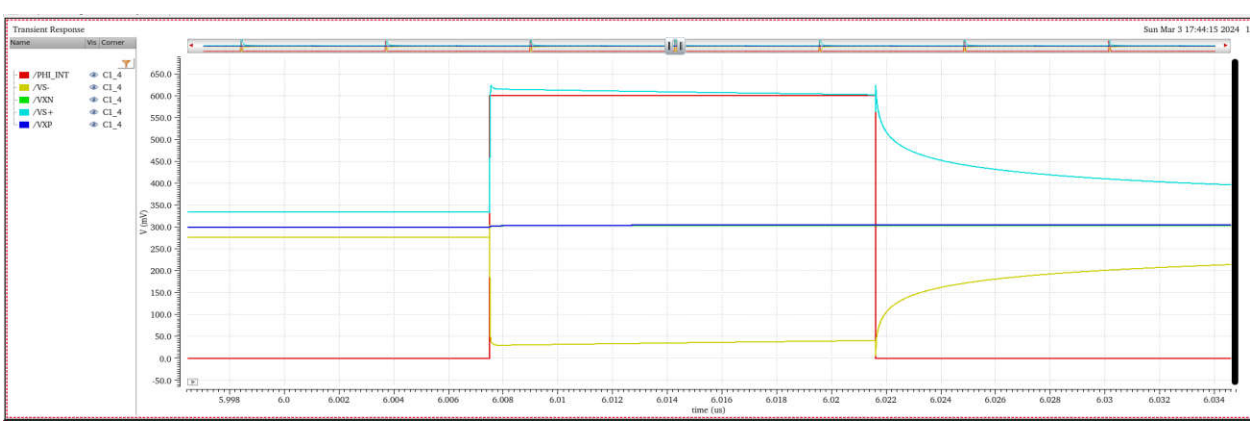


Figure 5-42, indicating that ΔVs is indeed about 60mv give or take.

So, Cres ≈ 150fF.

And from the gain relation,

$$A_V(T_{\text{INT}}) = \frac{\Delta V_{X,\text{DM}}(T_{\text{INT}})}{\Delta V_{I,\text{DM}}} = \frac{2 \cdot C_{\text{RES}} \cdot \Delta V_S(T_{\text{INT}})}{n \cdot C_X \cdot U_T}. \tag{5.10}$$

Assuming a preamp gain of 20 which seems to be a good choice, Ut (thermal voltage of 26mv), then Cx can be about 25 fF.

Now showing the proposed schematic :

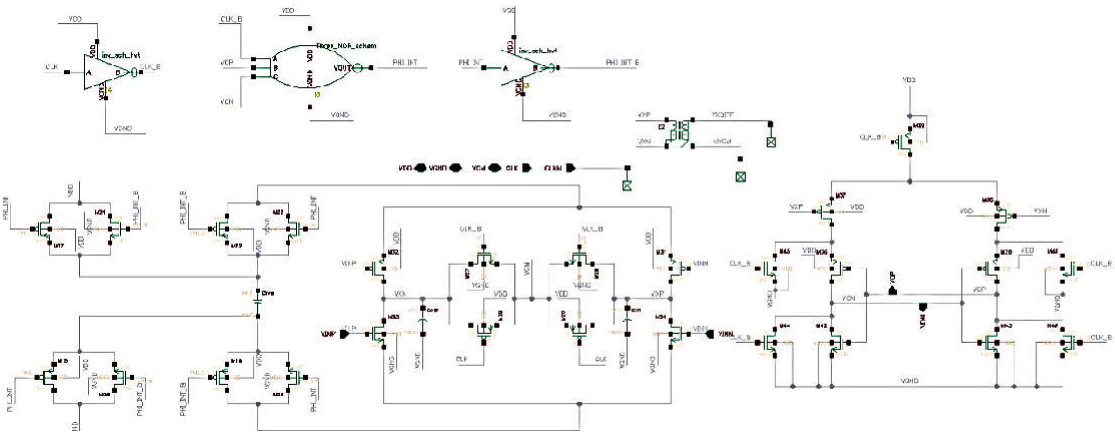


Figure 5-43, FIA schematic

We did some test for that comparator to check all of its merits :

1- Overdrive recovery test for the hysteresis :

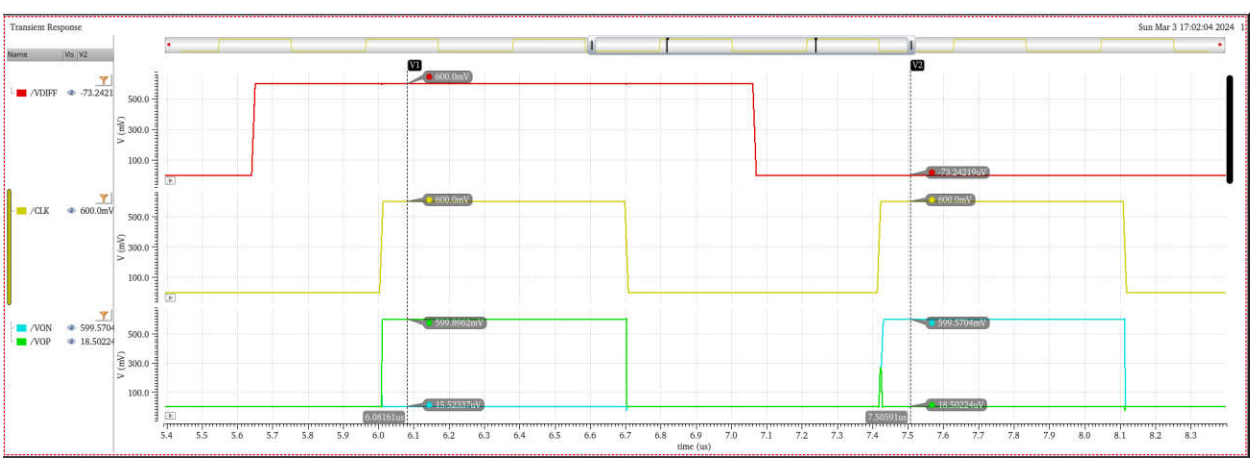


Figure 5-44, Passed overdrive test

2- Input common mode sensitivity test :

Case 1 : at VICM = 300mv (no variations)

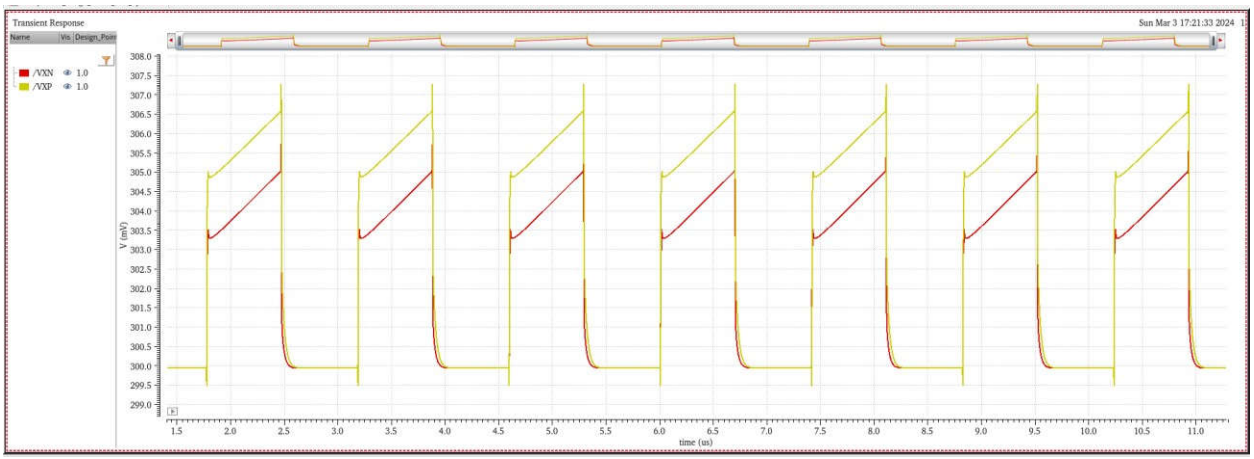


Figure 5-45, Variations on Vx nodes are so small indicating very high-power efficiency.

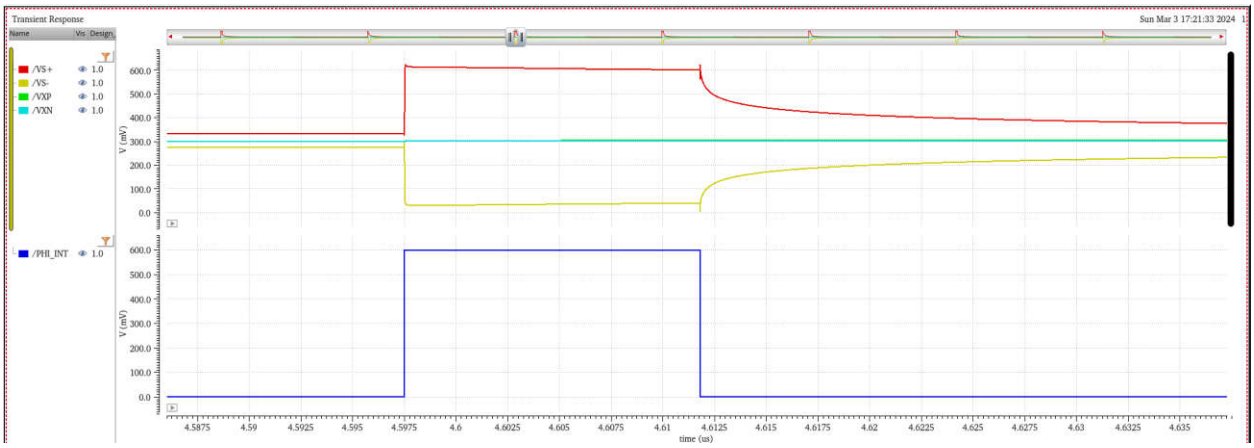


Figure 5-45(b), As we can see here, during PHI_INT active, the residue cap is connected to the preamplifier but as we can see, VS+ and VS- do not lose much charges during that period, which is the required, also the variation in VX+ and VX- is negligible providing high efficiency, but after PHI_INT is off, VS+ starts to degrade and VS+ starts to charge, and the current carried by the input pair starts to decrease gradually and thus CX does not discharge entirely as required.

Case 2 : at VICM = 200mv

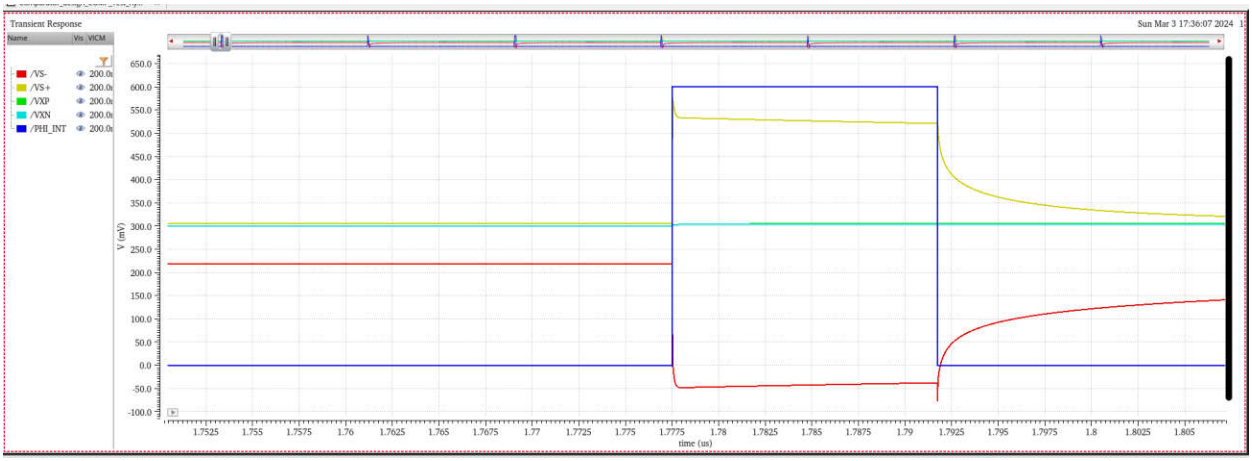


Figure 5-46 As we can see, VS+ and VS- were affected as their values shifted downwards but VX+ and VX- remain as they were around VCM = 0.3v.

Case 3 : at VICM = 400mv

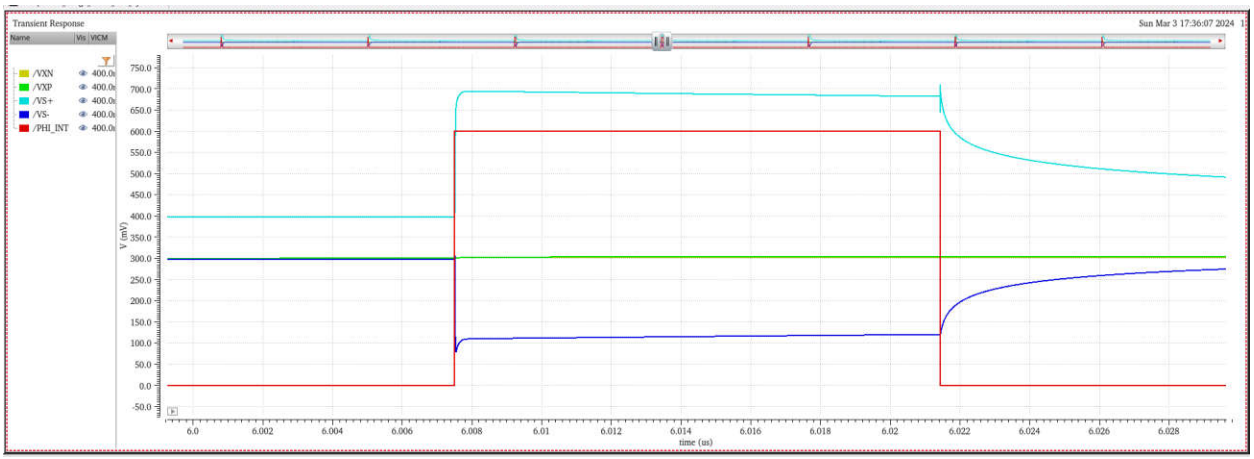


Figure 5-47 As we can see, VS+ and VS- were affected as their values shifted upwards but VX+ and VX- remain as they were around VCM = 0.3v.

We have also run the ss,ff,fs,sf corners on the comparator and turns out it does satisfy all those merits, later on we tested its power consumption, and it was relatively good, we were able to size it to get good noise and offset requirements.

But when we integrated it with the system it degraded the ENOB, which is the main performance metric of the resolution in the SAR ADC system, upon careful study and debugging, we found that the large sizes of the input pair of the Preamp introduced large kickback noise on the sensitive input pair connected to the DAC, and since we have two NMOS and two PMOS connected to the input pair, and since their sizes is large in order to maintain the good noise and offset spec, this caused the ENOB not to increase over 10 bit in the ideal testing conditions with no noise, the only way was to create kickback recovery mechanism which would consume large power, power that we do not have, aside from that it had a large Cres which increased with design and simulation up to 250fF and CX up to 50fF, these will consume huge area in the Layout phase as we will see later.

This made us think that the FIA will not be suitable for our 12-bit mode, 512fF DAC highly sensitive SAR ADC, and drove us to our next and final candidate which is the Fast low leakage comparator.

5.11.2 FLL initial design

Now, we will discuss the Fast low leakage comparator [41] and its initial design point and then later on using simulation we will reach the optimal design point.

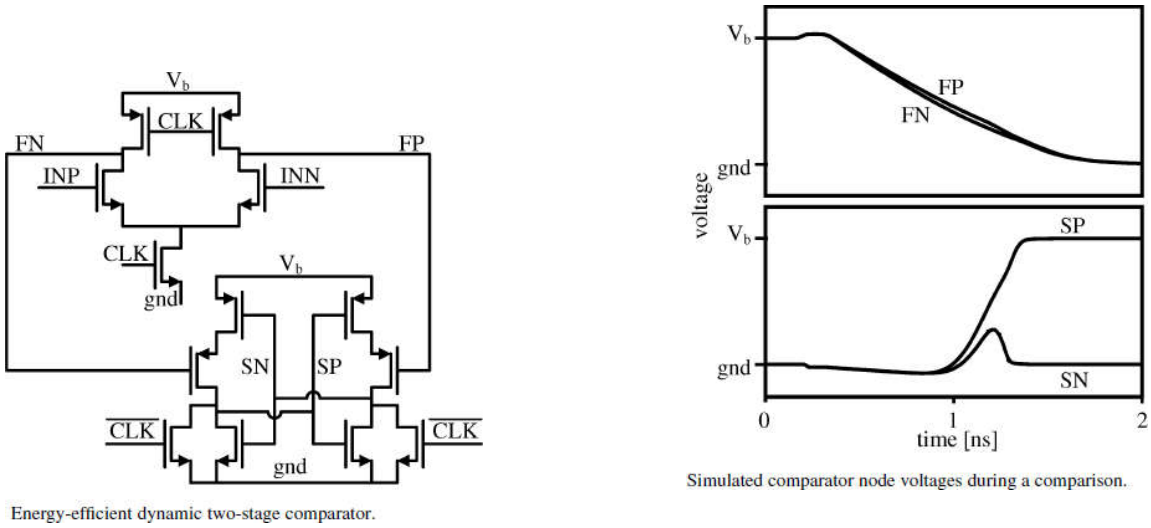


Figure 5-48 FLL, Fast-low-leakage comparator operation

The flow of operation of the FLL was discussed in the first part of section 5-11, here we will follow with some of the equation that describe the performance of such a comparator.

Because of the significant voltage amplification in the first stage, the noise and therefore the energy efficiency of the comparator depend largely on the first stage. The dominant noise source in the first stage is its input pair. The thermal noise of a single MOS transistor can be estimated with an equivalent noise resistor Rn,MOS,

$$R_{n,MOST} \approx \frac{\gamma}{gm_{\mathrm{MOST}}} \tag{5.11}$$

where the noise excess factor γ is close to 1. The equivalent noise resistor of each input transistor appears in series with the DAC voltage. The equivalent noise resistor of the input pair Rn,FS is approximately,

$$R_{n,FS} \approx \frac{2}{gm_{\mathrm{MOST}}}. \tag{5.12}$$

The first stage of the comparator simultaneously integrates the input signal and the noise from the equivalent noise resistor. The input-equivalent integrated noise can be estimated by,

$$\sigma_{\mathrm{V}} \approx \sqrt{4 \cdot k \cdot T \cdot R_{n,FS} \cdot \mathrm{NBW}} \approx \sqrt{4 \cdot k \cdot T \cdot \frac{2}{gm_{\mathrm{MOST}}} \cdot \mathrm{NBW}} \tag{5.13}$$

The noise bandwidth (NBW) for noise integrated over integration time Tint is,

$$\mathrm{NBW} = \frac{1}{2 \cdot T_{\mathrm{int}}}. \tag{5.14}$$

The second stage samples the output voltage of the first stage when the common mode voltage on nodes FP and FN has changed by approximately the threshold voltage Vthreshold of its input pair. The current through each input transistor in the first stage Imos is integrated on the (parasitic) capacitances CFP and CFN at nodes FP and FN, the integration time Tint can be approximated by,

$$T_{\text{int}} \approx \frac{V_{\text{threshold}} \cdot C_{\text{FP}}}{I_{\text{MOST}}} \approx \frac{V_{\text{threshold}} \cdot C_{\text{FN}}}{I_{\text{MOST}}}. \tag{5.15}$$

After inserting (5.15) and (5.14) into (5.13) we get :

$$\sigma_{\text{v}} \approx \sqrt{4 \cdot k \cdot T \cdot \frac{1}{V_{\text{threshold}} \cdot C_{\text{FP}}} \cdot \frac{I_{\text{MOST}}}{gm_{\text{MOST}}}}. \tag{5.16}$$

For an energy-efficient amplification, the input pair is biased in weak inversion. The transconductance of a MOS transistor in weak inversion depends on the subthreshold slope. According to simulations for the 65 nm process that was used the transconductance can be approximated by :

$$gm_{\text{MOST}} \approx \frac{I_{\text{MOST}}}{2 \cdot V_{\text{thermal}}} \tag{5.17}$$

where V_{thermal} is

$$V_{\text{thermal}} = \frac{k \cdot T}{q}.$$

(5.18)

In a SAR ADC based on a charge-redistribution DAC, the thermal noise from the DAC switches sampled by the comparator can potentially also be significant. The equivalent noise resistor of the DAC Rn,DAC is determined by the on-resistances of the switches. The first stage of the comparator is relatively slow because it is biased in weak inversion. This makes the NBW relatively small and at the same time its equivalent noise resistor Rn,FS relatively large, therefore

$$R_{\text{n,FS}} \gg R_{\text{n,DAC}}. \tag{5.19}$$

As a consequence, Rn,DAC contributes relatively little noise, by inserting (5.17) into (5.16) we get:

$$\sigma_{\mathrm{v}} \approx \sqrt{\frac{k \cdot T}{C_{\mathrm{FP}}}} \cdot \sqrt{8 \cdot \frac{V_{\mathrm{thermal}}}{V_{\mathrm{threshold}}}}. \tag{5.20}$$

And thus, to be able to get the required values of CFP and CFN we will re-organize the equation :

$$C_{\mathrm{FP}} = C_{\mathrm{FN}} \geq \frac{k \cdot T}{\sigma_{\mathrm{v}}^2} \cdot 8 \cdot \frac{V_{\mathrm{thermal}}}{V_{\mathrm{threshold}}}. \tag{5.21}$$

A level of 0.3 mV is suitable because it puts the thermal noise of the comparator in the same order as the quantization noise of this ADC. In this comparator CFP and CFN mostly consist of parasitic MOST capacitances and are somewhat larger than 10 fF each. The energy dissipation of the first stage depends mostly on CFP and CFN being discharged and charged for every comparison, through simulation now we will reconfigure this comparator.

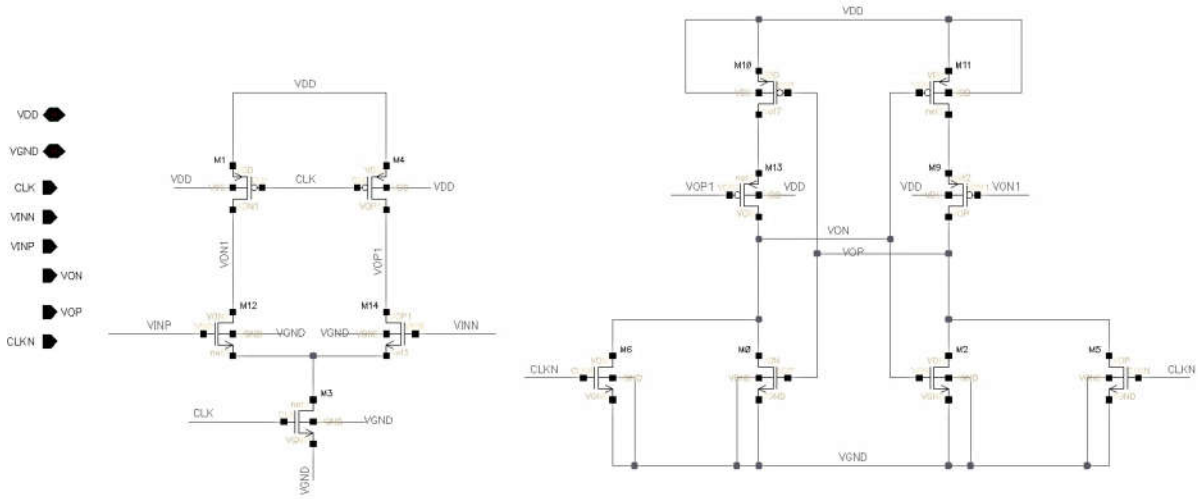


Figure 5-49 Actual schematic of the FLL comparator

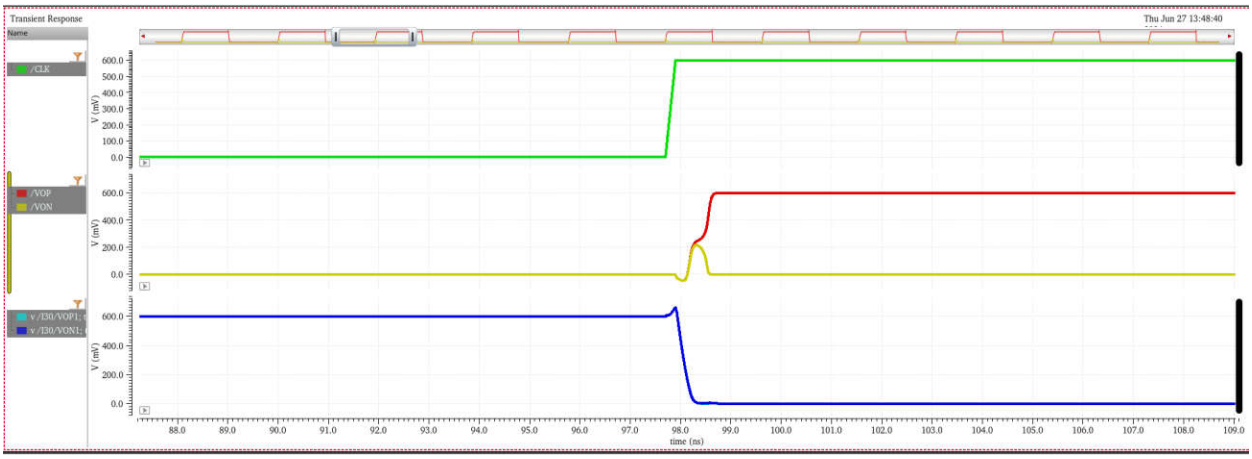


Figure 5-50 Output of both the Latch and the PREAMP

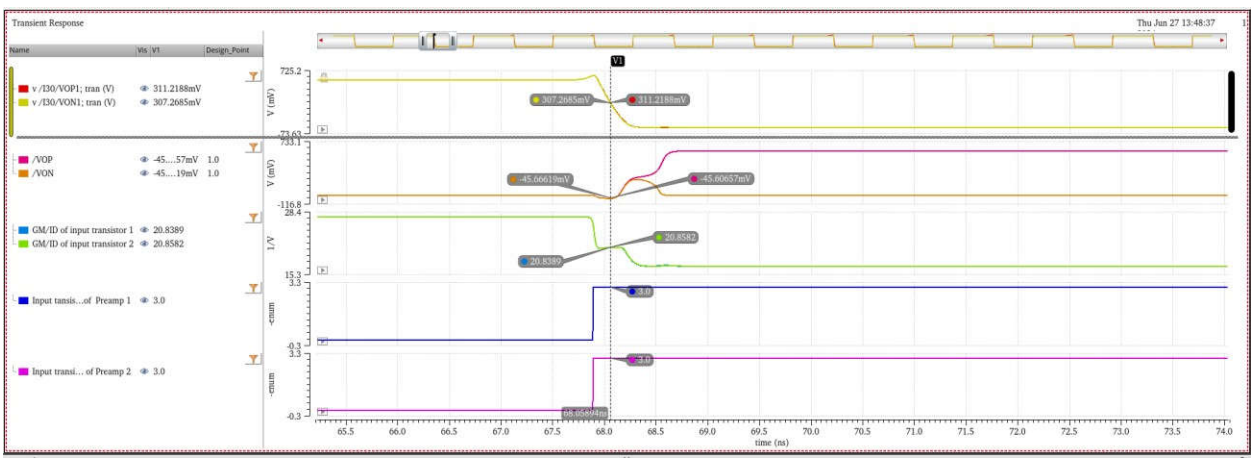


Figure 5-51 Region of operation of the input pair of the PREAMP

As we can see, the input pair is in weak (subthreshold) region, which is desired, in order to reduce power consumption, improve noise performance and increase the gain and many other factors as discussed earlier. Also, observe that the input pair is in cutoff when the clock is Low which means that the comparator is inactive, and that there is no active power consumption in this part.

The reason why Gm/ID is decreasing during the integration region, is that the voltage on VS (the source of the input pairs) is decreasing as well, and since the gate voltage is constant, Then VGS increases leading to decreasing the GM/ID, where the current is increasing until CFP and CFN are completely discharged, and the current goes to a fixed value causing the Gm/ID to saturate at certain value.

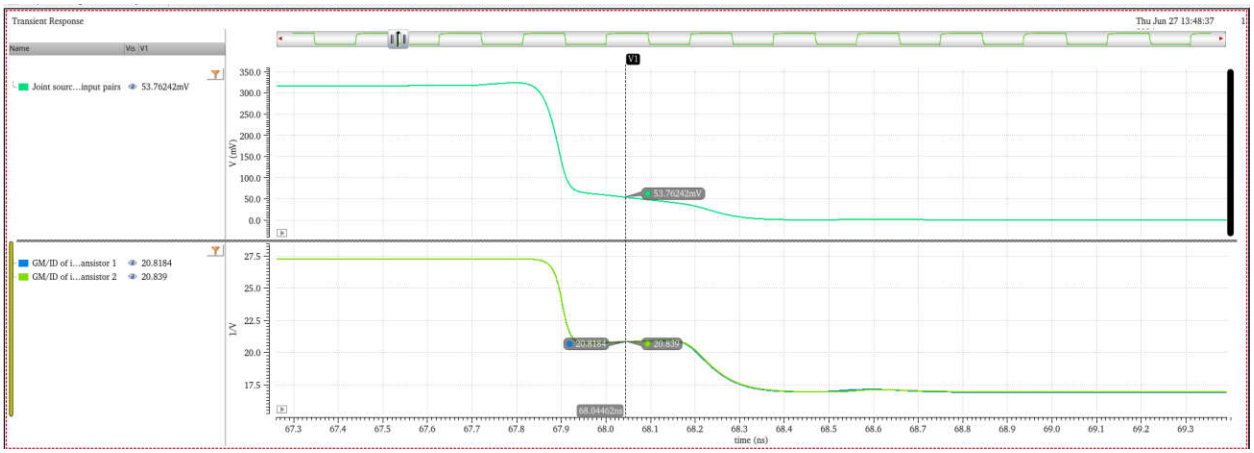


Figure 5-52 Source voltage and GM/ID of the input pairs

Now let's describe the peak at VOP1 and VON1, and also the DROP at VOP and VON :

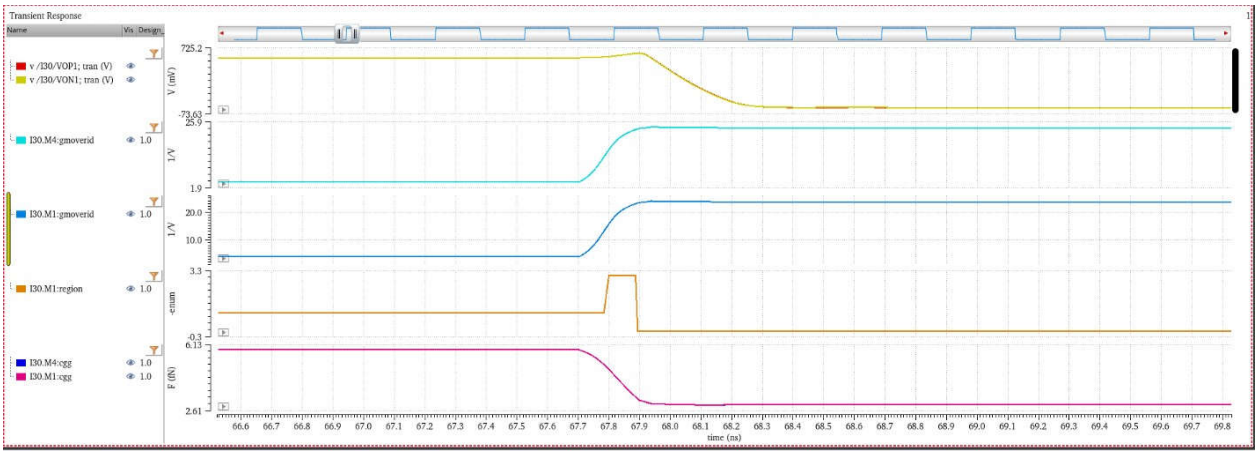


Figure 5-53 Parasitic caps on PMOS load transistors with the GM/ID

As we can see the PMOS load transistors, change from the saturation region to subthreshold then cut off region, during these transitions the parasitic caps on these transistors that were initially charged start to decrease, but since they were carrying charges, and according to the law of conservation of charge, these charges must go somewhere and thus go for the caps on VOP1 and VON1 creating that overshoot in voltage.

As for the DROP on VON and VOP, the exact reverse is done, at first the reset switches were ON, M0,2,13,9 were OFF, and their caps were not charged, once the Clock is activated these transistors start operating one by one, and their parasitic caps that were previously discharged now need to be charged creating that little drop on the nodes.

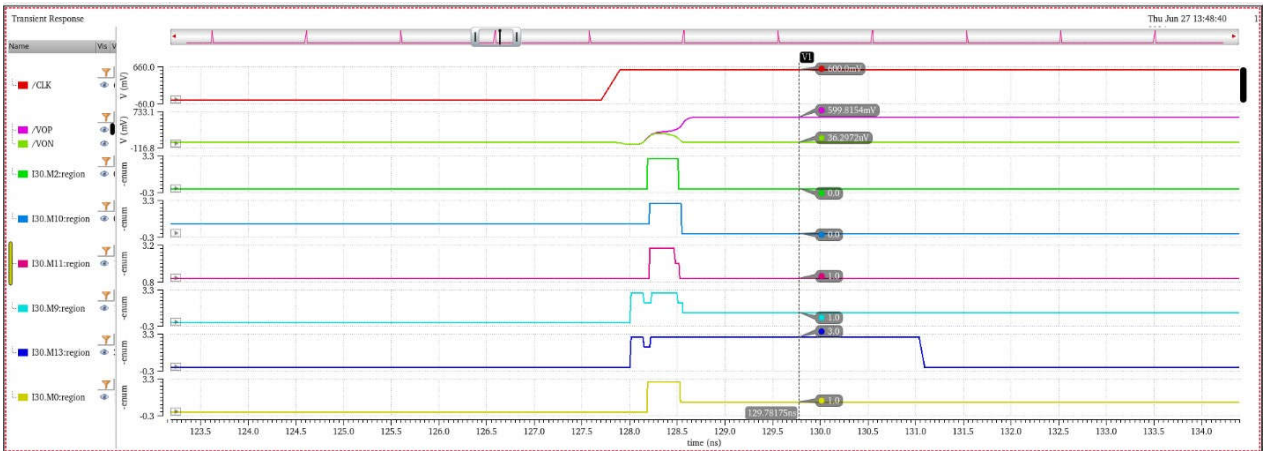


Figure 5-54 Regions of operation for different transistors in Latch

Since we have the regions of operation plotted, let's discuss what exactly happens in that comparator, at first, the clock is Low, which is the precharge phase, VOP1 and VON1 are pre-charged to VDD while VON and VOP are discharged to the ground.

When the clock goes HIGH, the 2nd phase starts which is the amplification phase or the integration phase, mainly driven by the Preamp, where the difference on the input of the preamp is integrated on the preamp output nodes, when the clock is HIGH, the tail current source is operational and current flows causing the discharge of the CFP and CFN, they are not discharged at the same rate because VINP is not equal to VINN, the output common mode voltage starts to decrease until it discharges completely (different from the FIA).

At some instant during that discharge the output common mode voltage decreases to a point where VSG of the Latch input pair is more than their threshold voltage, giving rise to a very small current and putting these input pairs in weak inversion. The current starts to increase and brings the input pair of the LATCH to strong inversion for some time, during which the Latch regeneration transistors are still off because VON and VOP have not yet overcome their threshold voltages, once VON and VOP exceeds the limit of the threshold voltage, at this point VOP and VON are not exactly similar, there is difference between them resulting from the input differential at the preamp, this brings us to third and final phase which is latch regeneration phase, M0 and M2 both enter the weak inversion and limit the current through the input pairs putting them back to weak inversion again.

Since VOP is initially larger than VON, then the current through M0 (responsible for the discharge of VON)is more, pushing it to triode , and through M10 (which would bring more voltage)is less, pushing it to cutoff and this cause VON to decrease at a fast rate, this decrease in VON cause the current through M2 to decrease pushing it to cutoff, and the current through M11

to increase keeping it in triode and thus VOP keeps increasing and VON keeps decreasing which holds the latch operation and provides the rail to rail output.

Now we need to identify the main specs in the comparator that will help us size the comparator, these specs are the input referred noise which is supposed to be less than 1 LSB = 300uv and the offset must be as low as possible also the power consumption should be less than 60% of the total power consumption.

5.12 Noise analysis and separate testbenches

There are two methods to test the noise of a dynamic comparator and we have implemented both of them, first is the periodic PSS + pnoise analysis which is not very accurate but is very fast, which helps in sizing and doing multiple runs, the second one is the thermal noise analysis which is more accurate but takes a long time, we will discuss both of them and project their results.

5.12.1 PSS+ Pnoise analysis + PXF

We have referred to multiple papers in order to set up this testbench in order to calculate the input referred noise of the dynamic comparator, referring to [43], The PNOISE is similar to a conventional noise simulation, but includes frequency conversion effects, making it useful for predicting the behavior of dynamic circuits [44]. It takes the periodic operating point computed by the PSS analysis, linearizes the circuit and accounts for the effects of varying operating points on the noise contributions of the devices. The PNOISE presents the output noise in the form of Power Spectral Density (PSD). Taking this PSD, we find that the total output noise power, $\sigma_{n,out}$, at a time of observation tobs, is given by:

$$\sigma_{n,o}(t_{obs}) = \sqrt{\int_0^{1/2T_b} \mathrm{PSD}(f, t_{obs})\,\mathrm{d}f}, \tag{5.22}$$

where Tb is the beat period of the PSS response. Note that the integration is made from 0 to 1/2Tb, as the analysis outputs a single-sided PSD [45]. Since we are interested in the IRN of the system, $\sigma_{n,in}$ the total output noise in (5.22) has to be divided by its gain G:

$$\sigma_{n,i}(t_{\text{obs}}) = \frac{\sigma_{\text{n,o}}(t_{\text{obs}})}{G} \tag{5.23}$$

The gain of the linearized circuit G can be found by using periodic AC (PAC) or PXF analyses. Both approaches compute the transfer function of a circuit in order to a signal source, taking into account frequency translation effects. The PXF simultaneously computes transfer functions from every source in a circuit to a given output [44], allowing to obtain several metrics in a circuit in a single analysis (e.g. gain, power supply rejection ratio, among others).

A key aspect in PXF and PNOISE analyses is the choice of the time of observation t_{obs}, determining at which stage of the comparator operation the circuit noise will be sampled, which in turn affects the computed transfer functions and noise models returned by the PXF and PNOISE analyses, respectively. In [46], we see that the comparator has a constant signal-to-noise ratio (SNR) in the regeneration phase, in which the comparator begins to resolve the output. Prior and subsequent phases correspond either to a reset state or to a fully resolved output, respectively, where the SNR has no influence in the decision-making process. Hence, a range of values for tobs within the regeneration phase is acceptable. In [45] the choice of t_{obs} is done by calculating the gain and noise when a specific trigger signal crosses a threshold level. In this case the triggering event is defined by the crossing of the differential output by VDD/2, which is a convenient value and corresponds to a t_{obs} within the regeneration phase, leading to accurate results for most cases. In Fig. 2 the testbench for measuring the comparator noise with the PSS + PNOISE approach is shown, together with the PSS, PNOISE and PXF setup for the SPECTRE simulator.

```
pss pss fund=1/Tb harms=10 errpreset=conservative

pnoise pnoise start=1 stop=1/(2*Tb)
+     pnoisemethod=fullspectrum maxsideband=10
+     noisetype=sampled measurement=[pm0] annotate=status
  pm0 jitterevent tigger=[op on] triggerthresh=0.6 triggerdir=rise
      target=[op on]

pxf ( op on op on ) pxf maxsamples=1 crossingdirection=rise
+  thresholdvalue=0.55 ptvtype=sampled start=1 maxsideband=10
+  annotate=status
```

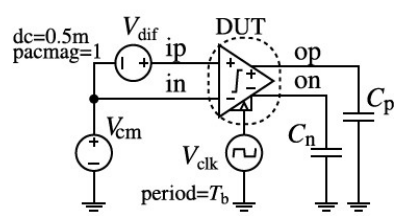


Figure 5-55 PSS + PNOISE + PXF setup and proposed testbench for noise measurement in SPECTRE 19.1 [João Silva 2021] DUT : device under test

In Figure 5-56, we can see the proposed testbenches for the noise analysis, as we stated the choice for t_{obs} is when the VODIFF crosses VDD/2, and the periodic integration of noise is from 0 to 1/2*Tb, there is no need for it be a very large value, since the test here is applying periodic noise, so all the noise will be aliased back in the first Nyquist zone, so all we need to do is calculate the output referred noise integrated under the first Nyquist zone, and then divide it by the gain obtained from the PXF or the PAC analysis.

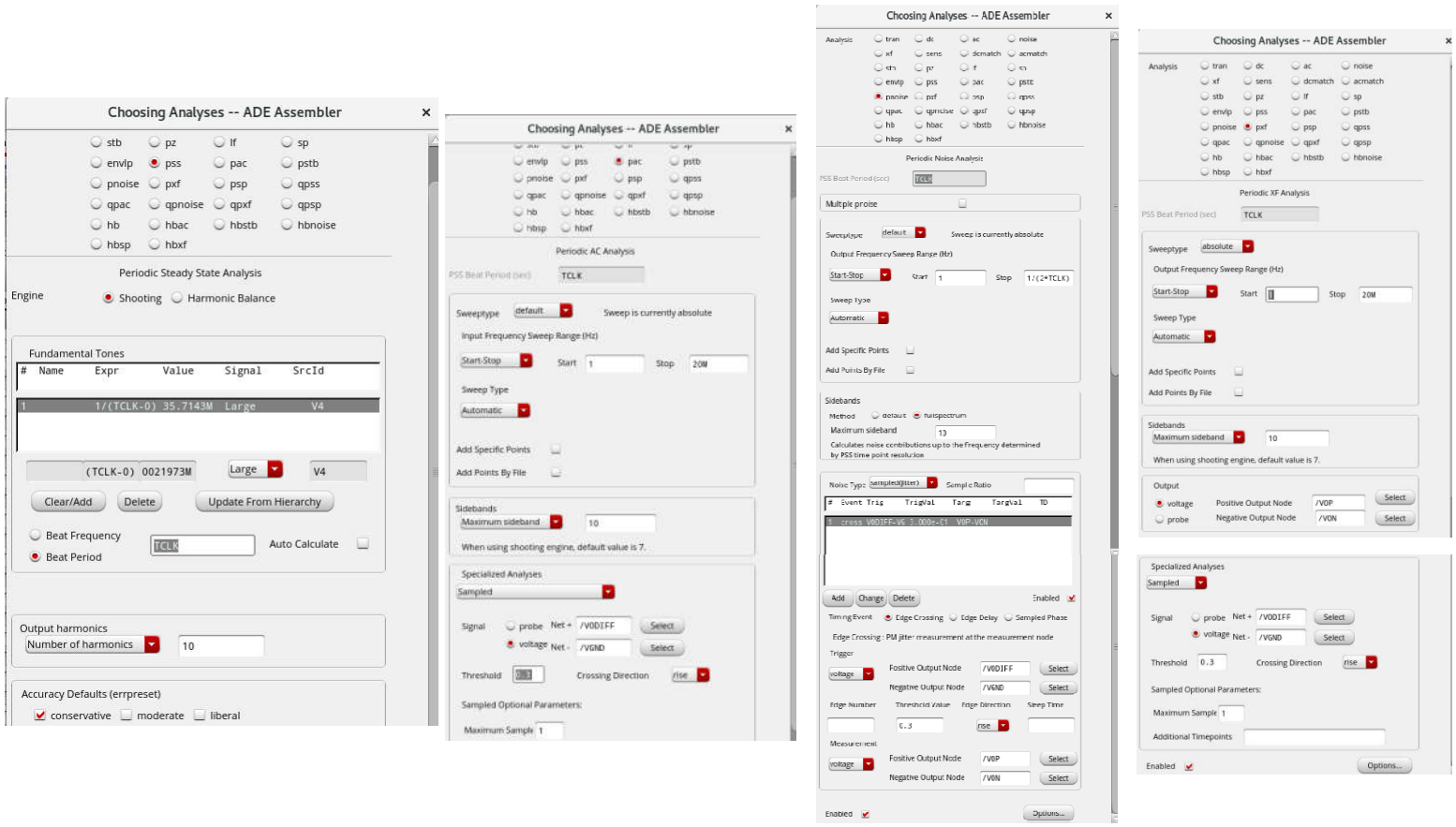


Figure 5-56 PSS + PAC + PNOISE + PXF setup and proposed testbench for noise measurement in SPECTRE

We went through many stages of sizing of the preamp and the latch in order to obtain the required spec on IRN to be less than LSB, using the same concepts discussed earlier about the IRN relation with gain, gm/id and the region of operation , etc.

And these are the results of the periodic noise test after multiple times of re-sizing :

Figure 5-57 Preamp gain measured by PAC analysis equals 5.4 and constant across frequencies, this value is considered good

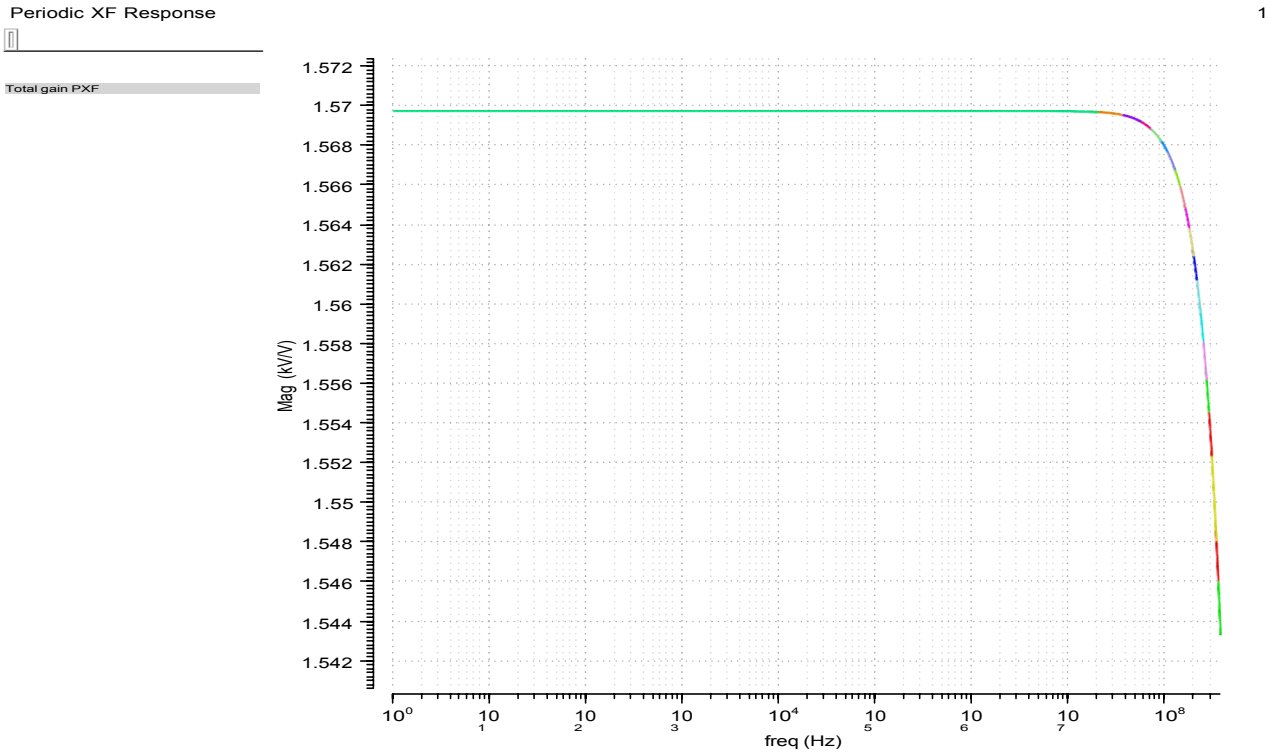


Figure 5-58 Total comparator gain is good 1.57 KV/V, so as we can see the latch is providing large gain, which contributes to the speed of the latch and its resolution capabilities.

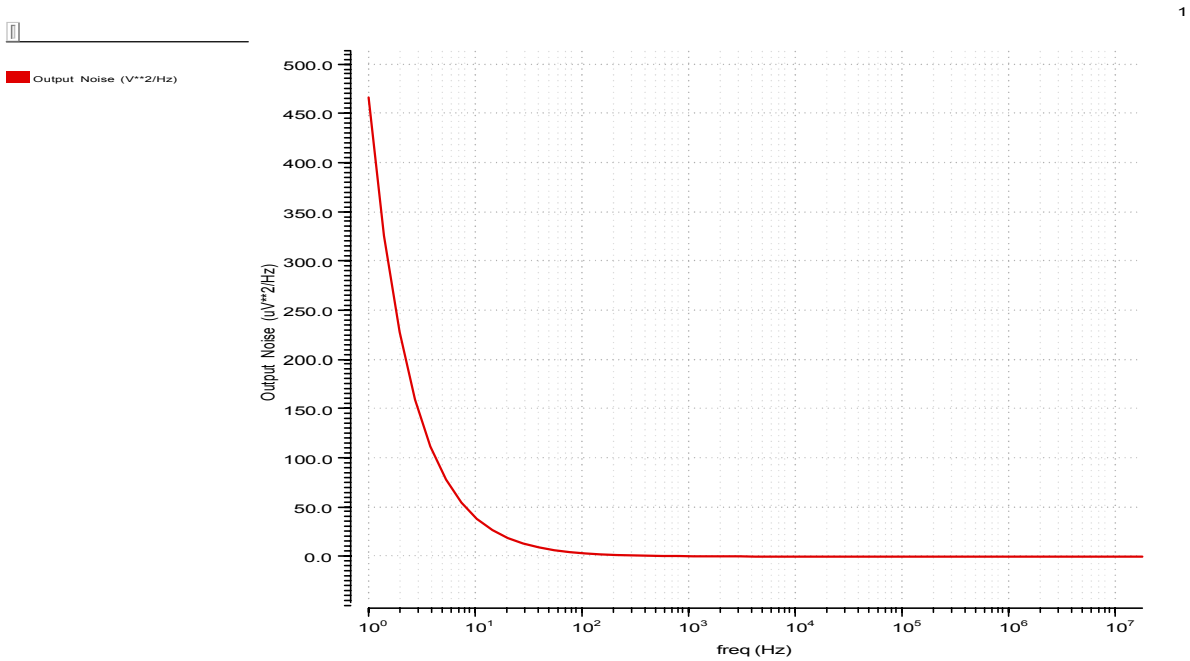


Figure 5-59 PSD of the output referred noise of the comparator

When we integrate from 0 to 1/2*Tb, we find that the total integrated output noise is about 632mv. Upon dividing that by the gain we find that the IRN = 402.5uv, which is still more than LSB, but it provides good results.

See, this will limit the resolution to $\frac{1.2(total\ dyn.range)}{402.5uv} = 2^{ENOB}$,giving ENOB = 11.54

After numerous iterations of cycles, we were not able to exceed this limit because we will have to increase the width of the of the PREAMP transistors, where at constant current (forced by the tail current source), the Vov will have to decrease and thus the GM/ID will increase, which will improve the noise performance, but at the same time increasing the width of the transistors will increase the gate leakage (secondary effect), and will increase the parasitics at the preamp output nodes which discharges every time completely causing a large increase in power consumption and AREA which both are the main concern here. So there had to be a sweet point, a point where a good tradeoff can be achieved, and we think it is this point.

Just keep in mind that this is not entirely accurate, and we will have to wait for the thermal noise analysis to make sure of that.

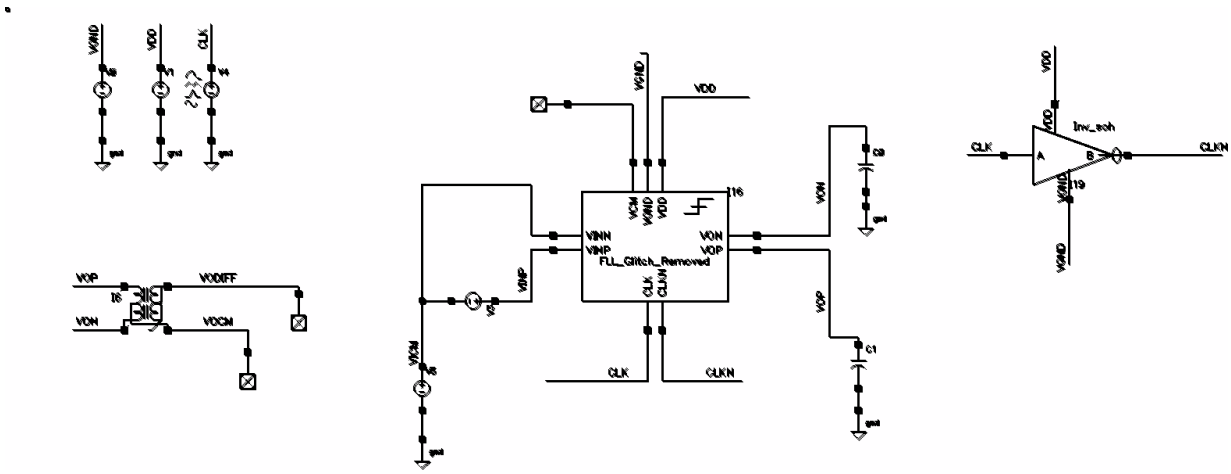


Figure 5-60 Testbench for the periodic noise analysis

5.12.2 Thermal noise analysis

The thermal noise analysis is the more accurate method for calculation of the IRN, referring to the tutorials of analog mind [47], The standard method of computing the output noise and dividing it by the gain does not apply to comparators because they produce a digital output. As explained in [48], we perform a transient noise simulation so that the comparator's time-domain decision is randomly affected by the noise of its constituent devices. We first set *V*in1 - *V*in2 to zero [Figure 5.61(a)] and ensure that the logical output assumes a value of zero or one with equal probabilities. Plotted in Figure 5.62(a) are *VX* and *VY*, in this case, for 100 clock cycles. We observe that *VX* goes to zero approximately 50 times.

Next, we select a small, constant value for abs(*V*in1 - *V*in2) so as to skew the decisions [see Figure 5.61(b)]. We recall that the area under a Gaussian distribution from $-\sigma$ to $+\sigma$ is equal to 68% and hence that from $-\infty$ to $-\sigma$ is 100% - (34% + 50%) = 16%. Thus, if *VS* is chosen so as to reduce the probability of zeros to 16%, then $VS = \sigma$, which is also the total root-mean square (rms) noise referred to the input. After a few iterations, we observe the waveforms in Figure 5.62(b), where *VX* goes to zero roughly 16 times for *VS* = 0.31mV. The comparator's input rms noise is approximately equal to this value. For greater precision, we can run the two simulations for a larger

number of clock cycles. So, it is either the 16% percent method, or the one mentioned in sec. 5.7 where the voltage from 50% to 90% of the probability of ones is equal to 1.28 σ.

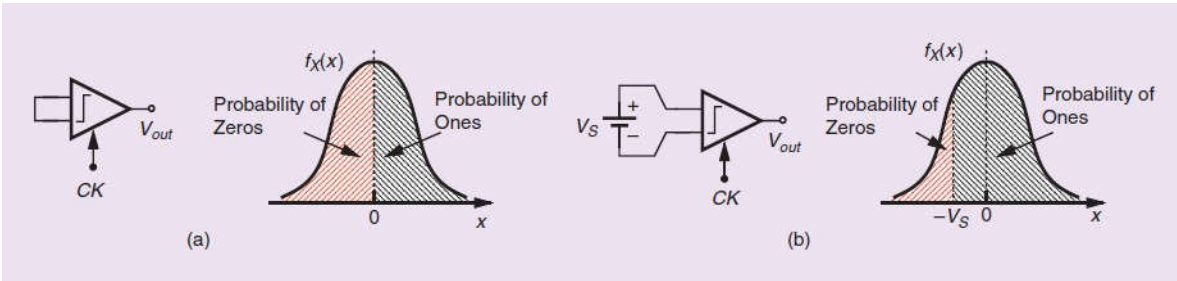


Figure 5-61 (a) A perfectly balanced comparator generates ones and zeros with equal probabilities; (b) a finite input imbalance skews the decisions.

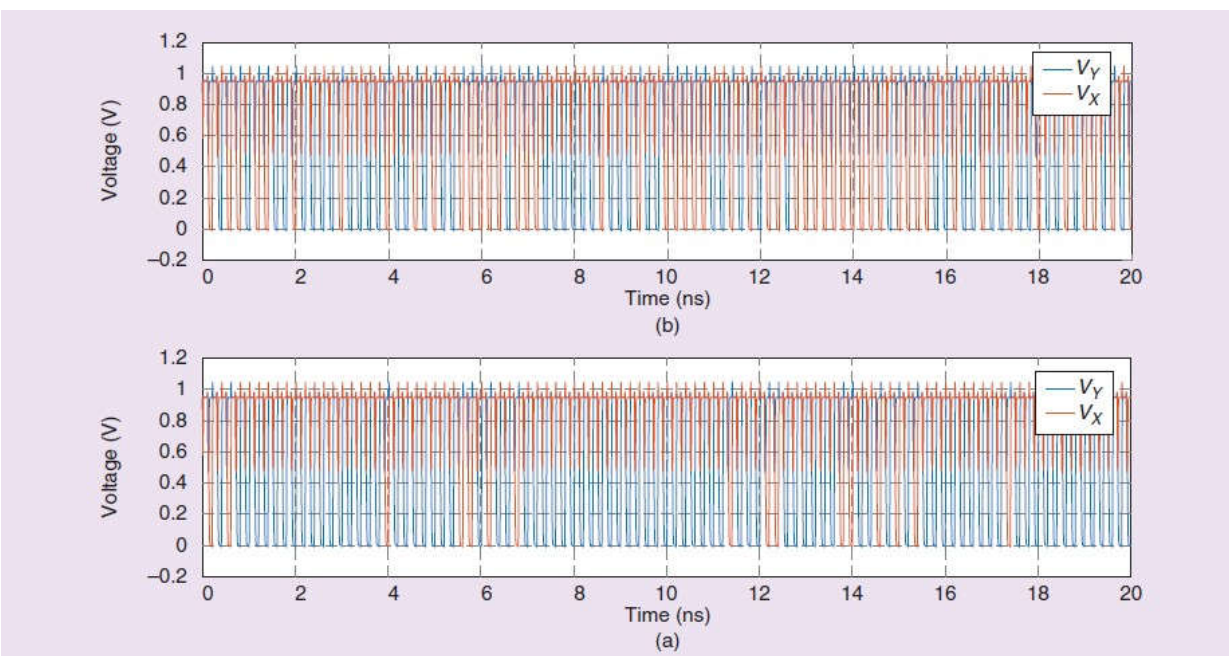


Figure 5-62 The comparator outputs (a) for perfect balance and (b) with an input difference equal to 0.31 mV.

And thus, in our testbench the VS source will be variable and will include a large range of possible values, since we already know that it will be less than 1mv we chose the range from -1mv to 1mv with a step of 12.5uv, we added the transient noise of 1G to the simulation, the more the

noise, the more accurate the test will be, but this is taking too much time already, and we set the testbench to run for 2000 clock cycles and run it.

The test bench :

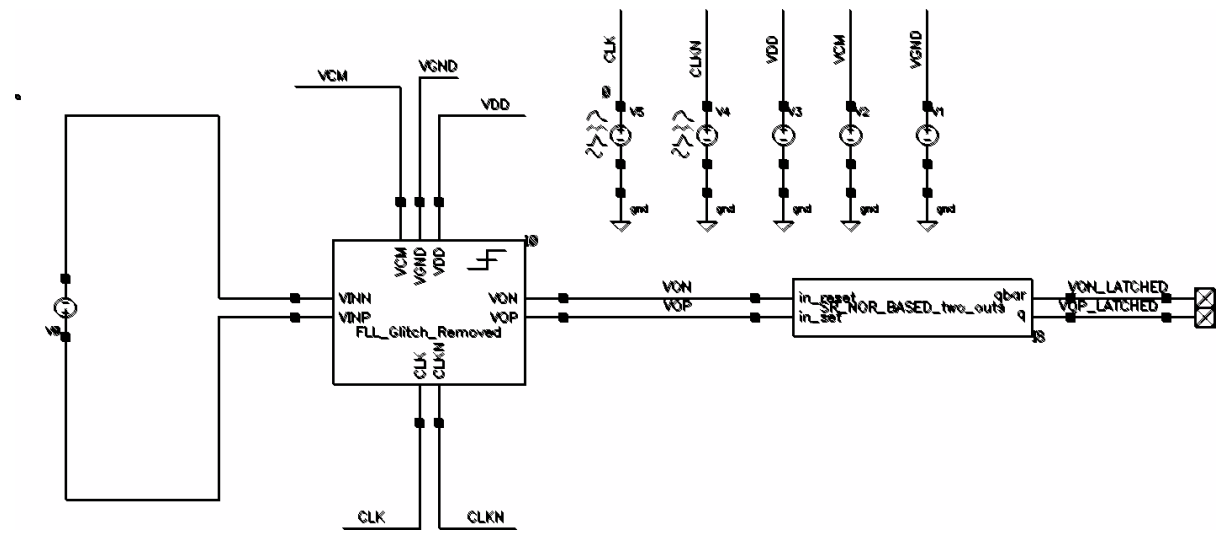


Figure 5-63 Thermal transient noise analysis testbench

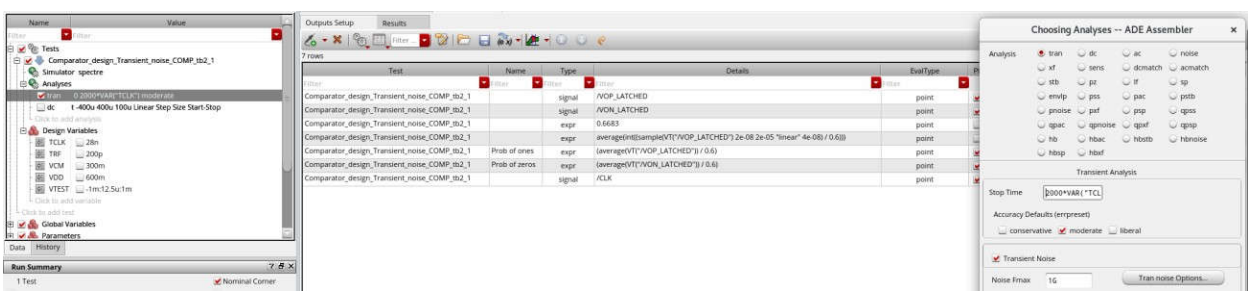


Figure 5-64 Thermal transient noise analysis testbench data variables

The probability of ones and zeros can be found getting the average of the output signal VOP and VON (it needs to be latched not resettable), then normalizing it to VDD.

The results will be as follows, with the probability of ones in Figure 5-65, the IRN is $\sigma in = \frac{391.93-2.0321}{1.28} = 304.6077uv$, or from the probability of zeros in Figure 5-66 at 16% which gives 300.76uv, both are nearly similar, the reason this is showing optimistic results better than the periodic noise tb, is that we have not integrated the full noise, the upper limit needs to be infinity not just 1G, but this will take very long time, at least we got a sense where the noise is placed. If we could do 100G we would have.

Using the probability of ones and zeros :

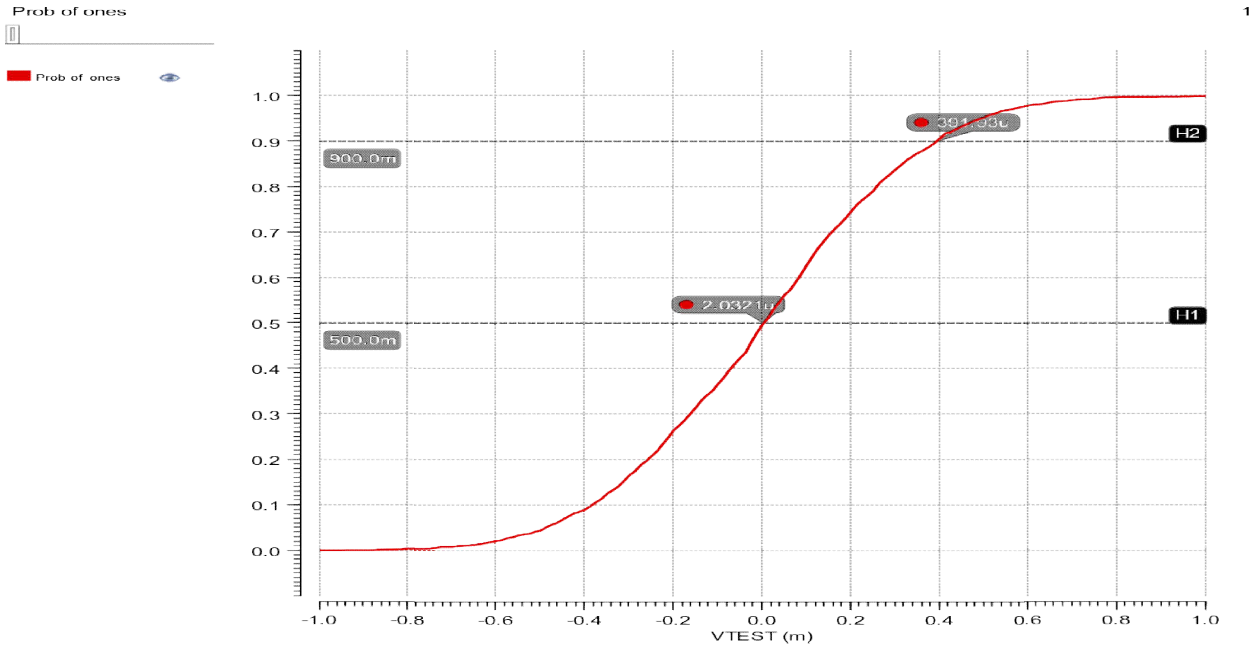


Figure 5-65 Probability of ones

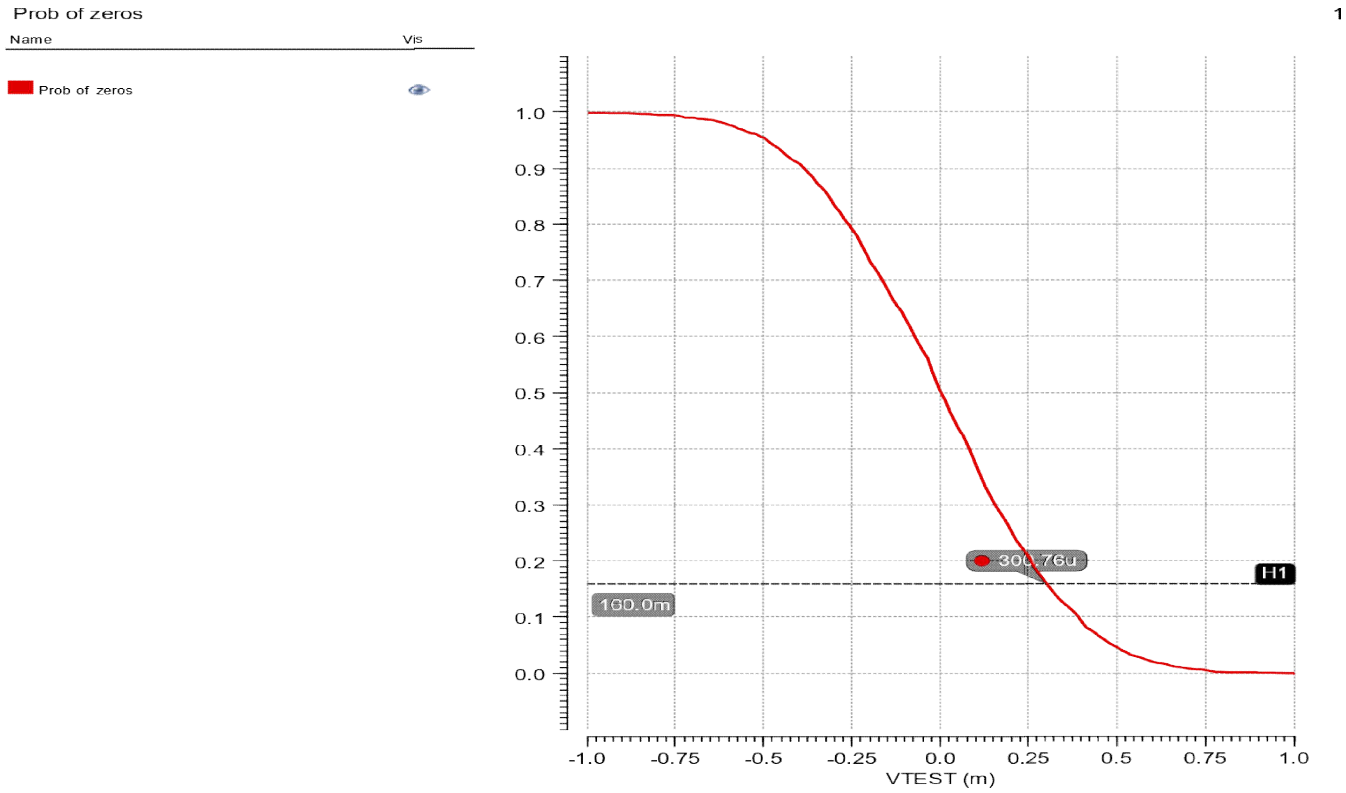


Figure 5-66 Probability of zeros

5.13 Offset analysis

The proposed technique in [49] uses a modified successive approximation algorithm to determine the offset voltage in short simulation time while preserving accuracy and precision. The comparator's input waveform is selectively reset to provide a **quasi-monotonic stimulus** to the comparator; thus, comparator's hysteresis can be identified and accurately determined. The technique is implemented in a parameterized Verilog-A model and can be applied to any type of comparator.

Monte Carlo simulation of Strong-Arm dynamic comparator implemented in 180 nm CMOS technology [49] shows that the proposed technique achieves 45 × speed-up in simulation time compared to the standard linear search, while achieving similar accuracy. One of the most important specifications of the comparator is its offset voltage (Vos). In many types of circuits, the comparator offset imposes a fundamental limit on the achievable performance. In every case, it is important for the designer to obtain an accurate estimation of Vos through circuit simulations, in order to validate his design and verify that the offset voltage meets the required specifications, prior to the lengthy and expensive fabrication cycle.

The standard technique for comparator offset simulation is to use a rising ramp (stair-case) input signal and detect the output transition [58,59]. The input voltage at which the output performs a low-to-high transition is Vos in the rising direction (Vos,R). Next, a falling ramp is applied, where the input voltage at which the output performs a high-to- low transition is Vos in the falling direction (Vos,F). If Vos,R and Vos,F are not equal, then the comparator suffers from hysteresis. Depending on the application, some comparators are intentionally designed with some degree of hysteresis, while others may be impaired with hysteresis due to unintentional and undesirable behavior [56,60]. You can see the full operation in Figure 5.68, with the VOS,R and VOS,F separately.

The drawback of the stair-case linear search is that it must use a fine voltage step to achieve adequate precision; thus, it requires very long simulation time to obtain a single measurement. Noting that Vos is a statistical random variable, the simulation must be repeated for hundreds of Monte Carlo (MC) runs to calculate Vos statistical distribution. Consequently, the total simulation time can be several hours or even longer, which limit productivity to a level that would hinder progress.

In [49], the author proposes a simulation technique that alleviates the drawbacks of the previous offset simulation techniques. The proposed technique uses binary search to determine the comparator offset voltage in short simulation time. However, the comparator's input stimulus is

selectively modified to provide a quasi-monotonic input waveform. Consequently, the comparator's threshold is crossed in a well-defined pre-determined direction. As a result, Vos,R and Vos,F can be accurately measured, even if the comparator suffers from hysteresis.

The proposed test bench is as shown in Figure 5-67, The Vos Tester block provides all the necessary signals to simulate the comparator; thus, no other circuit elements are required in the test bench. It provides the comparator with supply voltage, clock signal, and input voltage. It continuously monitors the comparator's output. For open loop offset measurement techniques, the input stimulus is predefined, and the comparator's output is monitored to detect the input voltage at which the output transition happens. On the other hand, for closed-loop techniques, the comparator's output is used to modify the input stimulus according to the implemented algorithm. The Vos Tester generates two signals Vos,R and Vos,F for the calculated offset voltages.

The Vos Tester is fully parameterized to be universally applied to any type of comparator regardless of its implementation details. The user can adjust the clock timing parameters in addition to the active clock edge. Moreover, the clock can be left unconnected for non-clocked comparators. Both comparator's input and output signals can be selected to be single-ended or differential. For singled-ended input $Vinn$ is pinned to a user-selected common-mode (CM) voltage and the stimulus waveform is applied to $Vinp$. For differential input, the waveform is split differentially between $Vinp$ and $Vinn$, where both are superimposed on a user-selected CM voltage. The user can also select the offset voltage search range and the search resolution which comes very handy in different tests.

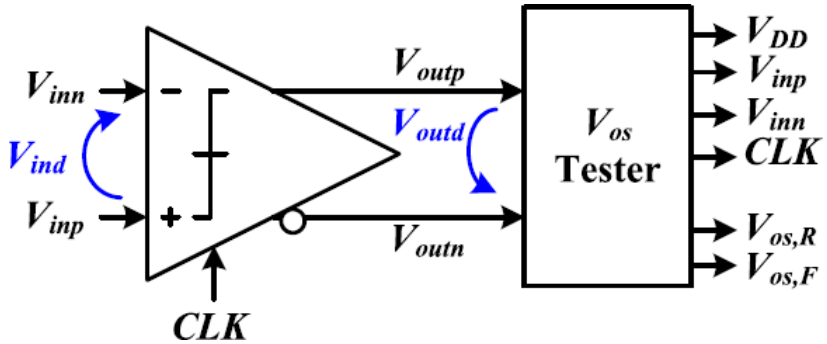
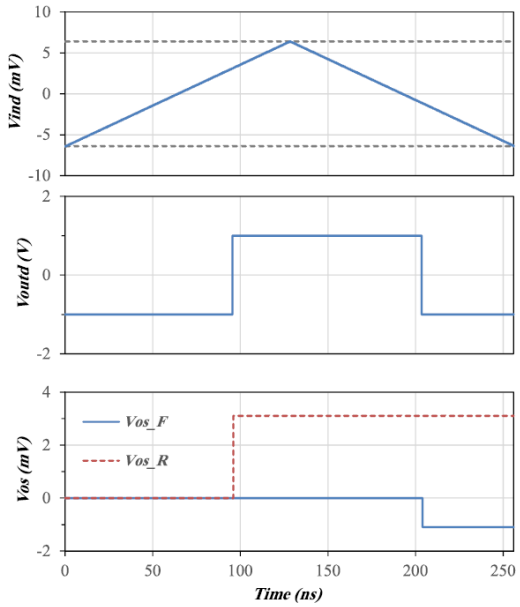
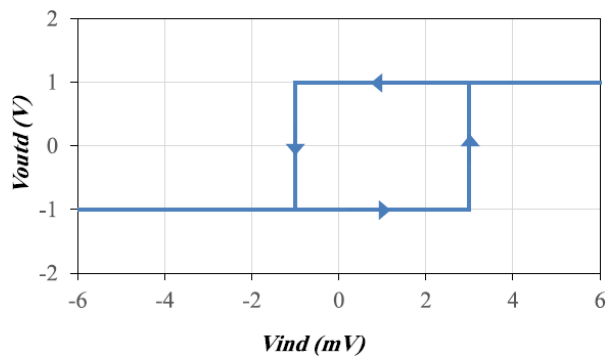


Figure 5-67 Schematic of the testbench used for comparator characterization showing inputs and outputs of Vos Tester block. [49]





Characteristics of a comparator with hysteresis used as an example.

Figure 5-68 Linear search technique transient simulation results showing comparator's input voltage ($Vind$), output voltage ($Voutd$) and offset voltage (Vos). The clock period is 1 ns. The value of Vos is calculated by Vos Tester block. [49]

In order to illustrate the limitations of conventional offset simulation techniques, consider the fully-differential comparator characteristics shown in Fig. 5-68 with Vos,R = +3 mV and Vos,F = −1 mV. The comparator suffers from a hysteresis window equal to 4 mV. Assume the required search range is −5.4 mV to +5.4 mV, the search resolution is 0.1 mV, and the clock period is 1 ns? If the standard linear search technique is used [58,59], rising and falling ramps are applied with 0.1 mVstep. The input waveform ($Vind$) is shown in Fig. 5.68, where the required number of clock cycles is given by:

$$NCLK,\ Linear = (Search\ Range/Search\ Resolution) \times 2 = 256 \qquad (5.24)$$

The transition of $Voutd$ during the rising ramp signals Vos,R and the transition during the falling ramp signals Vos,F. As shown in Fig. 5.68, the Vos Tester block successfully calculates Vos,R and Vos,F using the linear search algorithm. Although this technique is simple and accurate, it suffers from long simulation time, especially for large search ranges which are commonly required for dynamic comparators that would drastically long amount of time that is inapplicable.

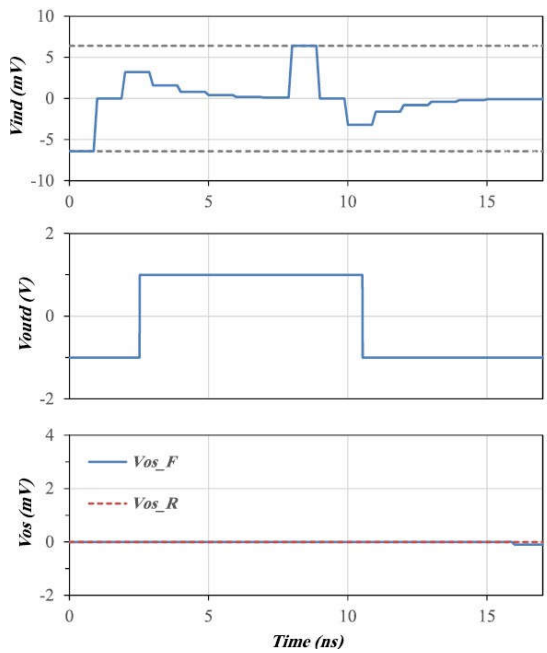


Figure 5-69 Successive approximation register (SAR) transient simulation results showing comparator's input voltage ($Vind$), output voltage ($Voutd$) and offset voltage (Vos). The clock period is 1 ns. The value of Vos is calculated by Vos Tester block. [49]

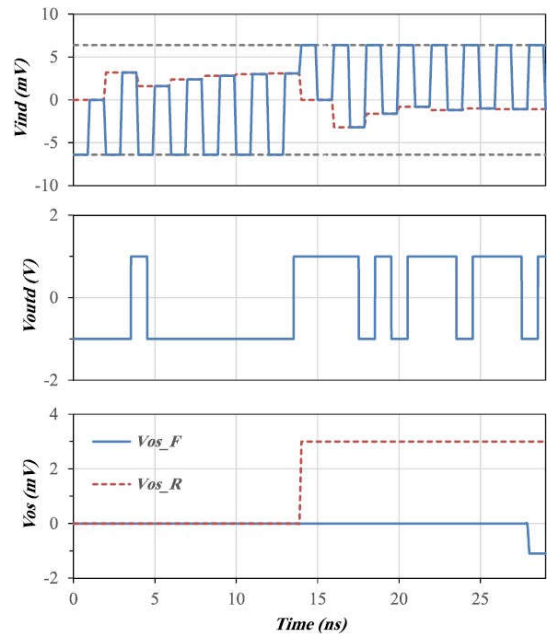


Figure 5-70 Resettable successive approximation register ($RSAR$) transient simulation results showing comparator's input voltage ($Vind$), output voltage ($Voutd$) and offset voltage (Vos). The clock period is 1 ns. The value of Vos is calculated by Vos Tester block. [49]

In order to alleviate the drawback of linear search, the SAR binary search algorithm can be used [63]. Using the same parameters of the previous example, the number of bits employed in the binary search (number of successive approximation cycles) is given by

$$N_{BIT} = \log_2 \frac{Search\,Range}{Search\,Resolution} = 7 \qquad (5.25)$$

Assuming that two search cycles will be performed, with one cycle starting at the minimum level of the search range and the other starting at the maximum, the total number of simulated clock cycles is given by

$$NCLK,SAR = 2\,(NBIT + 1) = 16 \qquad (5.26)$$

where the SAR binary search takes $NBIT$ clock cycles in every search direction, in addition to one initial cycle. Comparing (5.26) and (5.24), it is clear that $NCLK,SAR$ is drastically smaller than $NCLK,Linear$, which demonstrates the speed advantage of the SAR technique. Fig. 5-69 shows the

results of 7-bit *SAR* algorithm. Although it has the same effective resolution as linear search (0.1 mV), it fails to get a correct result for $V_{OS,R}$ and $V_{OS,F}$. Regardless of starting with initial value at the minimum or maximum levels, the input voltage gets trapped to the value with which it first hits the hysteresis region. Consequently, the calculated V_{OS} is neither $V_{OS,R}$ nor $V_{OS,F}$ nor their average. In conclusion, although the *SAR* technique has a clear speed advantage, it does not yield a correct result for the offset voltage if the comparator suffers from hysteresis.

The inaccuracy of the *SAR* technique is due to the fact that the comparator's threshold crossing direction is not user-controlled. The input waveform is solely determined by the binary search algorithm; thus, the probabilities of rising and falling crossings are equal. A direct remedy to this limitation can be achieved by using a resettable version of the *SAR* technique. The idea of resettable *SAR* (*RSAR*) is to reset the comparator by forcing a predefined input before applying the next input voltage as calculated by the *SAR* algorithm. As shown in Fig. 5.70, during the search for $V_{OS,R}$ the comparator is reset by applying the minimum value of the search range for one cycle before applying each *SAR* value; thus, a rising threshold crossing is guaranteed. On the other hand, during the search for $V_{OS,F}$, the input voltage of the reset state is the maximum level of the search range; thus, a falling threshold crossing is guaranteed. As a result, correct $V_{OS,R}$ and $V_{OS,F}$ are calculated similar to linear search. The required number of clock cycles is given by

$$N_{CLK,RSAR} = 4N_{BIT} = 28 \qquad (5.27)$$

where the number of clock cycles in every search direction is $2N_{BIT}$, since each of the N_{BIT} *SAR* cycles is preceded by an initial reset cycle.

Although the *RSAR* technique can accurately resolve the comparator's hysteresis, it roughly doubles the simulation time compared to conventional *SAR*. This drawback can be mitigated by noting that it is not necessary to reset the comparator every time before applying the next *SAR* value**. On the contrary, resetting the comparator is only required if the next *SAR* value will dictate a threshold crossing in a direction opposite to the one being tested**. The smart resettable *SAR* (*SRSAR*) technique checks for this condition programmatically inside the V_{OS} Tester Verilog-A model, thus saving simulation time.

As shown in Fig. 5.71, by using *SRSAR* technique the input waveform does not go to the reset state after each comparison. However, it goes to the reset state if one of the following two conditions are detected:

1. Rising output transition during $Vos.R$ search
2. Falling output transition during Vos,F search

The input stimulus starts at the initial reset value for one clock cycle at the beginning of every search direction, i.e., before testing the most significant bit (MSB). Thus, noting that the probability that the aforementioned conditions will be satisfied for the remaining 2 ($NBIT - 1$) test cycles is 50% on average, the average number of clock cycles required by $SRSAR$ technique is given by

$$NCLK,SRSAR = NCLK,SAR + (NBIT - 1) = 3NBIT + 1 = 22 \qquad (5.28)$$

which translates to 21% time saving compared to $RSAR$. For large $NBIT$, the speed-up of $SRSAR$ compared to $RSAR$ will approach 25%. Thus, $SRSAR$ can calculate correct Vos,R and Vos,F taking hysteresis into account like $RSAR$, but with a reduced overhead.

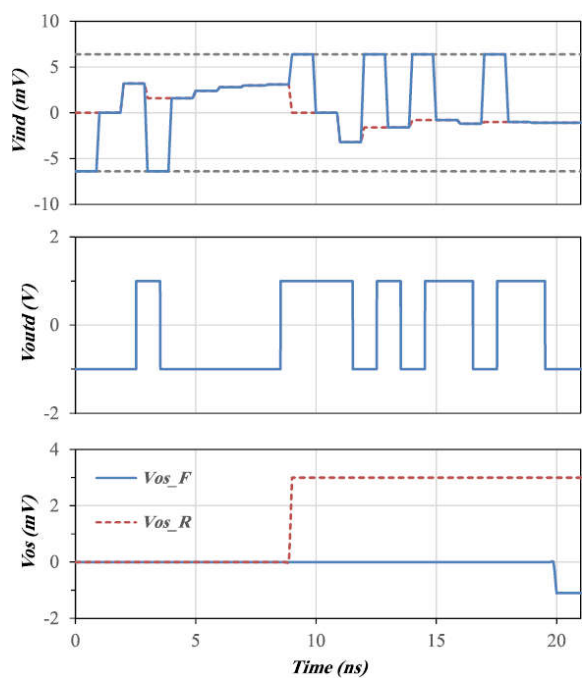


Figure 5-71 Smart resettable successive approximation register ($SRSAR$) transient simulation results showing comparator's input voltage ($Vind$), output voltage ($Voutd$) and offset voltage (Vos). The clock period is 1 ns. The value of Vos is calculated by Vos Tester block. [49]

The case of incremental voltage steps is an artificial case that is not typically present when the comparator is employed in a real mixed-signal system. Therefore, the slightly pessimistic estimate of the *SRSAR* technique provides the designer with a more reliable result that mimics the actual behavior of the comparator. Remarkably, the 500 MC runs simulation time for *SRSAR* is 6 min only, compared to 4 h and 30 min for linear search, which translates to a 45 × speed-up. It is worth noting that if the search range is doubled, i.e., −100 mV–100 mV instead of −50 mV to 50 mV, the simulation time of the linear search will be doubled, while the simulation time of the *SRSAR* search will increase by 10% only (only one additional bit). This will translate to an 82 × speed-up, i.e., the speed-up becomes even higher as the search range increases.

Now let's discuss how we implemented this in our design :

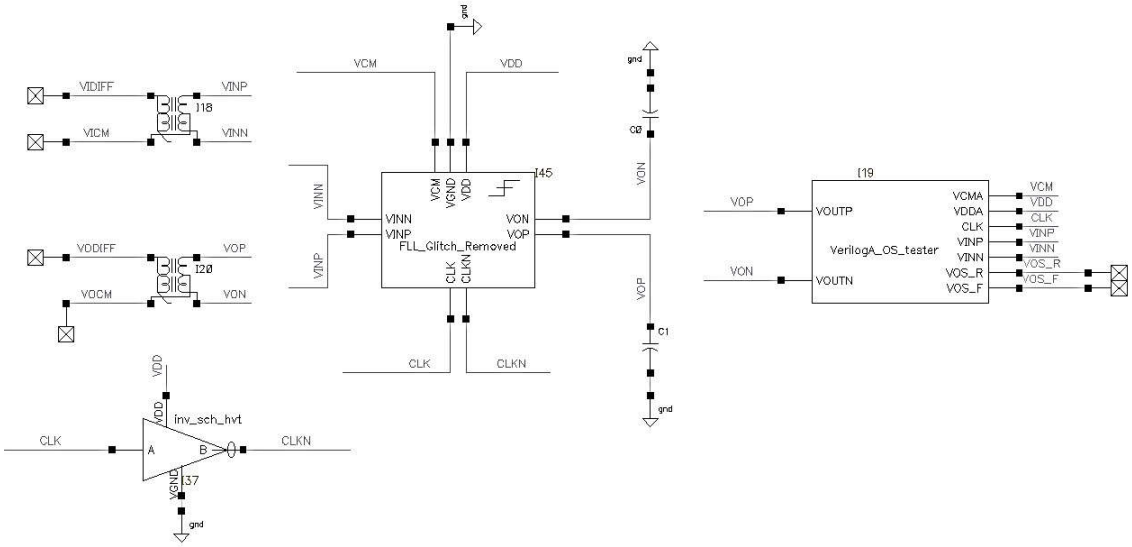


Figure 5-72 Offset calculation using VerilogA testing by the SRSAR method

```
1  // Smart resettable SAR VOS Tester
2  // H.O
3  `include "disciplines.vams"
4  // code conventions
5  // 1 : positive / true / yes / rising / max
6  // 0 : negative / false/ no / falling / min
7  // upper_case : parameters
8  // lower_case : internal variables
9  module vos_tester_srsar(VDDA,VCMA,CLK,VINP,VINN,VOUTP,VOUTN,VOS_R,VOS_F);
10
11 input VOUTP,VOUTN;
12 output VDDA,VCMA,CLK,VOS_R,VOS_F;
13 electrical VDDA,VCMA,CLK,VINP,VINN,VOUTP,VOUTN,VOS_R,VOS_F;
14
15 // comparator supply voltage
16 parameter real VDD = 0.6 from (0:100];
17 // comparator CM input (for diff input)
18 parameter real VCM = 0.3 from (0:VDD);
19 // comparator min_diff input voltage
20 parameter real VIN_MIN = -50m from [-VDD:0];
21 // comparator max_diff input voltage
22 parameter real VIN_MAX = 50m from [0:VDD];
23 // comparator high output voltage
24 parameter real VOUT_HIGH = VDD  from [0:VDD];
25 // comparator low output voltage
26 parameter real VOUT_LOW = 0  from [-VDD:VDD];
27 // is comparator input diff?
28 parameter real DIFF_IN = 1  from [0:1];
29 // is comparator output diff?
30 parameter real DIFF_OUT = 1  from [0:1];
31 // clock period
32 parameter real TCLK = 100n from (0:1];
33 // rise time
34 parameter real TR = 10n from [0:TCLK/2);
35 // fall time
36 parameter real TF = 10n from [0:TCLK/2);
37
38 // Clock active edge (1:rising, 0 : falling)
39 // the active edge fires the comparison
40 // input is updated and output is checked
41 // just before the non-active edge
42
43 parameter integer CLK_EDGE = 1 from [0:1];
44 // no. of SAR bits
45 parameter integer NBIT = 12 from [4:20];
46
47 // real variables by default init to zero
48 // current clock state (0 or 1)
```

```
46
47 // real variables by default init to zero
48 // current clock state (0 or 1)
49
50 integer clk_state;
51 // small delay to avoid race condition
52 real td = TCLK / 1e3;
53 // start n reset state
54 integer reset_state = 1;
55 // comparator input is modeled as diff
56 // input direction (1:rising,0:falling)
57 integer vind_dir = 1;
58 // range of diff input
59 real vind_range;
60 // vind SAR value (ignores reset)
61 real vind_sar_val;
62 // vind value (SAR or reset value)
63 real vind_val;
64 // comparator positive input (active signal)
65 real vinp_val;
66 // comparator negative input (active signal
67 // for diff input and DC for SE input)
68 real vinn_val;
69 // comparator output threshold
70 real voutd_th ;
71 // comparator output is modeled as diff
72 real voutd_val;
73 // detect low-to-high output transition
74 integer l2h_event = 0;
75 // detect high-to-ow output transition
76 integer h2l_event = 0;
77 // counter used for SAR
78 integer count;
79 // measured Vos, rising
80 real vos_r_val;
81 // measured Vos, falling
82 real vos_f_val;
83
84
85 analog begin
86 V(VDDA) <+ VDD;
87 V(VCMA) <+ VCM;
88 voutd_th = (DIFF_OUT == 1) ? 0 : (VOUT_HIGH + VOUT_LOW)/2;
89 // for SE output connect VOUIN to GND
90 voutd_val = V(VOUTP,VOUTN);
91 @(initial_step) begin
92 // init clock so 1st edge will be active
93     clk_state = (CLK_EDGE == 1) ? 0 : 1;
```

```
91 @(initial_step) begin
92 // init clock so 1st edge will be active
93     clk_state = (CLK_EDGE == 1) ? 0 : 1;
94     vind_range = VIN_MAX - VIN_MIN;
95     count = 1;
96 // nit SAR value
97     vind_sar_val = VIN_MIN + vind_range/pow(2,count);
98 // nit reset value
99     vind_val = VIN_MIN;
100    vinp_val = (DIFF_IN == 1) ? VCM + vind_val/2 : VCM + vind_val;
101    vinn_val = (DIFF_IN == 1) ? VCM - vind_val/2 : VCM;
102 end
103
104
105 // Timer triggers before every clock edge
106 // due to transition delay (td)
107 // 1st active edge triggered after TCLK/2
108 // assume 50% duty cycle
109 @(timer(TCLK/2,TCLK/2)) begin
110     clk_state = !clk_state;
111     //update vind and check voutd_val just
112     //before CLK non-active edge
113     if(clk_state != CLK_EDGE) begin
114     // skip output checking if in reset stati
115         if (reset_state == 1) begin
116         // apply next sar value
117             vind_val = vind_sar_val;
118         // go to SAR state
119             reset_state = 0;
120         end
121         // update vind and check voutd
122         else begin
123         // resetting count is the done flag
124             if (count == 0)
125         // end tran sim (but not the MC)
126             $finish_current_analysis;
127             else if (count < NBIT) begin
128                 count = count + 1;
129         // calculate next vind sar value
130                 if (voutd_val > voutd_th) begin
131         // vind should jump down
132                     vind_sar_val = vind_sar_val- vind_range/pow(2,count)
133         // is vind in the rising test?
134                     if (vind_dir == +1)
135         // reset to VIN_MIN before
136         // jumping down to avoid hys
137                         reset_state = 1;
```

```
125         // end tran sim (but not the MC)
126             $finish_current_analysis;
127             else if (count < NBIT) begin
128                 count = count + 1;
129         // calculate next vind sar value
130                 if (voutd_val > voutd_th) begin
131         // vind should jump down
132                     vind_sar_val = vind_sar_val- vind_range/pow(2,count);
133         // is vind in the rising test?
134                     if (vind_dir == +1)
135         // reset to VIN_MIN before
136         // jumping down to avoid hys
137                         reset_state = 1;
138                 end
139                 else begin
140         // vind should jump up
141                     vind_sar_val = vind_sar_val + vind_range/pow(2,count);
142                     // is vind in the falling test?
143                     if (vind_dir == 0)
144                     // reset to VINL_MAX before
145                     // jumping up to avoid hys
146                         reset_state = 1;
147                 end
148             end
149             // end of SAR for rising/falling test
150             else if (count == NBIT) begin
151             // retain or remove SAR LSB
152                 if (voutd_val > voutd_th)
153                     vind_sar_val = vind_sar_val- vind_range/pow(2, count);
154                 if (vind_dir == +1) begin
155                     vos_r_val = vind_sar_val;
156                     // switch from rising to falling
157                     vind_dir = 0;
158                     // reset count
159                     count = 1;
160                     // init SAR value
161                     vind_sar_val = VIN_MIN + vind_range/pow(2,count);
162                     // init reset value
163                     vind_val = VIN_MAX;
164                     // go to reset state
165                     reset_state = 1;
166                 end
167                 else if (vind_dir == 0) begin
168                     vos_f_val = vind_sar_val;
169                     // resetting count is the done flag
170                     count = 0;
171                 end
172             end
```

```
151             // retain or remove SAR LSB
152                 if (voutd_val > voutd_th)
153                     vind_sar_val = vind_sar_val- vind_range/pow(2, count);
154                 if (vind_dir == +1) begin
155                     vos_r_val = vind_sar_val;
156                     // switch from rising to falling
157                     vind_dir = 0;
158                     // reset count
159                     count = 1;
160                     // init SAR value
161                     vind_sar_val = VIN_MIN + vind_range/pow(2,count);
162                     // init reset value
163                     vind_val = VIN_MAX;
164                     // go to reset state
165                     reset_state = 1;
166                 end
167                 else if (vind_dir == 0) begin
168                     vos_f_val = vind_sar_val;
169                     // resetting count is the done flag
170                     count = 0;
171                 end
172             end
173             if (reset_state == 1)
174             // reset state: assign reset value
175                 vind_val = (vind_dir == 1) ? VIN_MIN : VIN_MAX;
176             else
177             // SAR state: assign vind SAR value
178                 vind_val = vind_sar_val;
179             end
180         end
181         vinp_val = (DIFF_IN == 1) ? VCM + vind_val/2 : VCM + vind_val;
182         vinn_val = (DIFF_IN == 1) ? VCM - vind_val/2 : VCM;
183     end
184     V(CLK)  <+ transition (clk_state*VDD, td ,TR,TF);
185     V(VINP) <+ transition (vinp_val,td ,TR,TF);
186     V(VINN) <+ transition (vinn_val,td ,TR,TF);
187     V(VOS_R)<+ vos_r_val;
188     V(VOS_F) <+ vos_f_val;
189 end
190 endmodule
191
192
193
194
195
196
197
```

Figure 5-73 VerilogA code for the VerilogA tester module.

The VerilogA code enables you to set the input range, define whether the system is differential or single ended, whether it is positive or negative edge triggered and the rise/fall time of the produced clock, which makes the OS-tester quite modifiable for any circumstance.

We have run 500 Monte-Carlo simulation run, we even used this method to complete the sizing of the comparator, as according to pelgrom's law $\sigma_{V_{TH}} = \frac{A_{V_{TH}}}{\sqrt{WL}}$, the mismatch is inversely proportional to the square root of the area, this may be a law for the threshold voltage mismatch, but the idea is definitely the same for most of the parameters, mainly as we increase the Area the mismatch decreases, so we know that the Preamp transistors are the main factors contributing to the static mismatch of the comparator we tried increasing the width of the Preamp until we found a good trade-off with the power consumption, noise and offset as we stated earlier. The Latch contributes to the dynamic offset like clock feedthrough and charge injection as discussed in section 5.6, so after multiple re-sizing and simulation runs the results are :

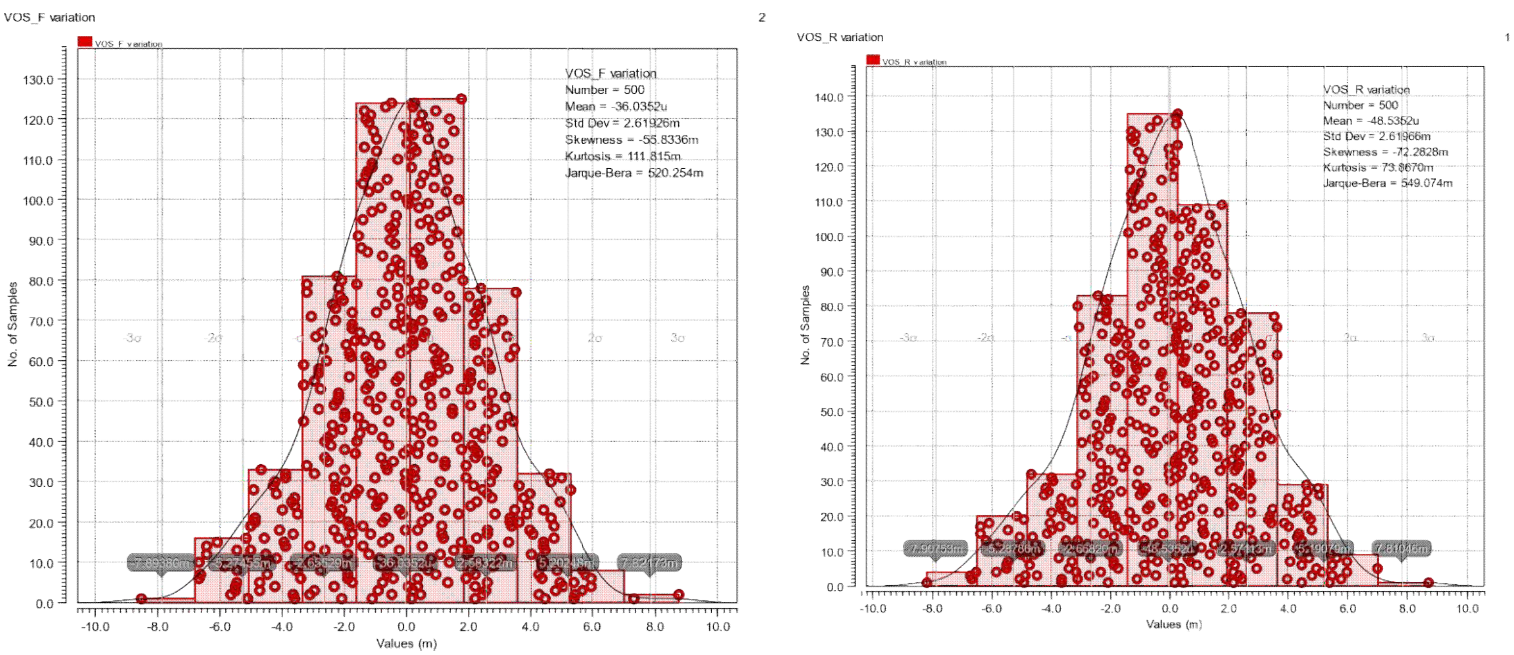


Figure 5-74 VOS-F and VOS-R of the final version of the comparator

Let's discuss a few things about the results :

1- The hysteresis mean is the difference between the VOS-R mean and the VOS-F mean and here the hysteresis is about 12.5uv, this is hysteresis window, and this is very small relative to the LSB which is 300uv and this is quite good, because in this design we are not

look for intentional hysteresis, may be can add it intentionally in future works to overcome the effects of noise.

2- We need to know what these values and what the Monte-Carlo simulations actually mean, first of all, Monte-Carlo only deals with static offset, it does not account for dynamic offset, it takes static offset from the mismatch models that we add, so now the results tell us that the mean value is about -36uv or -48uv, and the standard deviation is 2.619mv, this means that after the fabrication process, most of the chips will suffer from an offset that is equal to the mean value result from the simulation, and some of the chips will suffer from an offset of mean $\pm$ standard deviation, depending on the percentage that we are taking, for example, most of the samples that are near the mean will suffer from -36uv or -48uv of a static offset mismatch, while 68% of the fabricated chips will suffer from -48uv$\pm$ 2.619mv of static offset, and 95% (corresponding to 2σ) will suffer from -48uv $\pm$ 2*2.619mv, about 99.7% of the fabricated chips will suffer from -48uv$\pm$3*2.619mv.

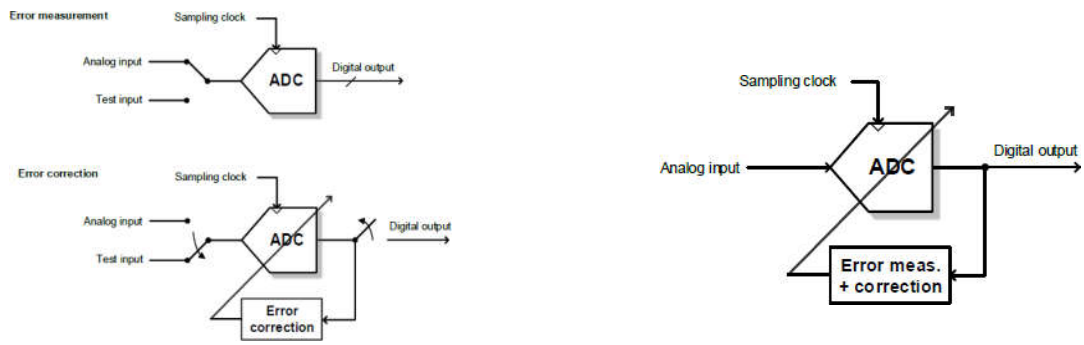
3- So, as we saw if we want to take into consideration 99.7% we would suffer from an offset of -48uv$\pm$3*2.619mv, which is NOT acceptable since the LSB is 300uv, this value of offset would limit the resolution of the system.

And thus, we have reached a point where cannot increase the sizes any further to obtain an offset of less than LSB, and thus we are in need of calibration technique as we shall see now :

Comparator offset calibration is divided into two parts background and foreground offset calibration.

Two types of calibration could be carried out depending on whether or not the conversion is interrupted to measure and correct the errors[71]. The first group, *Foreground Calibration*, needs a dedicated period of time to perform the calibration. This approach is also called offline calibration, since the normal operation of the converter is stopped to perform the error measurement or its correction. Both error measurement and correction are not simultaneously executed.

In applications in which the ADC cannot be disconnected or where the parameters have great variations, a second approach has been considered, which is faster than the foreground method. *Background Calibration*, also called online calibration, works simultaneously with the ADC. Error measurement and calibration are accomplished simultaneously and continuously without ceasing the normal operation of the ADC. Unlike in foreground techniques, no break in the input signal path exists. All parameters are estimated and corrected in a concurrent way, in contrast to online techniques where the calculation and correction of the parameters are done in a sequential manner.



` Figure 5-75 Foreground (left) and background (right) offset calibration [64]

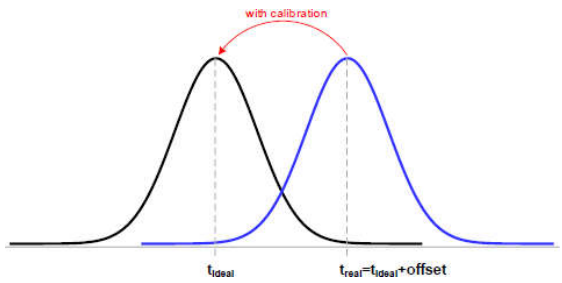


Figure 5-76 Offset calibration effect [64]

5.13.1 Background offset calibration

Background offset calibration as we have discussed is an online calibration technique which would calibrate the offset of the comparator during its operation, which is a very good and accurate measure of protection against offset mismatches, especially, the random mismatches and dynamic mismatches which might occur in the circuit and can be predetermined, this method of calibration would be able to prevent these errors from affecting the operation of the comparator.

They are many methods of background offset (online) calibration including the autozeroing and chopping and Dynamic element matching might be considered, also digital trimming I guess could be in this field as well , the one that is most commonly used is Autozeroing, this method was discussed in section 5.6, in the input and output offset storage then cancellation techniques, the main idea behind this method as we discussed in section 5.6, is the storage of the offset voltage

either on capacitor at the input or at the input, and in the amplification phase the offset gets cancelled with the previously stored value from the offset storage phase, we learnt that input offset storage is not completely perfect and there might not be full offset suppression, and that is why output offset storage might be better, observe Figure 5-77.

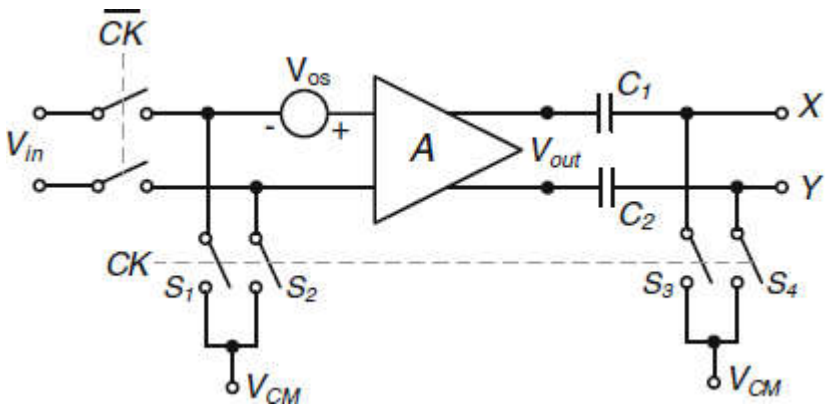


Figure 5-77 Autozeroing with output offset storage [72]

We tried to implement this method of output storage offset cancellation, what we learned is that it requires certain modification in the operation of the comparator, where during the inactive comparator period, at the time the logic is doing its function and the DAC is settling, the comparator would operate with its inputs connected together and thus the output that is stored on the output caps represent the offset, this is the offset storage phase, during the next phase which is the amplification phase, as discussed in section 5.6 the offset would be cancelled and the only value that would reach the latch is the amplified version of the differential input without the offset. We tried to implement a version of it, it did not work correctly but one thing we knew is that the power consumption increased extensively because the comparator (a power hungry block) is now operating twice the number of times with twice the power, also we are at the risk of a metastability where the comparator is supposed to be resetting its nodes at that time, even if we somehow made sure that the comparator had enough time to reset after every operation (twice per cycle) we would still be facing a huge problem which is the extensive increase in the power, some thing that we can not shake. We tried to give a preliminary view on the circuit in Figure 5-78 with the modulated clock operating the comparator two times per cycle.

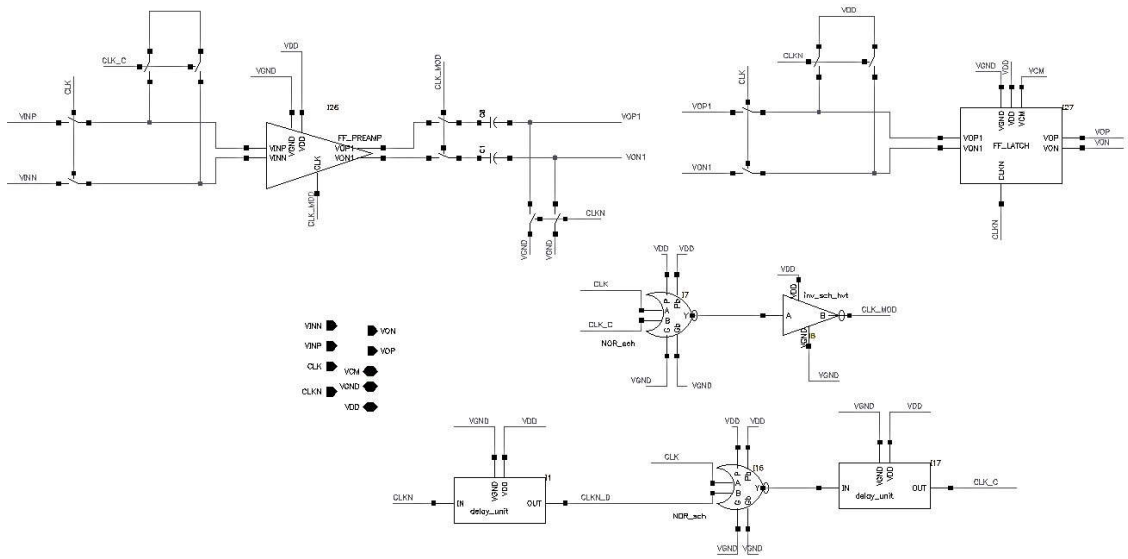


Figure 5-78 Preliminary view on the circuit

Again, this circuit is just a conceptual view, it still needs a lot of modifications.

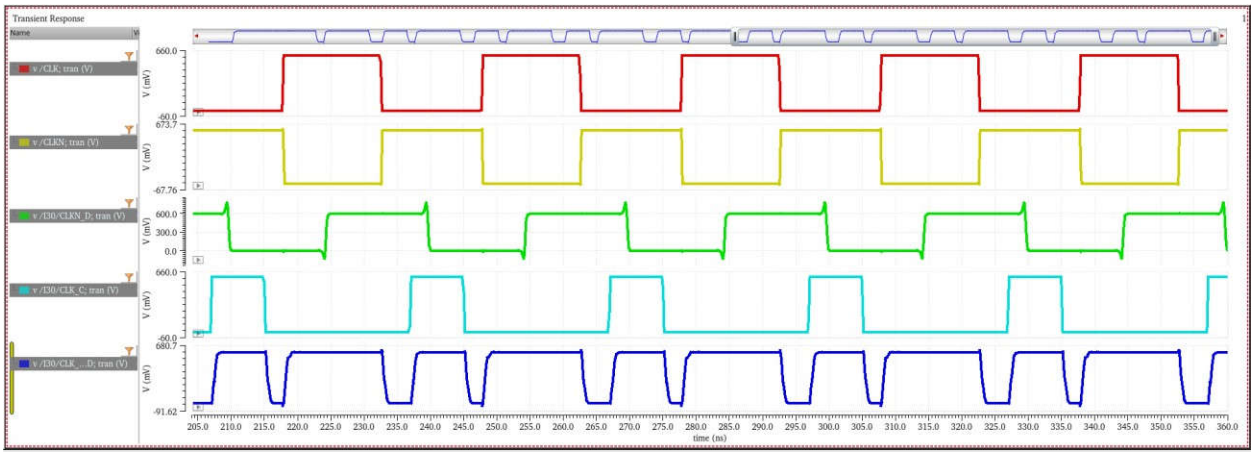


Figure 5-79 Controlling signals

But this did not work or was not what we needed so we went for the second method which is the foreground offset calibration.

5.13.2 Foreground offset calibration

The foreground offset calibration mechanism as we discussed earlier is an offline offset calibration mechanism, where the chip is not operational when the offset is being corrected, at the first phase, offset is tested using certain testing mechanism and then the chip is disconnected, and the calibration technique is implemented and then the chip is now suitable for customer use.

The concept of foreground calibration implemented in the comparator is the placement of inherent capacitive mismatch at the output nodes of the preamp, so instead of adding one cap or two caps, we can add a group of caps that can be digitally controlled, where each cap has a certain weight and a certain effect on the circuit offset ,say the measured offset is -1mv, then we can add a certain capacitance value to the negative preamp output VON1 which would decrease this offset to 0mv, of course we would have to sweep on the value of that cap. On the contrary, if the offset is 1mv we can add the same cap on the VOP1 instead of the VON1 to reduce the offset to 0mv, observe Figure 5-80.

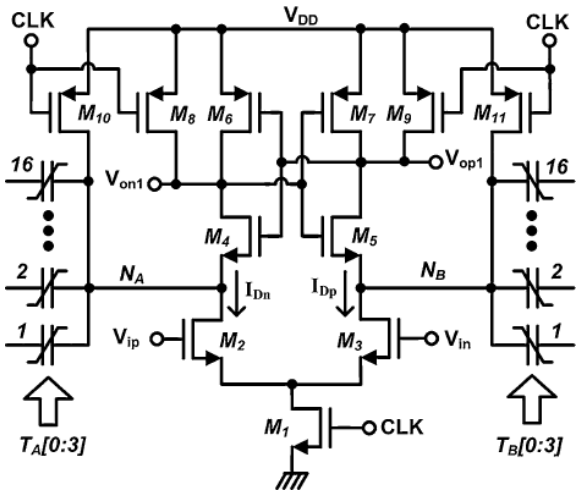


Figure 5-80 Digitally controlled capacitive weights on the preamp outputs.

So, what we propose is a coarse-fine capacitive array that can be digitally controlled by external signals, so that at the foundry, after the chip is manufactured, it will be tested and depending on the value of the offset the foundry can enable certain caps and disable the others.

The idea of the coarse-fine cap-array is the same method as the DAC, we have small caps in the fine array that is responsible for the fine tuning of the offset with an LSB value of LSB of the

ADC, say if the ADC has an LSB of 300uv, then the smallest change that this calibration array can do is an LSB change, then the fine array is binary weighted, where the 2nd cap is twice the LSB and the third is 4 times the LSB and so on. The coarse array on the other hand is implemented un the unary format, where every cap can be 16 to 32 times the LSB depending on the calibration range, of course all the large caps are made of unit caps placed in parallel together

As we have seen the offset for a target of 3σ is -48uv±3*2.619mv which is 8mv worth of calibration.

The first circuit is a foreground calibration circuit that calibrates up to 68% of the offset carrying chips which means only one 1σ.

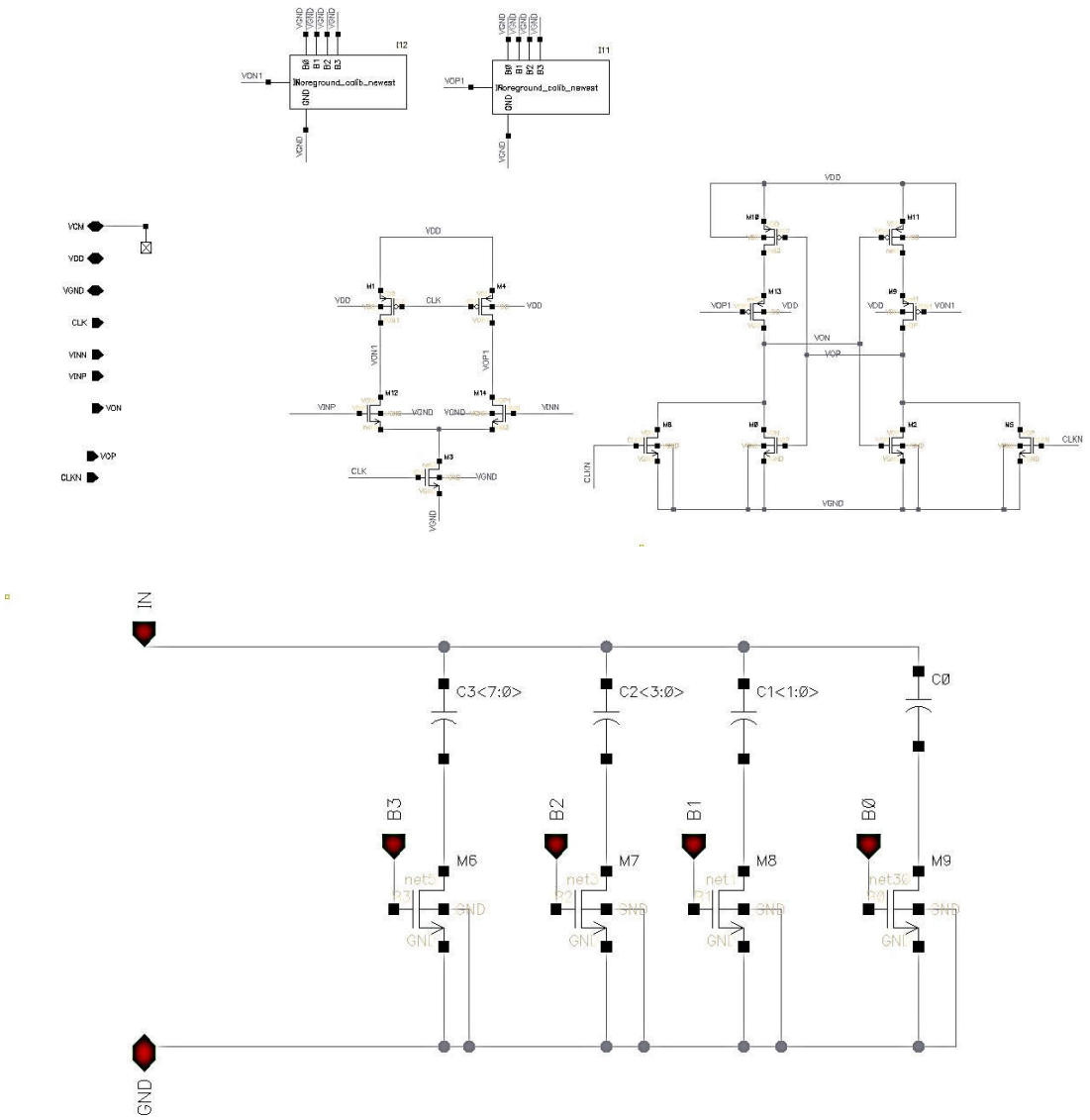


Figure 5-81 FLL circuit with foreground calibration circuit at the PREAMP outputs

When the calibration circuit is connected to VGND it means that it is not connected but once it is connected to VDD it will work the same way we said it would, B0 connection with calibrate 160uv (which is half LSB), the B1 is 320uv then 640uv then 1280uv in total this circuit calibrate up to 2.4mv which would leave an Offset of less than LSB, as you can see this circuit is simple and only binary weighted as there is no need for coarse-fine array here.

But if we want to calibrate more than just 1 σ , we will have to use this next circuit which has the ability to calibrate up to 23mv worth of offset mismatch.

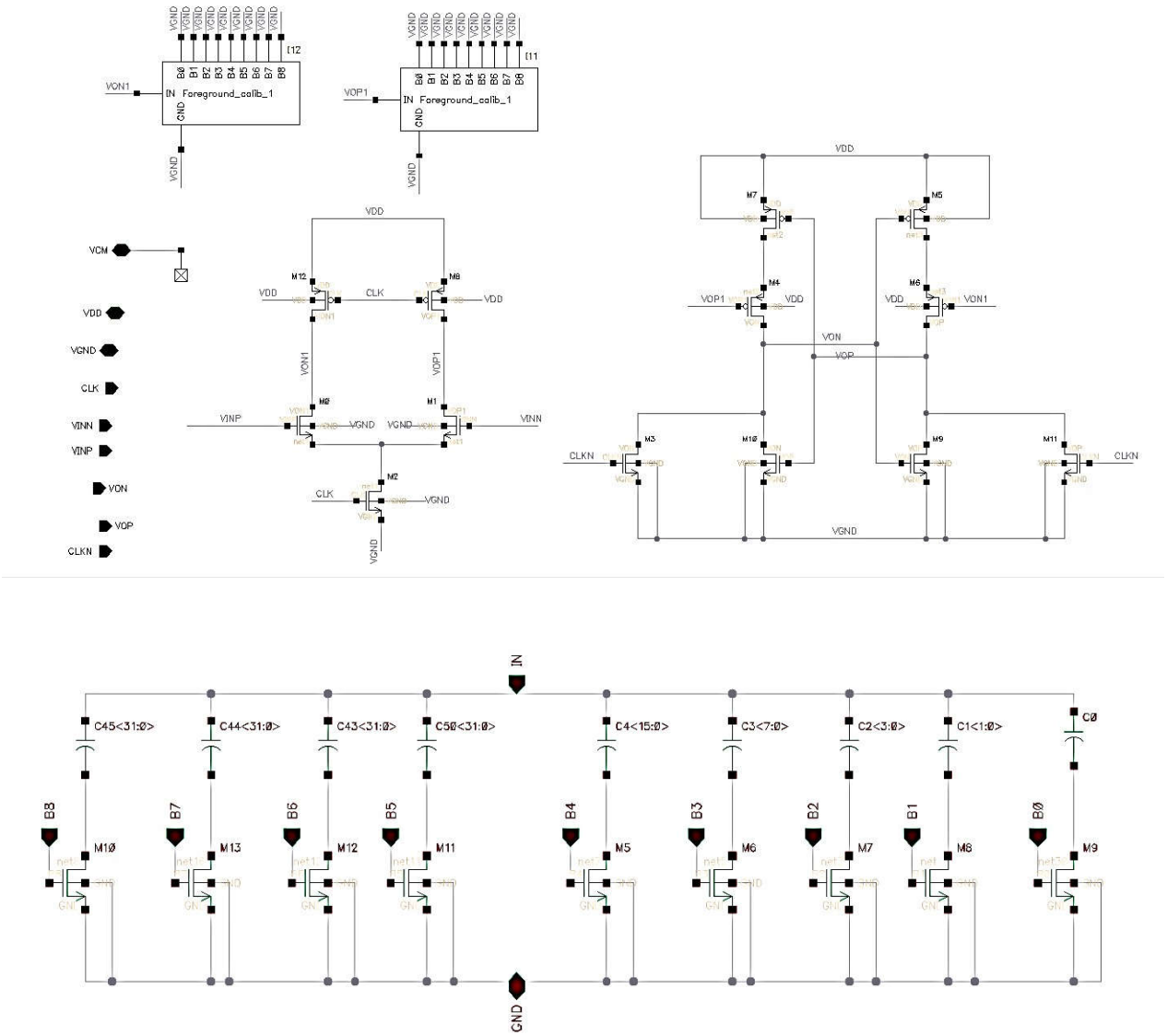


Figure 5-82 FLL circuit with foreground calibration circuit at the PREAMP outputs

Same concept is used here, where every unit cap is responsible for a change of 150uv, theoretically, B0 would be responsible for 150uv, then B1 for 300uv and so on, where the fine binary weighted array would be responsible for about 4.65mv of offset change, the coarse array is responsible for 18.6 mv with every unary weight of 4.65mv, of course the practical results is quite a bit different due to correlation between charges and many other factors but the weights are similar and we can still calibrate up to 23mv worth of mismatch which is about 9σ.

This large circuit does not actually need to be implemented because we do not need all that calibration, we only need up to 3 σ (99.7%) worth of calibration which can be achieved by the fine array with only one unary capacitor, but it is okay, we just wanted to present the idea for further use.

5.14 Power analysis

This chip is an Ultra-low power ADC chip, which consumes power les than 300nw as we shall see in the final chapter, and since this block (the comparator) is a power hungry block, it requires careful design procedures, as we have seen during the previous sections that discussed noise and offset analysis, power has always been a restriction on the sizing and came on the other side of the trade-off, the sizes of the comparator cannot be increased to a large extent because of the power thus it limited some performance specs like noise and offset which drove us to create some techniques like offset calibration to overcome such inefficiencies.

As we know the power in chips is divided into dynamic power and static power, the dynamic power is due to switching load capacitances and short circuit currents, while the static power is mainly due to leakage and rail to rail short circuit currents (in case there is a direct path between the power and the ground), Static power is consumed even when chip is idle.

-Leakage draws power from nominally OFF devices

-Prior to the 90 nm node

-Leakage power was negligible compared to dynamic power

-It was of concern primarily during sleep mode

-In nanometer processes

-Low threshold voltages and thin gate oxides

-Leakage can account for as much as a third of total active power

Since the supply voltage is 0.6v, this has helped us to achieve such a low power operation, but one of the major problems that faced us was the leakage, Aggressive scaling of CMOS devices at each technology generation has been offering ever-increasing integration capacity and circuit performance. However, the off-state leakage has also been rising significantly with technology scaling . Under 65nm CMOS technology, leakage power increases to 30%-40% of the total chip power and dominates the switching power .

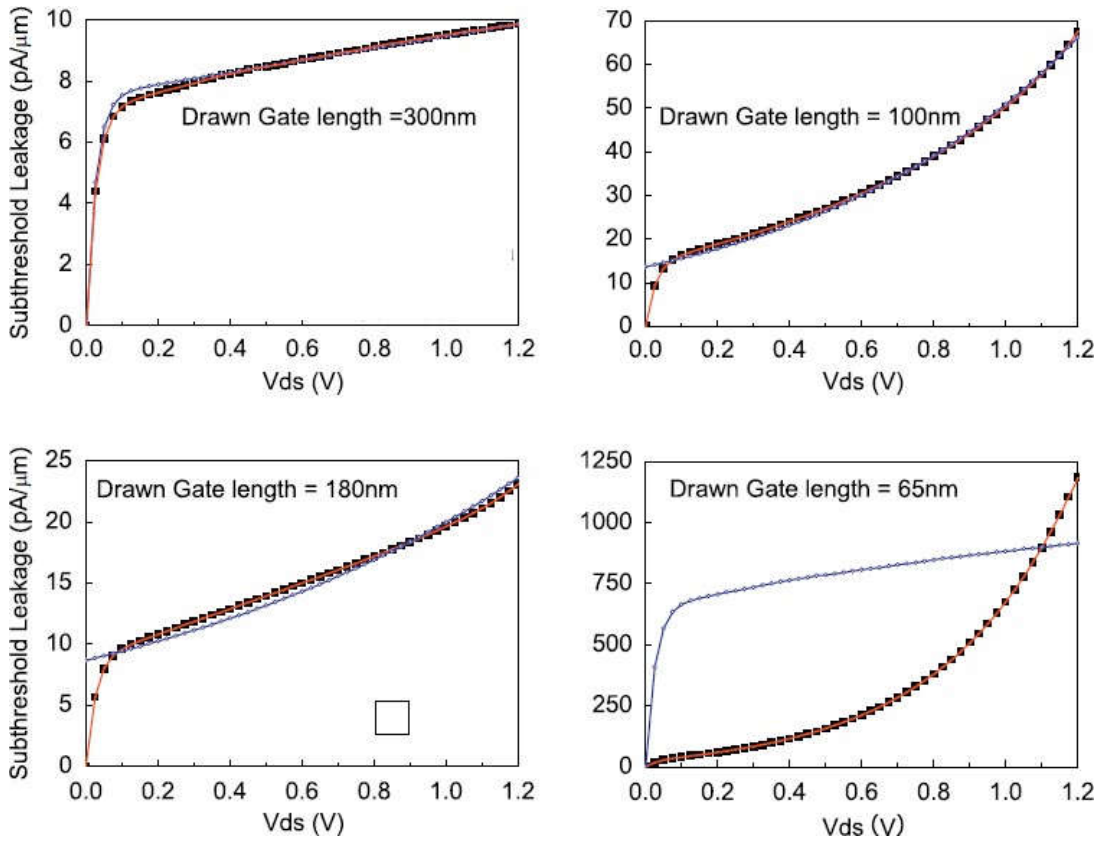


Figure 5-83 Leakage currents across different technologies [J. Xue et al. / INTEGRATION, the VLSI journal 43 (2010) 353–364]

Leakage in short channel devices can come through many things, but the main sources of leakage are subthreshold leakage, gate tunnelling leakage, and reverse bias junction leakage.

About the subthreshold leakage we can see in [66] from (5.29), as we can see as the width of the MOSFET decreases and as the threshold voltage increases the Isub decreases and thus the subthreshold leakage power decreases.

$$I_{sub} = \frac{\alpha_{sub} \cdot \sqrt{q\varepsilon_{si} N_{cheff}(W^2 + \alpha_W W)}}{(V_{ds}^2 + \alpha_{ds1} V_{ds} + \alpha_{ds2}) \cdot \exp(\alpha_{L1} L_{eff}^2 + \alpha_{L2} L_{eff})} \left(1 - \exp\left(-\frac{V_{ds}}{V_T}\right)\right) \cdot \left(2 - \exp\left(-\frac{SA}{SA_0}\right) - \exp\left(-\frac{SB}{SB_0}\right)\right) \cdot \exp\left(\frac{V_{gs} - V_{th}}{nV_T}\right) \tag{5.29}$$

As for the gate oxide tunneling, it occurs for short channel devices because as we scale down the length of the MOSFETs we scale down the oxide thickness too, and at some point the electrons can tunnel through the oxide due to large vertical electric fields, so as the width of the MOS increases the of the gate through which the electrons tunnel increases and thus gate leakage increase, so it is in our favor to decrease the width of the gate not to increase it.

As we have seen, the width of the gate and the threshold voltage are very important for leakage currents, so what we did was, decrease the width of the transistors as much as possible, while keeping in mind the noise and offset specs, and we used hvt-devices in non-critical paths in order to decrease the leakage. The hvt-devices causes extra delay because it allows less current through it due to high threshold voltage, but since we are a low-speed application, speed is not an issue, on the other hand power is the main issue.

5.14 Asynchronous loop design

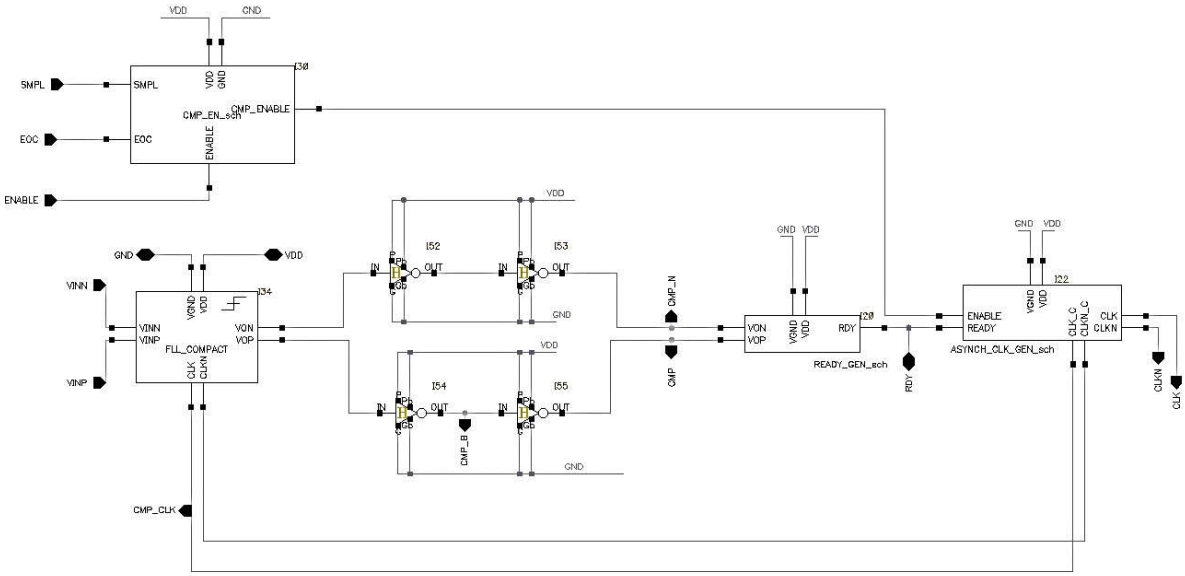


Figure 5-84 Internal design schematic of the comparator

As we have seen in section 5-10, the internal design schematic of the READY-GEN block and the ASYNCH-CLK-GEN block, there has been a few modifications to achieve our desired performance.

First, the CMP-ENABLE-SCH, this block is responsible for enabling and disabling the comparator when it needs to start the operation or when it needs to put the comparator to rest after it has completed 12 cycles, in Figure 5-85.

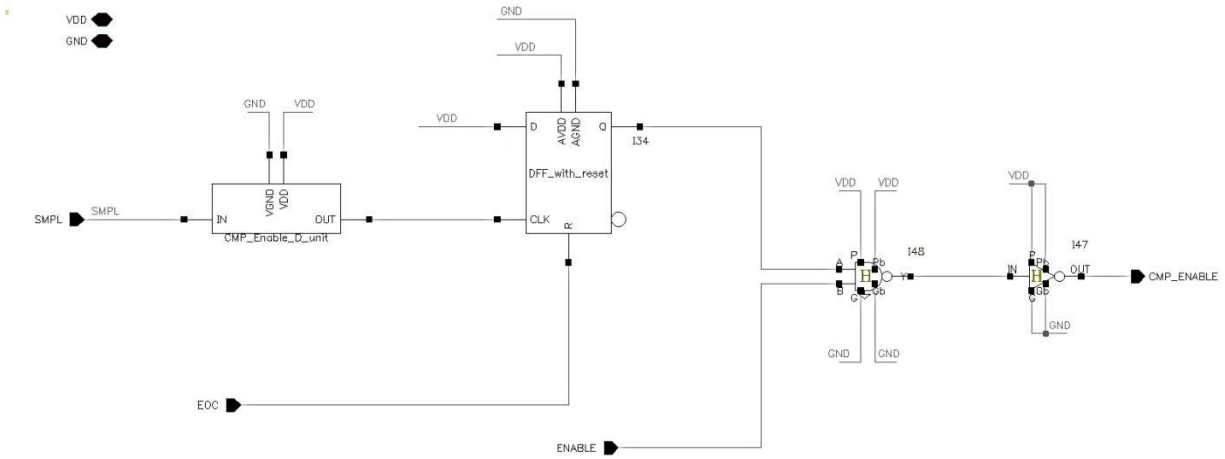


Figure 5-85 Internal design of the CMP-ENABLE

Enable operation :

We can see the internal structure of the block, the ENABLE pin is responsible for the manual enable/disable of the comparator, the input attached to the SMPL pin is the inverted version to the sample signal, when the sample signal goes low, the input to the SMPL of the block goes high (as it is the inverted version), this positive edge is delayed for some time (about 10ns in one of the iterations) to give time for the charges to settle on the DAC caps once the VCM switch on the TOP plate is shut off and VCM is connected the bottom plates, we need to give time for these charges to settle before we enable the comparator, then after the delay the positive edge triggers the D-flipflop to pass VDD, which after passing through the gates will enable the comparator.

Disable operation :

Once the comparator hits the 12 cycle and the EOC (End of conversion) is high, it resets the d-flipflop and causes the comparator enable to go LOW and thus disabling the comparator.

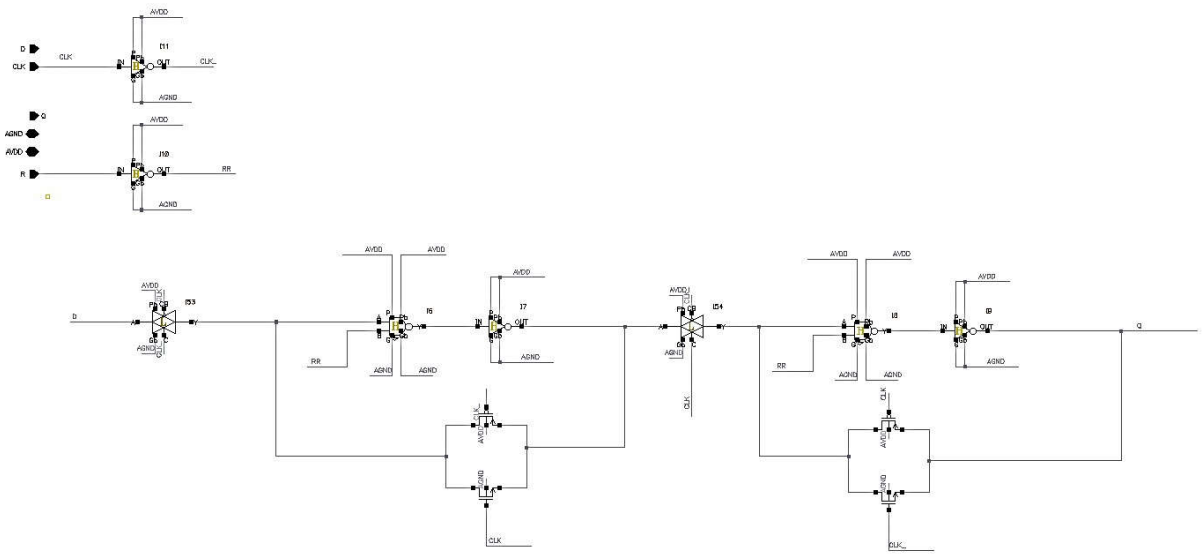


Figure 5-86 DFF with reset internal structure

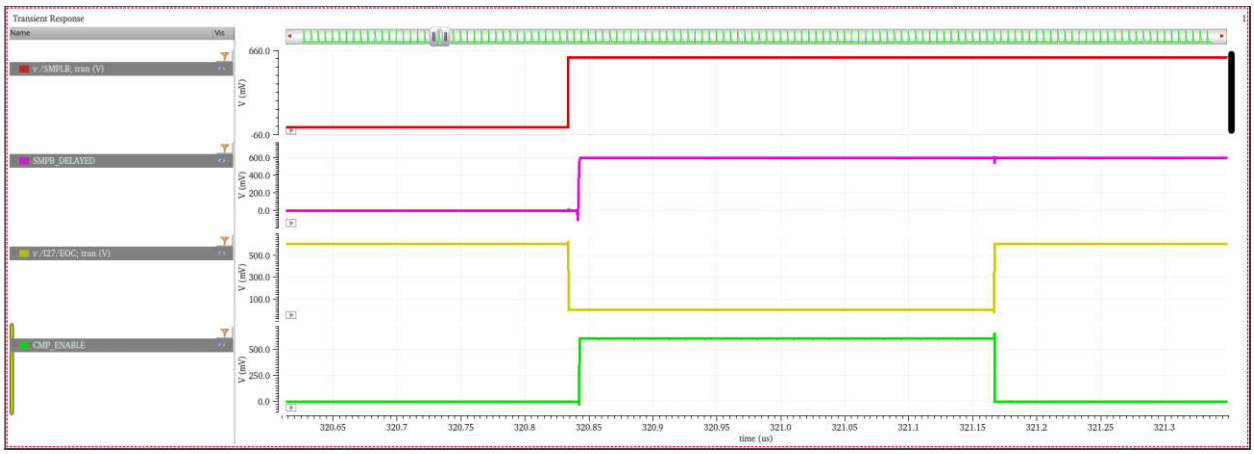


Figure 5-87 Wave forms of comparator enable block

Secondly, the four inverters that can see directly after the FLL comparator, are used to suppress the glitch that occurs specially in LSBs because the inputs to the comparator are close to each other, the amplification phase takes long time and thus the two outputs approach the VDD/2 before the latch regeneration takes place, this glitch for the LOW output might cause some circuits in the logic block to turn on specially with corners and process variations. The solution to this problem initially had to be, increase the size of the lower latch regenerative transistors so as to the speed up the process of pulling down the LOW output in short time even with small input differential, but we could not do that here, this method got us only so far where the smallest glitch size that we could get was 170mv but we increased the power of the LATCH drastically in the process, so we had to place these inverters to prevent the glitch from going into the system, we

made these inverters large sizes and hvt to make them robust against the glitches and be able to deliver good table outputs to the rest of the system.

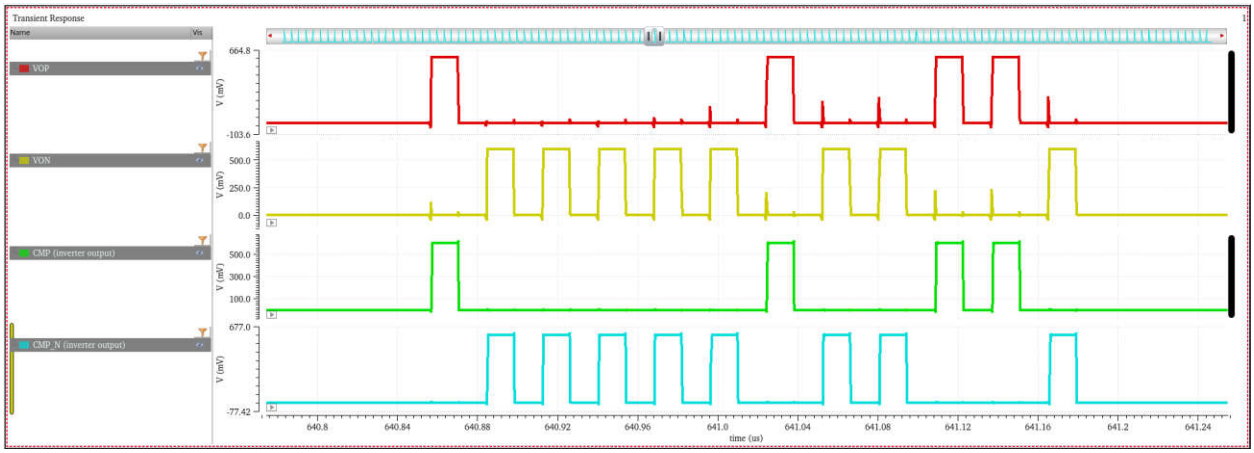


Figure 5-88 Wave forms showing inverters operation

As for the RDY-GEN, as in Figure 5-89, when the comparator resorts its outputs one of them is high and the other is low, it generates the ready signal which pushes the next clock cycle.

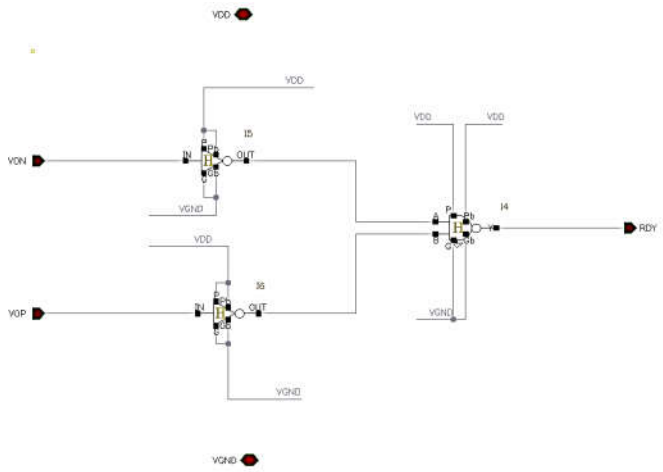


Figure 5-89 RDY schematic

The final piece of the puzzle is the Asnych-clock-generation block,

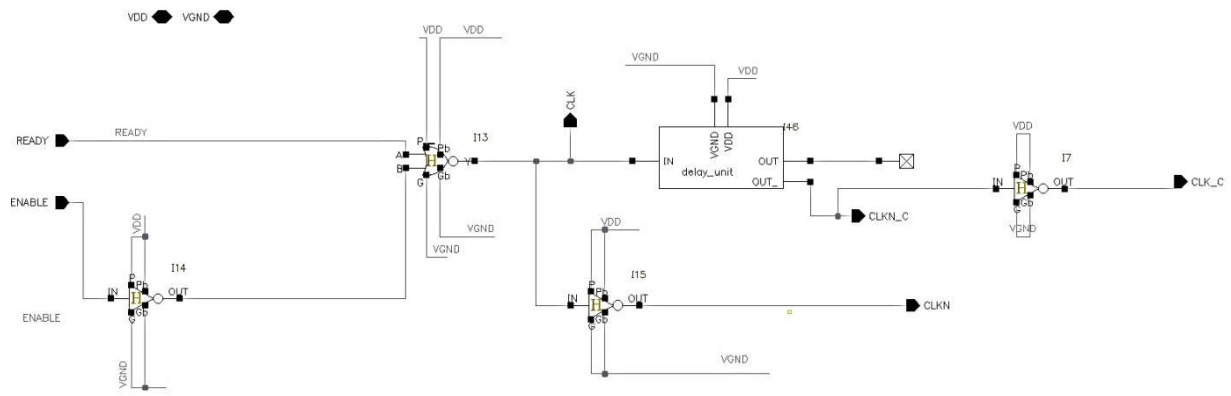


Figure 5-90 Asnych-clk-gen schematic

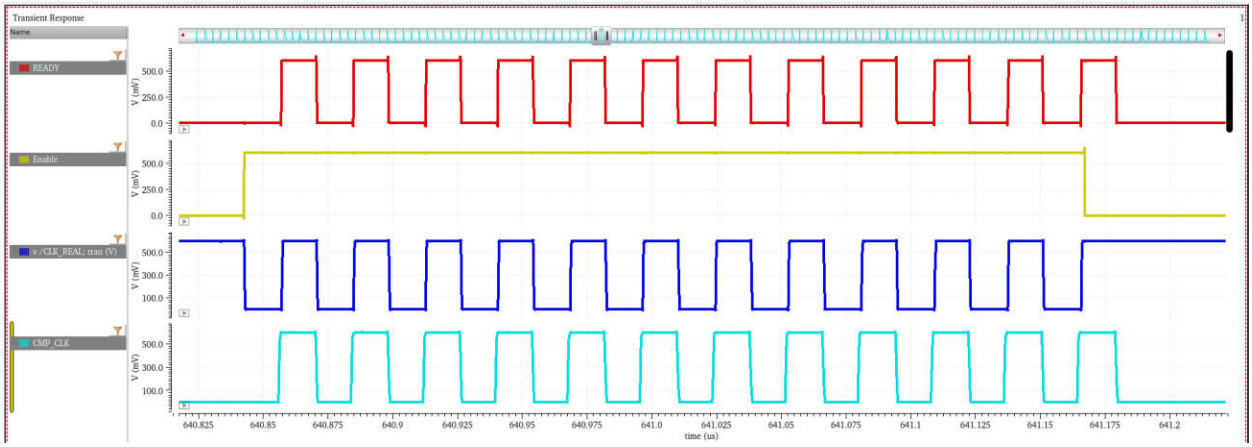


Figure 5-91 Asnych-clk-gen Waveform operation

As we can see here, the READY signal starts off at LOW because the comparator outputs are both LOW, and the Enable initially is LOW too, the CLK is low and CLKN is high, it is important to mention that we operate on CLKN as our major SAR_CLK, then the Enable comes high, the CLKN goes low, and after some delay which is about 14ns the CMP-CLK goes high enabling the comparator and setting READY signal after the comparison is complete, the READY signal causes CLKN to go high which enables the LOGIC blocks and completes the SAR operation and the process goes on and on until the EOC disables the comparator.

What we have discussed so far is the full operation of the asynchronous clock generation loop for the fast low leakage comparator and now we proceed to the next section.

5.15 Finalized comparator results

Some of the results that we have not discussed so far, something like the propagation delay of the comparator, the propagation delay differs depending on the differential input on the comparator, so for example the comparator would take longer time to evaluate an LSB difference at the input than to evaluate an MSB difference for example. It all depends on the current steering capability of the Preamp, where it would long time to steer the current to one side over the other if the differential input favors this side, we did simulations on 200mv difference and on LSB difference, the results are shown in Figure 5-92 and 5-93.

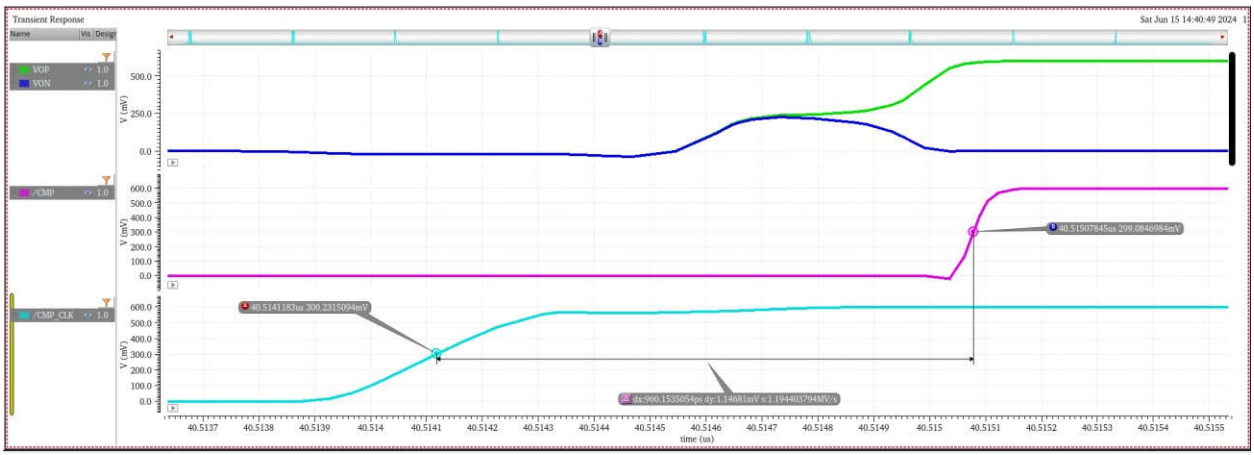


Figure 5-92 LSB difference on the input, observe how it takes longer time to evaluate and how the glitch that we talked about occur

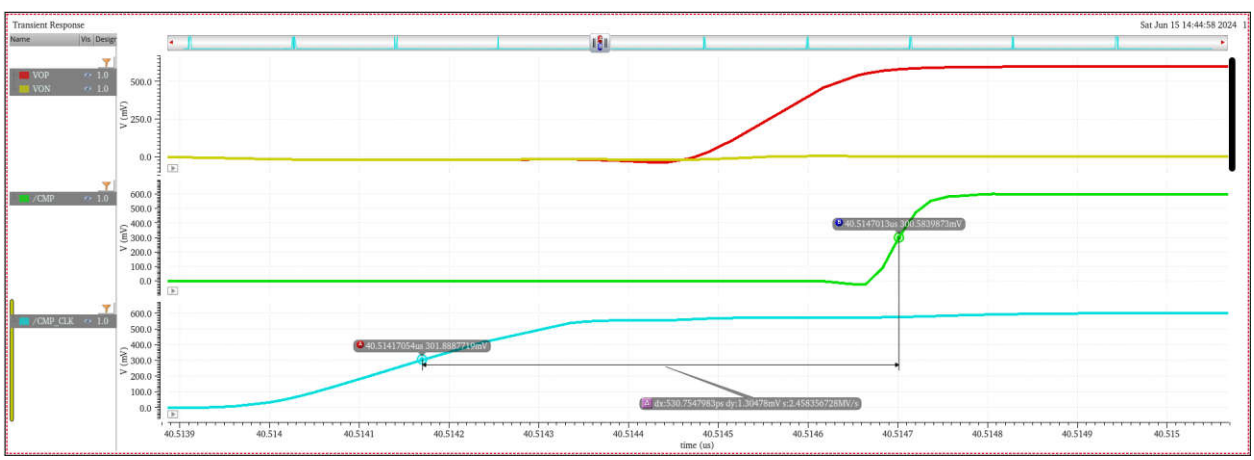


Figure 5-93 200mv difference on the input, observe how it takes shorter time to evaluate and how the glitch that we talked about does not occur

As for the temperature variance, as we increase the temperature the transistors driving capability increases as the charge carriers carry more energy and currents increase resulting in less delay.

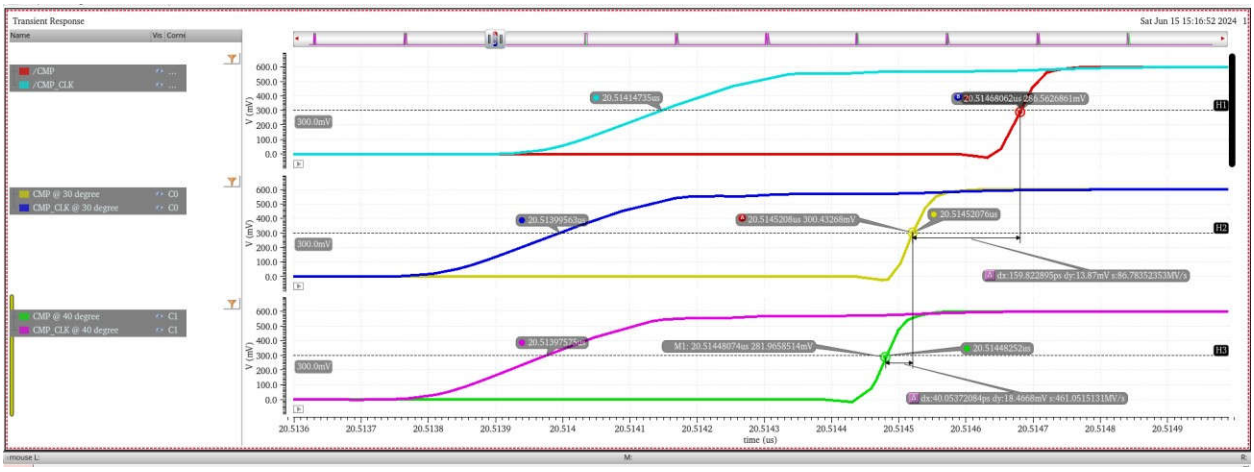


Figure 5-94 the higher the temperature, the faster the system, and the less the delay

The overdrive recovery test for the FLL is also made :

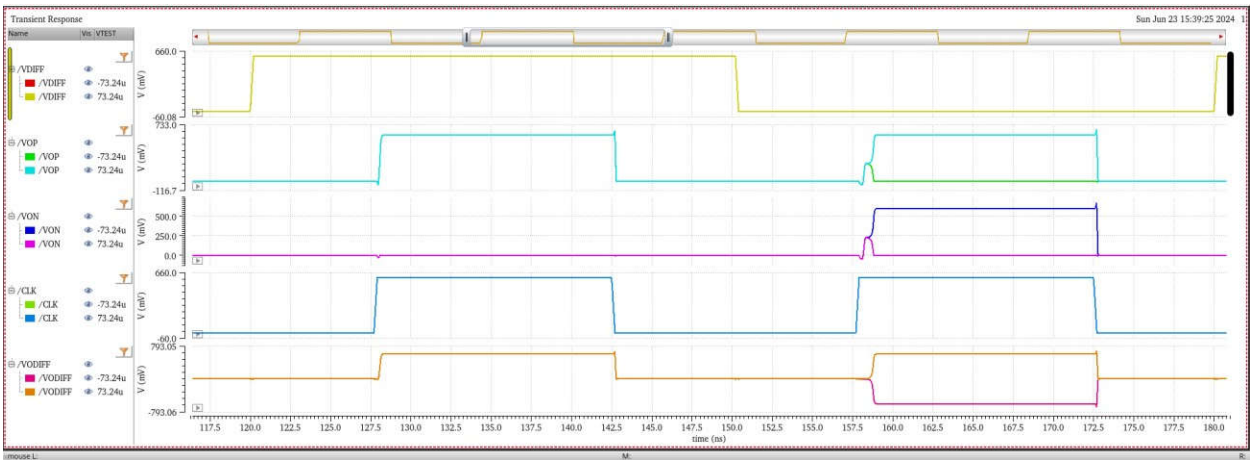


Figure 5-95 Overdrive recovery test for the FLL

Finally, the kickback noise coming from the charge injection and clock feedthrough couples through the Cgd (gate drain capacitance) of the input pair, upon integration (assembly) of the comparator with the DAC, when the CMP-CLK goes from low to high, the caps on the input pair start to charge, as the transistors are switched on (in subthreshold region) this cause a drop in the DAC voltage and this drop depends on the input difference whether we are on the LSB or the MSB, also the effect when the CMP-CLK rises transistors turn on and caps charge causing a drop on the

VDACP and VDACN and when the CMP-CLK falls the transistors turn off and the caps discharge causing rise on the VDACP and VDACN terminals, in the MSB the difference between VDACP and VDACN is too large that it creates asymmetries in the formed parasitic caps on the input pairs and thus the drops are not the same, one the other hand the LSB the input difference is small and thus the difference between the formed caps is small and the drop or the rise is nearly equal.

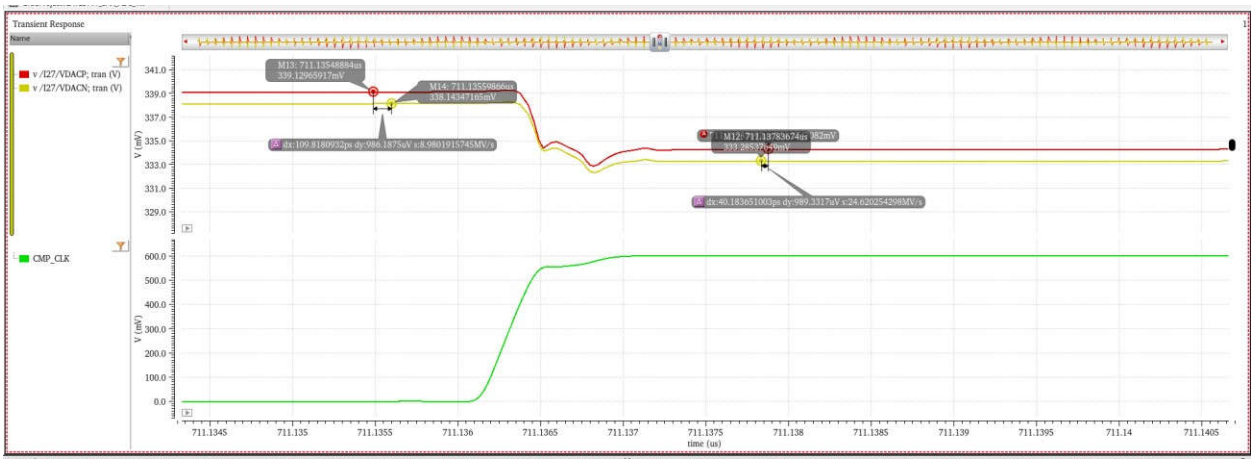


Figure 5-96 CMP-CLK goes high as we discussed causing drop in the input terminals, and since this is an LSB difference the drop is symmetrical

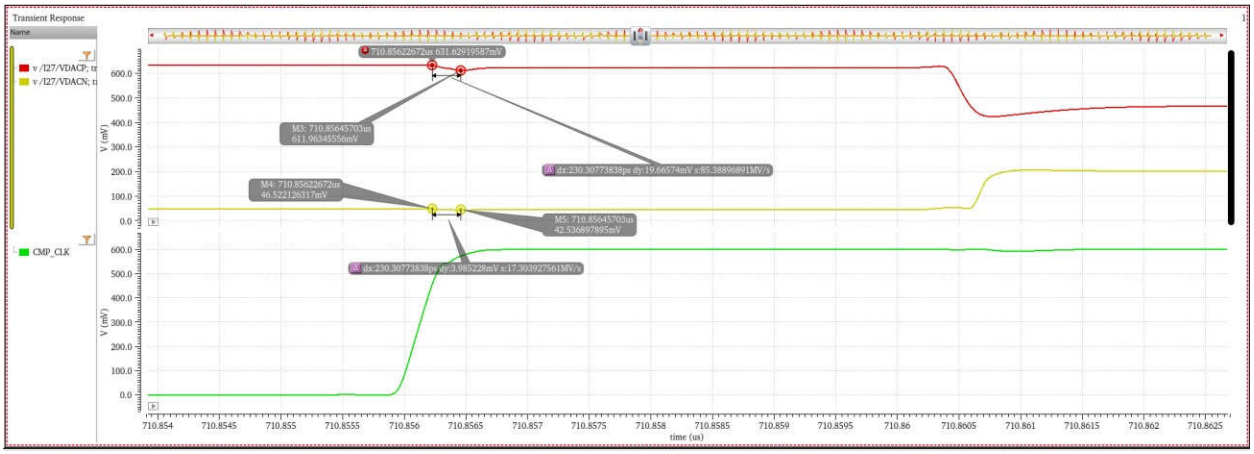


Figure 5-97 CMP-CLK goes high as we discussed causing drop in the input terminals, and since this is an MSB difference the drop is asymmetrical, the DACP drops by 19mv while VDACN drops by about 4mv only, this can cause tremendous errors with noise included.

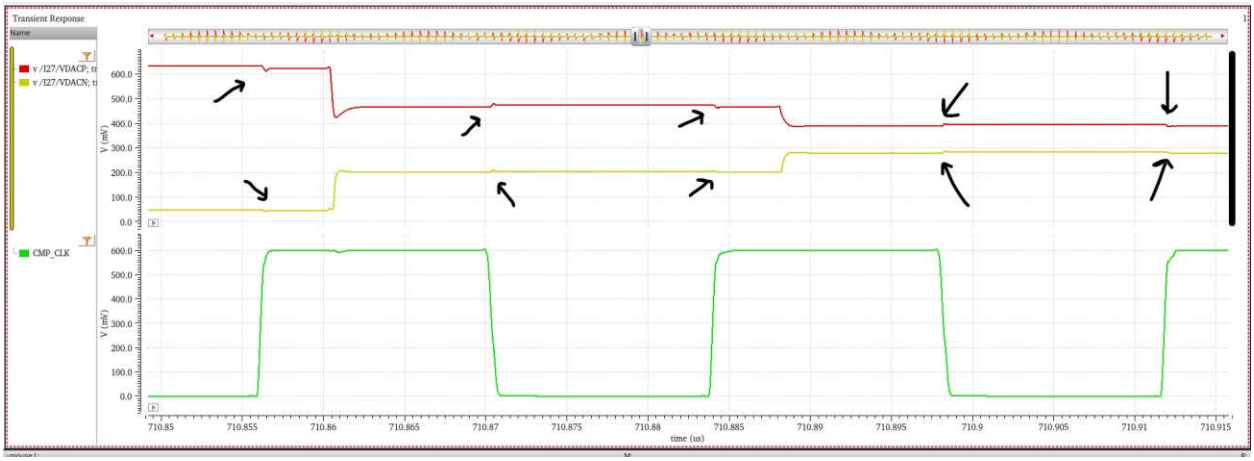


Figure 5-98 This figure shows the drops and rises in the DACP and DACN terminals corresponding to the rising and the falling of the CMP-CLK respectively.

5.16 Layout

The Layout phase of the comparator started right after finishing the schematic view of the comparator and integration with the DAC has brought great results. Following the standards of layout rules we were able to implement the Layout of all the blocks of the comparator and integrate them together and test its performance. We will discuss each block separately.

FLL LAYOUT

The FLL contains two main parts the Preamp and the Latch, starting with the Preamplifier, it has an input pair and a PMOS load, since the input pair is very sensitive to mismatches, we made two formations one with linear common centroid **ABBAABBA**, the common centroid technique reduces the effects of process and thermal gradients, we could have made cross bi-quad common centroid but the connections would have been difficult and would create routing mismatches and asymmetries, the second formation is interdigitation with dummies in the form of **DAABBAABBD** to reduce the edge effects, we did both of them, but as a preference the first formation is way better as common centroid is much better for mismatches and process variants, also dummies on the input pairs would increase the capacitive loading which is not helpful in our application.

As for the PMOS load, we also implemented it in a common centroid technique of **XYYXXYYX**, also to protect the device from process variants and to create symmetry in the device as if any mismatches were to occur it would affect all equally, of course we did all this using multi-fingered devices, each transistor with 4 fingers enabled us to do that.

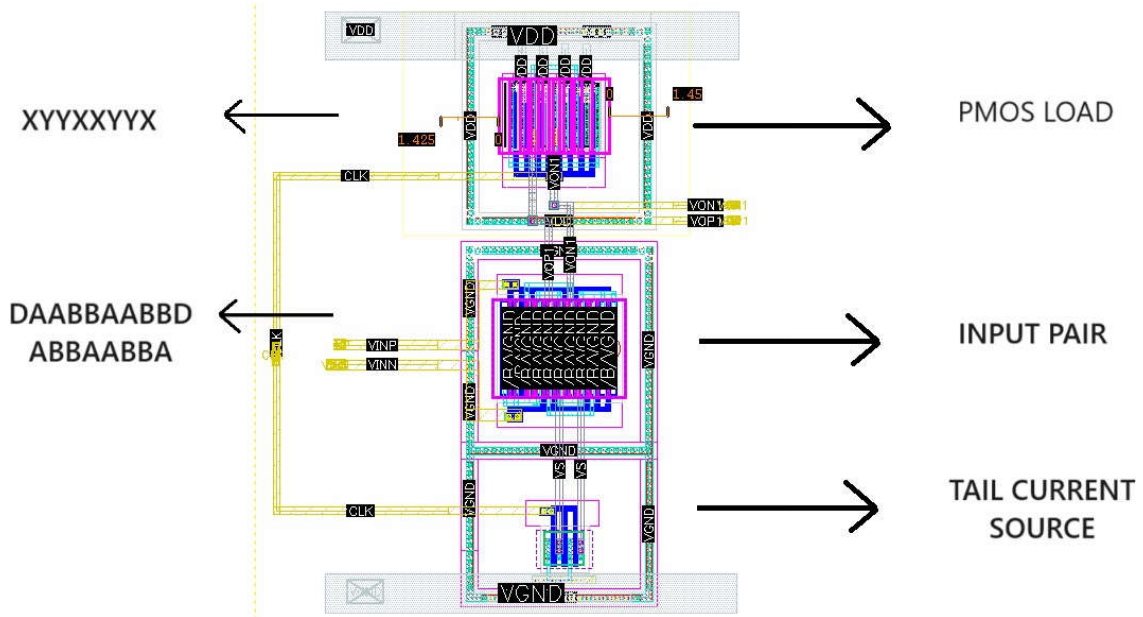


Figure 5-99 Layout of the PREAMP, the clock is injected in a way so that it would reach the PMOS load and the Tail current source at the same time.

Of course this is just the initial view, there is huge waste of area here, but this is just to show the idea, we will show the compacted version later on.

Now the Latch, the latch was made with the same techniques, where the input pair was made by Bi-Quad cross common centroid techniques to avoid any mismatches and to overcome process variants and inherent mismatches also to instill symmetry.

The PMOS load above was made with common centroid **ABBAABBA** technique, the input pair was made $\frac{ABBA}{BAAB}$ cross-quad assembly to instill symmetry and robustness, as for the latch regenerators (C,D) and the switches (A,B), each transistor had two fingers, so we employed common centroid technique while protecting the latch regenerators from edge effects by putting them inside and enclosed them by the switches as in **ABCDDCBA**, observe Figure 5-101 to see the latch, we also added guard rings here in the initial preview, although it is not done in the full compact design because it separates the blocks and consumes large areas, its advantage is that, if there is any noise in the substrate, it will not flow to the next circuit, on the contrary, it will flow to the ground

through the guard ring connection, so it helps isolate the noisy blocks from the quite ones as seen in figure 5-100.

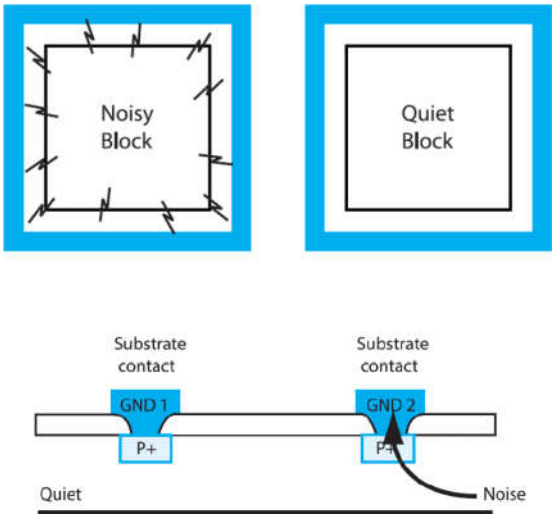


Figure 5-100 Guard rings isolation principle

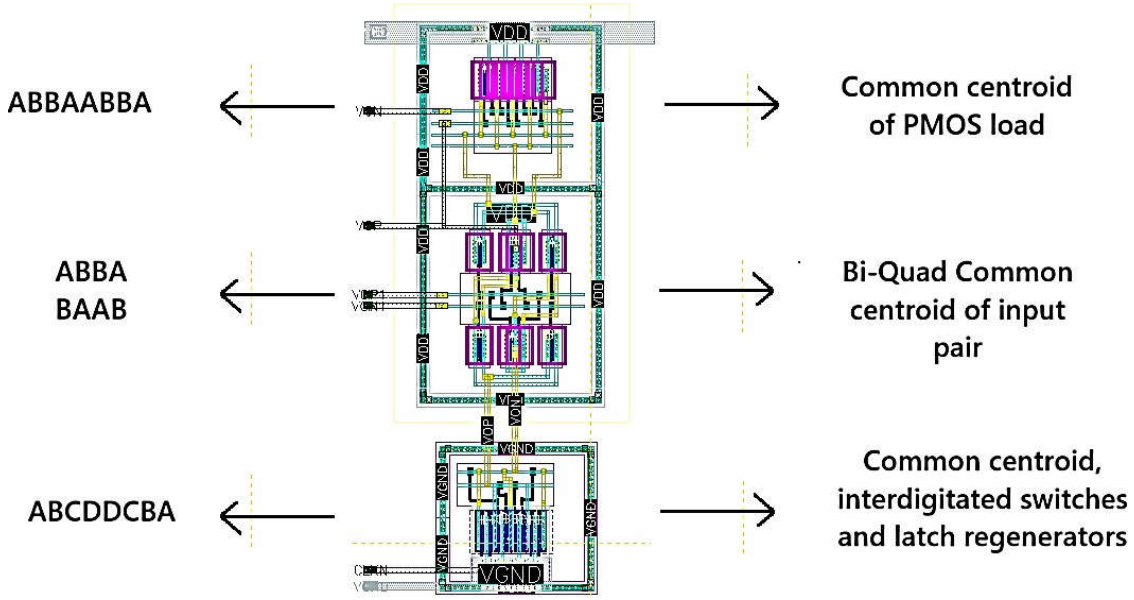


Figure 5-101 Layout of the latch, the clock is injected in a way so that it would reach the all the switch fingers at the same time.

Now we can see the compacted version of the FLL :

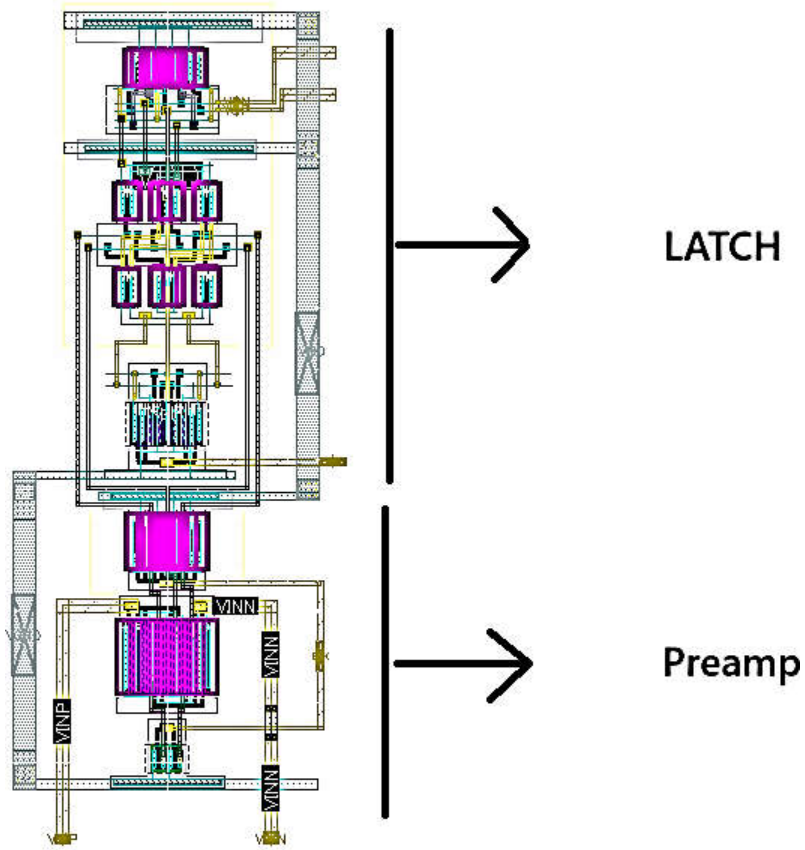


Figure 5-102 Layout of the complete FLL block

We went on and tested both of these approaches, the results were great, degradation in ENOB was only about 0.2 or 0.1 which could even be simulation accuracy.

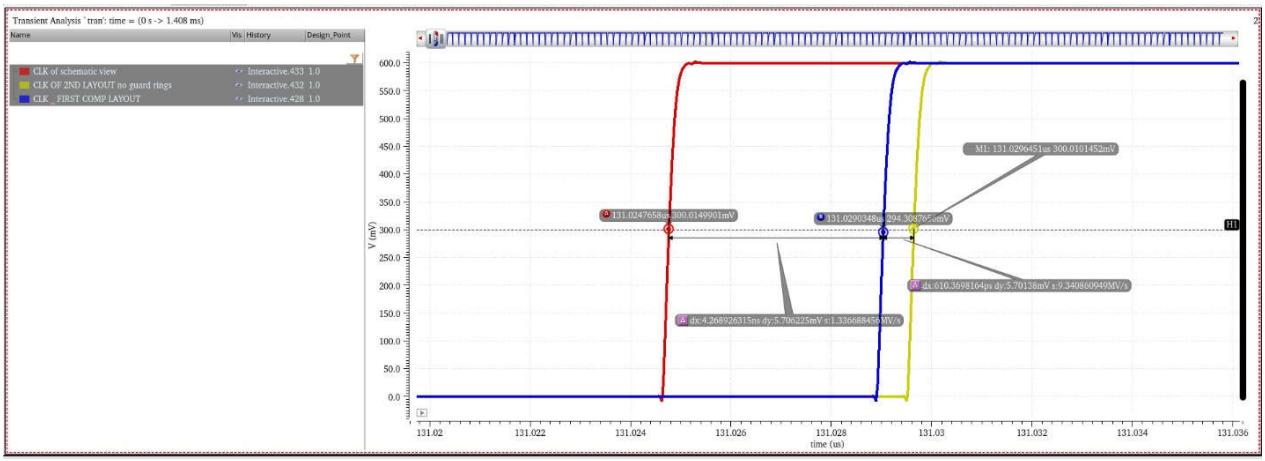


Figure 5-103 Comparison of CLK showing the extra delay caused by the Layout models

Now we look upon the Comparator enable block :

The two major blocks inside are the d-flipflop and the delay unit, the d flipflop is as follows, observe Figure 5-104.

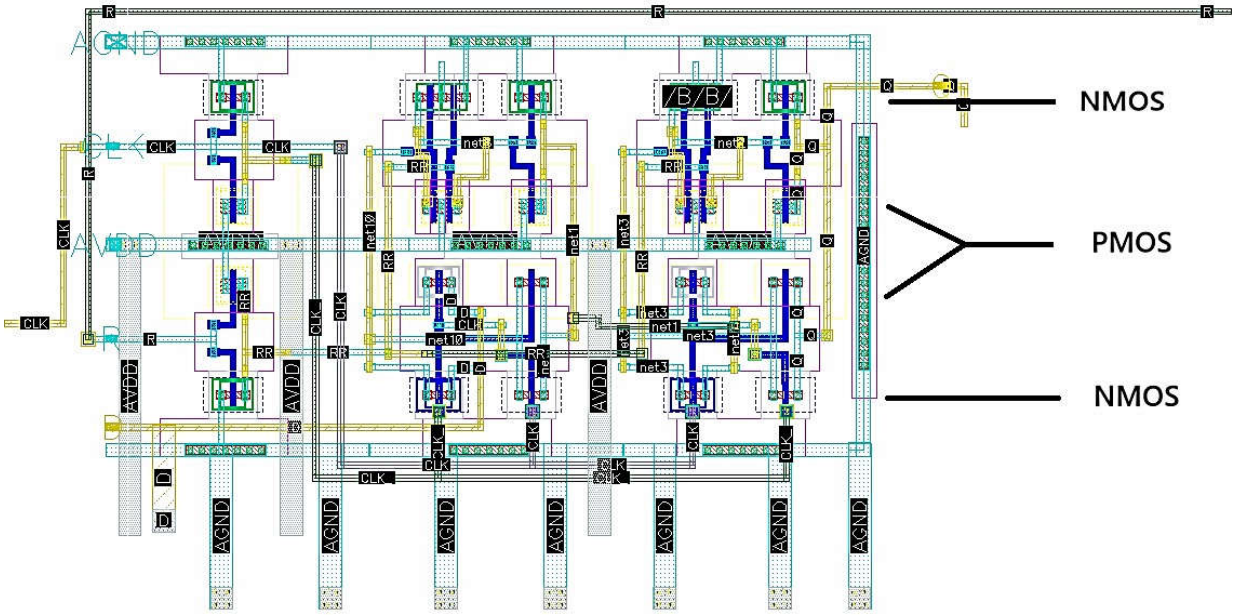


Figure 5-104 D-flipflop block inside the comparator enable block

Now the delay unit, it was implemented in a serpentine form in order to take up the least possible area while still providing the required delay.

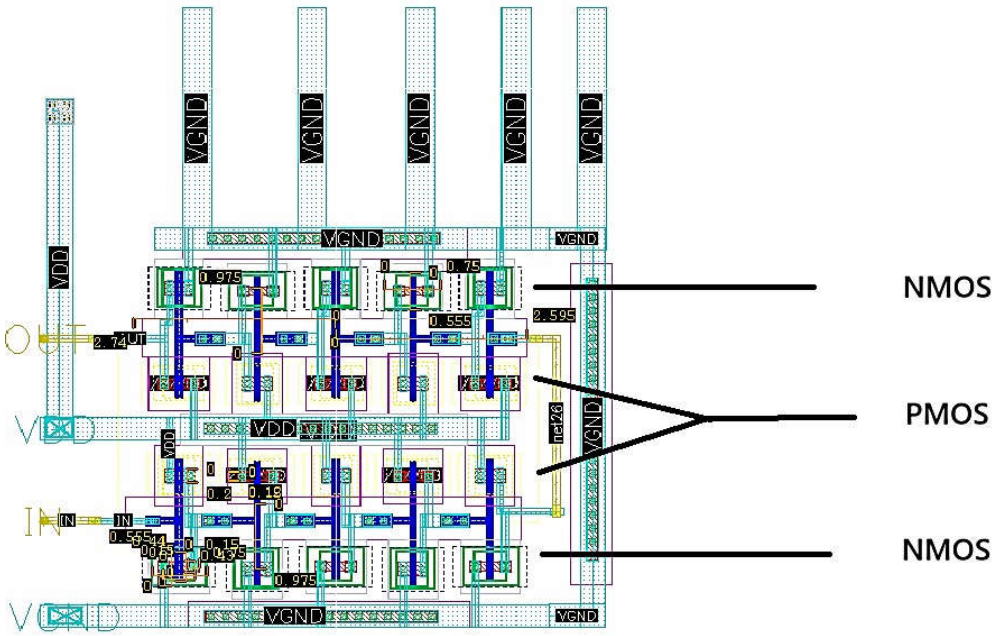


Figure 5-105 Delay unit block inside the comparator enable block

Now we add these two pieces together and create the full CMP-enable Layout :

Observe Figure 5-106 and Fig 5-107

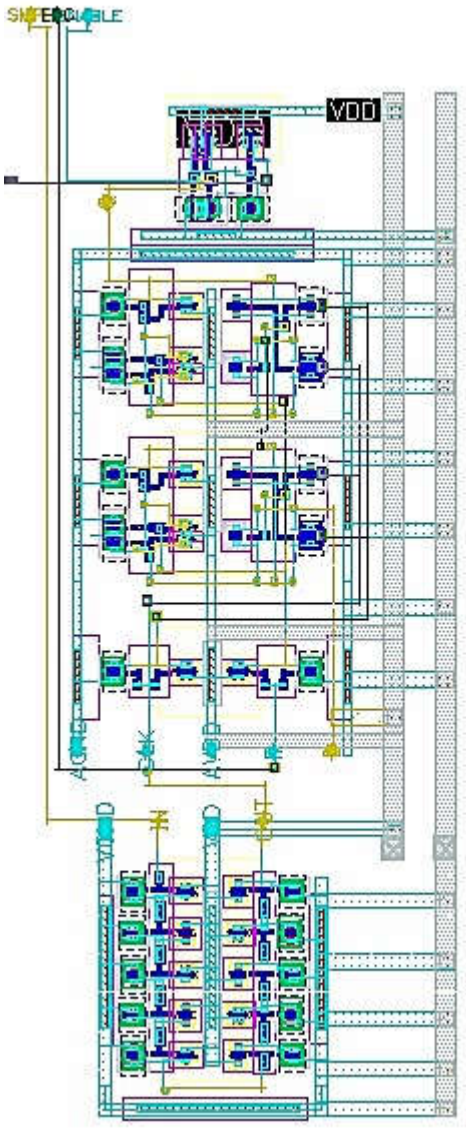


Figure 5-106 CMP Enable Layout

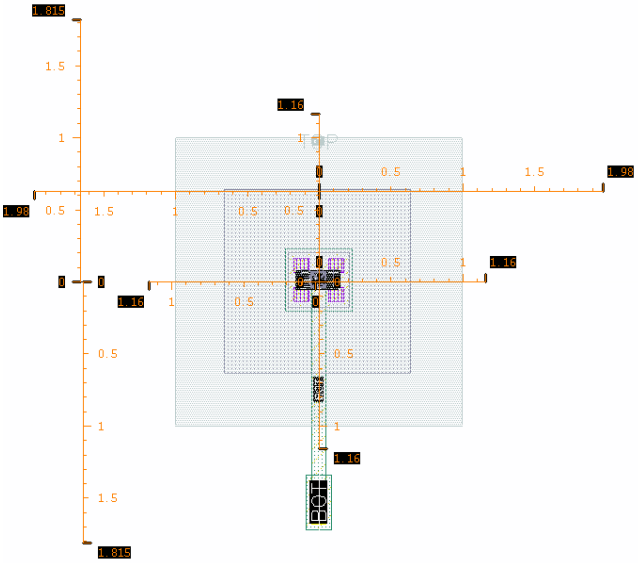


Figure 5-109 Vertical capacitor of 500aF

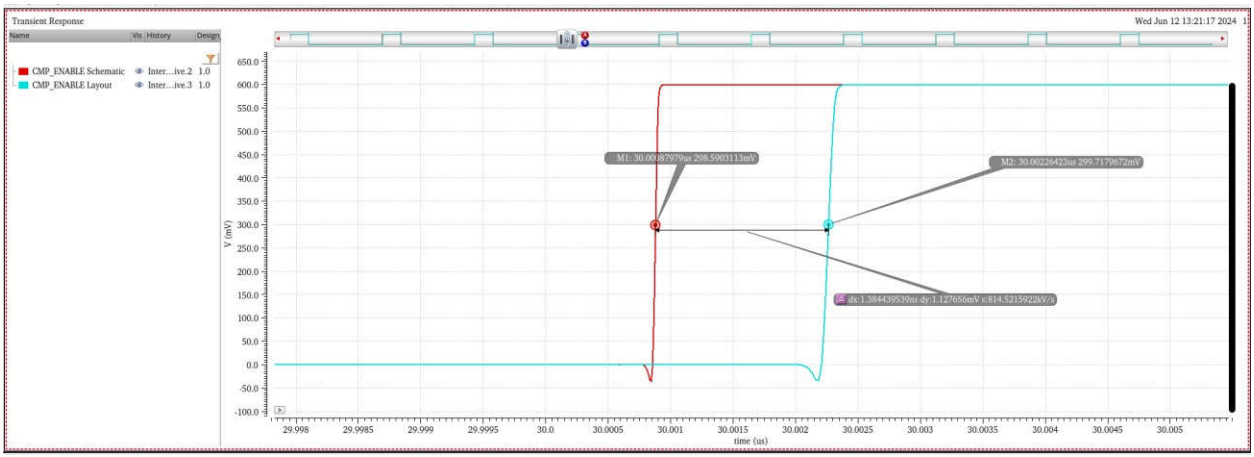


Figure 5-107 CMP Enable Layout and schematic delay differences of 1.38ns

Now the asynchronous CLK generator, the main component is the Delay unit which contains caps as well as placed in serpentine geometry.

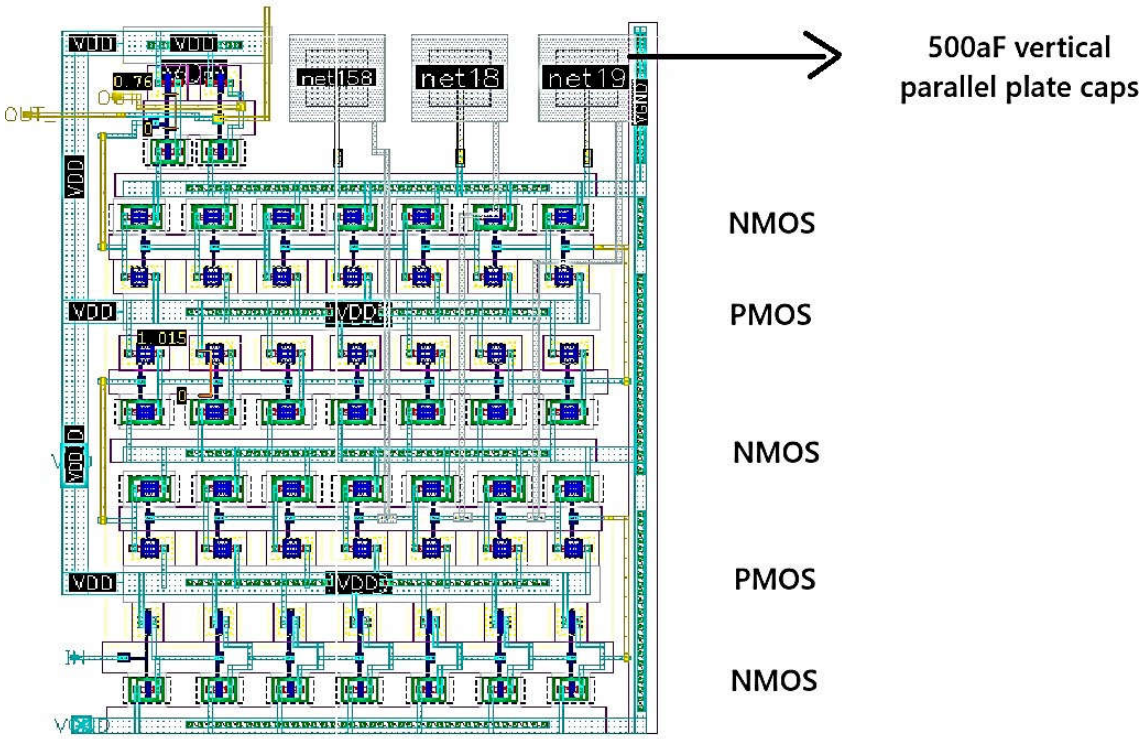


Figure 5-108 Delay unit of the Asynchronous CLK generator circuit

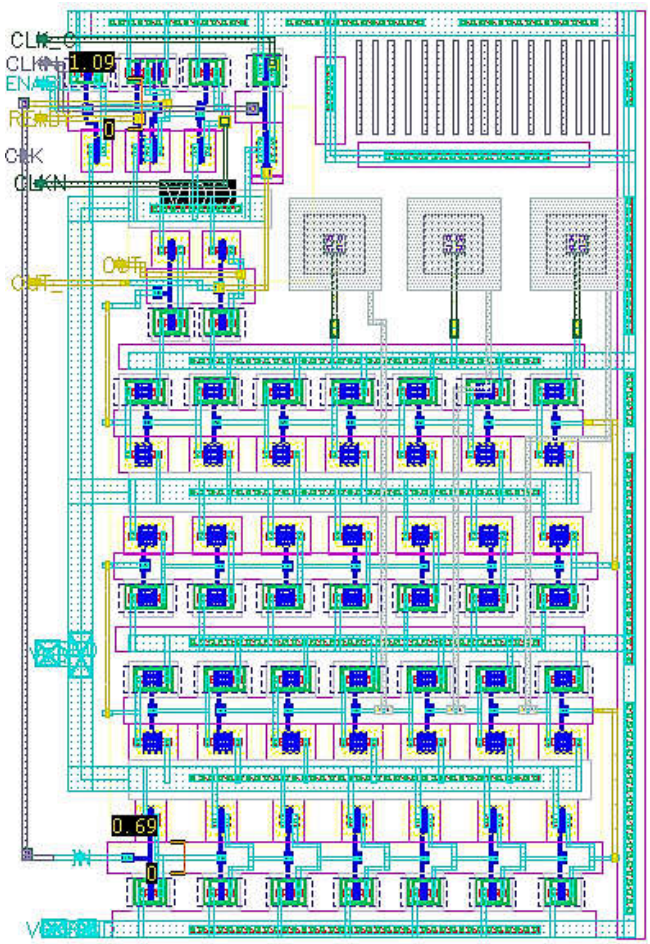


Figure 5-109 Layout of the Asynchronous CLK generator circuit

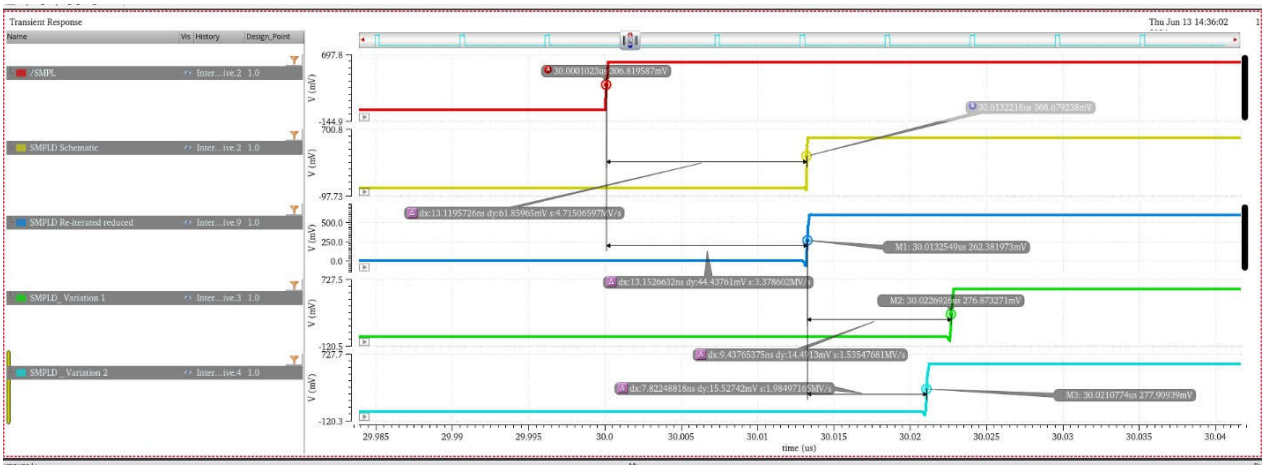
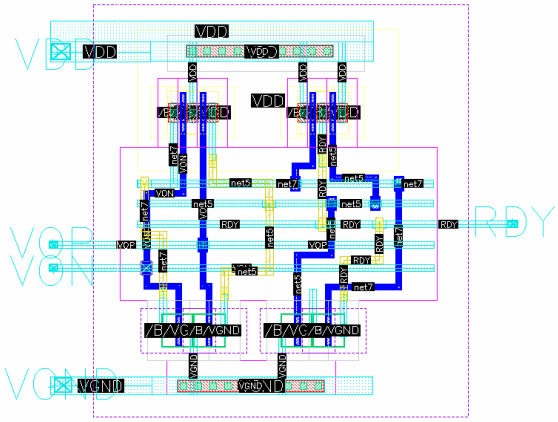


Figure 5-110 Waveforms of delays until we reach the closest to the schematic

And now the remaining part is the READY-Signal generator circuit :


VDD
VDD
VDD

Upon testing the comparator chip, it only showed a degradation of 0.4 ENOB from the schematic view, since we tried to mimic the schematic view in our design.

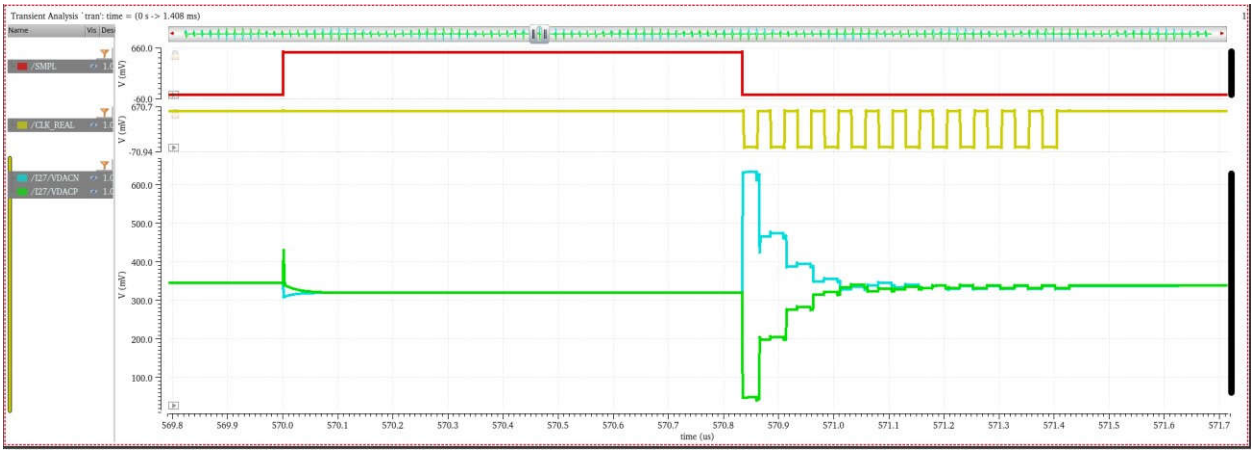


Figure 5-113 Post layout simulation of the comparator showing a very good SAR operation with degradation in ENOB of 0.4

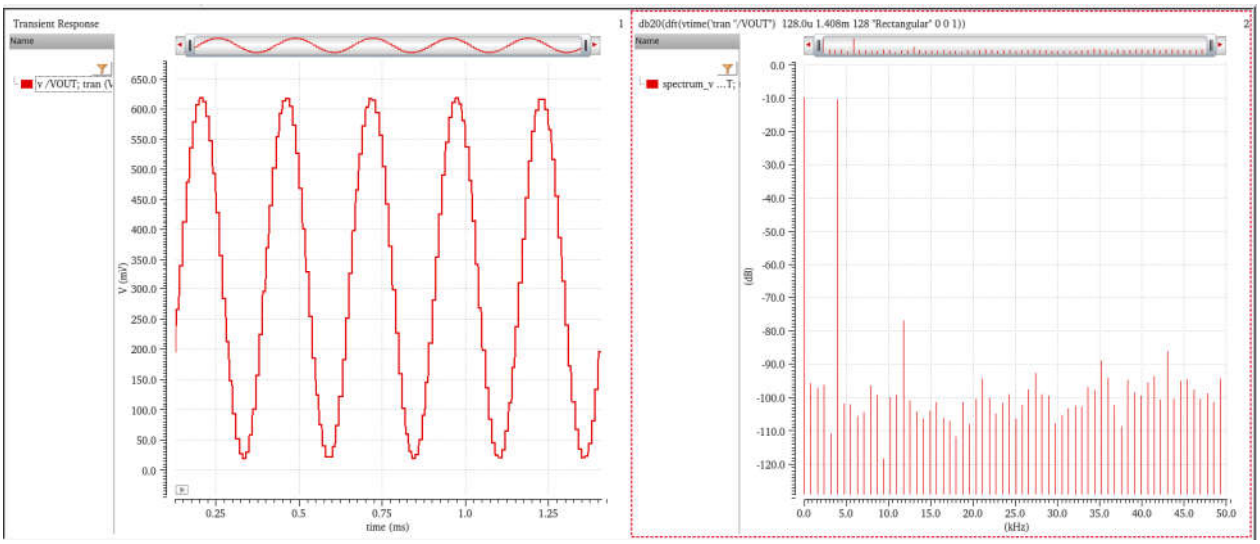


Figure 5-114 Post layout simulation of the comparator showing a very good SAR operation with the PSD of the output signal on the left

5.17 DDNR

Here we are at the final section of our chapter, this section introduces "Data driven noise reduction technique", a technique which is used in the comparator assembly to get better noise performance at the expense of some power.

Besides designing an energy-efficient comparator, we can also take advantage of SAR ADC's architectural level property to reduce the total comparator power. For an N-bit SAR ADC, its comparator needs to fire N times, but the noise requirements for different comparisons are not the same. With a small comparator input, the comparator noise needs to be low to ensure a correct decision. However, with a large input, the comparator noise requirement can be significantly relaxed while still producing the correct decision. Notice that out of the N comparisons, the comparator input is less than LSB/2 only once. For all other comparisons, the noise requirement can be loosened to save energy. Nevertheless, the challenge is that we do not know when the critical decision will happen, as it depends on an unknown input. Researchers have developed several techniques to address this concern.

In the chosen paper [69] , it states that in high-resolution ADCs, which are limited by thermal noise, the power consumption of the analog components will increase by a factor of 4 for each additional 6 dB of SNDR (equivalent to 1 bit gain in ENOB). For that reason, it is difficult to maintain the excellent power efficiency of SAR ADCs when extending the resolution beyond 10 bits. Considering the analog blocks inside the ADC, the comparator usually dominates the power consumption. In view of this, a method is presented here that scales down the comparator noise in a much more power-efficient way as compared to the conventional approach. Before this method is explained, a digital majority-voting technique is discussed as an alternative to analog scaling.

A. Majority voting

Consider a comparator with a certain input-referred noise, denoted by its standard deviation σ_{noise}, Due to this noise, the comparator has a certain output probability function, as shown in Fig. 5-115, where P0 denotes the probability of output '0' as function of the input voltage and P1 denotes

the probability of output '1' as function of the input voltage. Note that a differential implementation is assumed here, with a comparator reference voltage at Vin=0. P0 and P1 are determined by the Cumulative Distribution Function (CDF). Assuming a Gaussian distribution of the input-referred noise, the CDF is given by:

$$\mathrm{CDF}(V_{\mathrm{in}}) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{V_{\mathrm{in}}}{\sqrt{2\sigma^2_{\mathrm{noise}}}}\right)\right] \tag{5.30}$$

And P0 and P1 are given by :

$$P_0(-V_{\mathrm{in}}) = P_1(V_{\mathrm{in}}) = \mathrm{CDF}(V_{\mathrm{in}}) \tag{5.31}$$

From P0 and P1 , the error probability Pe (i.e., the probability that the comparator takes the wrong decision) can be deducted as:

$$P_e(V_{\mathrm{in}}) = \mathrm{CDF}(-|V_{\mathrm{in}}|) \tag{5.32}$$

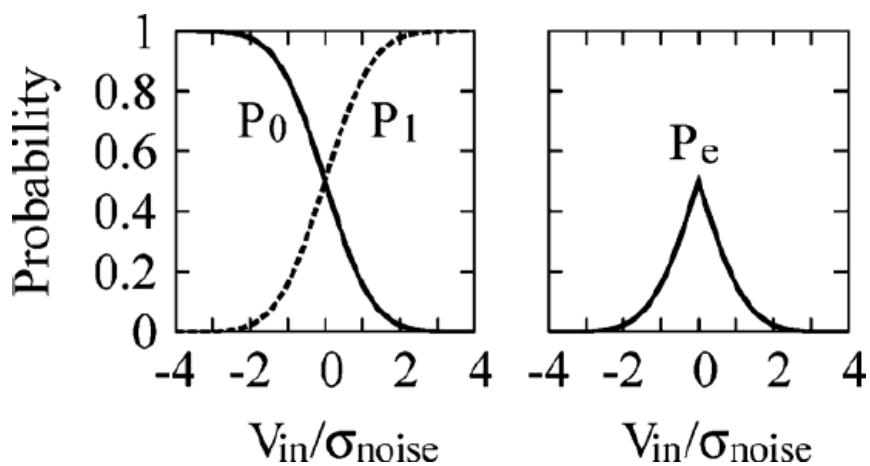


Figure 5-115 Output probabilities of P0 and P1 and error probability of the Pe of the comparator [69]

For illustration, Pe is also shown in Fig. 5-115. As can be seen, Pe approximates zero for $|VIN| >> \sigma_{noise}$, which logically means that the comparator makes no mistakes when the input signal is sufficiently large as compared to the noise level.

When the same comparator is used several times to repeat the same decision, a majority-voting logic can be applied on the obtained output samples to decide the final output. Using more samples increases the probability that the majority decision is correct.

As an example, Fig. 5-116 shows the error probability of an initial comparator, as well as the performance after 5x voting (Pe,5v). For reference, the error probability using 4x analog scaling is also shown (Pe,4v).

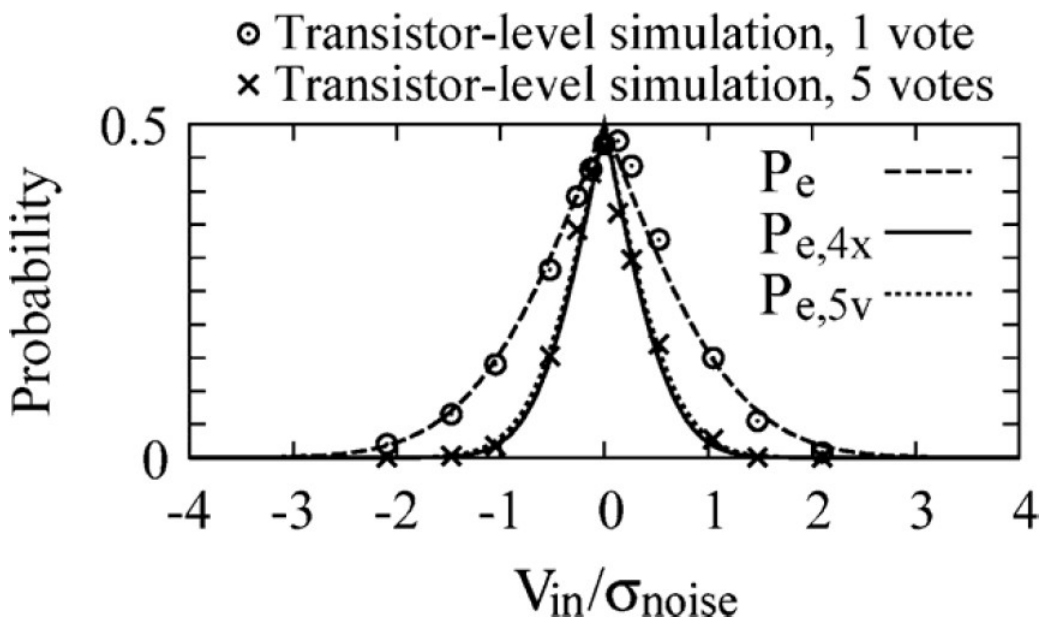


Figure 5-116 Error probability of an initial comparator Pe, after 4x analog scaling (Pe,4v) or after 5x majority voting (Pe,5v) . Also shown are transistor-level transient-noise simulations of a comparator without voting and after 5x voting. [69]

It can be observed that in this example the analog scaling and majority voting are similar in error performance. For the voting scheme, it is not always necessary to determine all 5 samples. E.g., if the first 3 votes are all equal to '0', the last 2 samples can be omitted as the majority is already clear, thereby saving power. In addition to the theoretical model, the statistical results from a transistor-level transient-noise simulation (including 1/f noise) of the actual comparator used in this work are also shown in Fig. 5-115.

The comparator has an input-referred noise of σ_{noise} = 305uV, which is reduced to 163 uV after 5x voting in agreement with the theoretical model. In this implementation, the comparator is a fully dynamic circuit. This ensures uncorrelated votes as potential memory effects and 1/f-noise are suppressed by the dynamic operation.

Analogous to the example for 5 samples, one can determine the error probability and the corresponding effect on σ_{noise} for any given number of samples. This is illustrated in Fig. 5-117, which confirms that voting on 5 samples will reduce the effective noise to about 50% of the original value, while voting on 25 samples will decrease it to about 25%, similar to 16x analog scaling. As can be seen, the theoretical model closely matches the transistor level simulations.

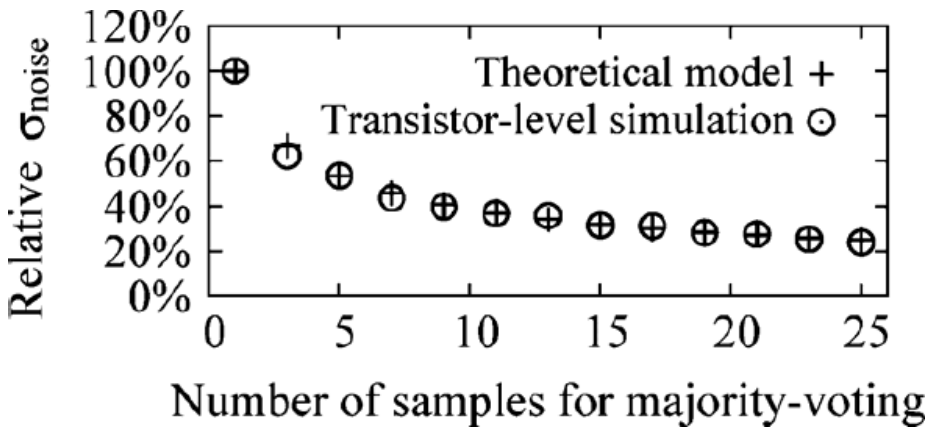


Figure 5-117 Relative equivalent comparator noise as a function of the number of samples in the voting scheme, using the theoretical model and transistor-level transient-noise simulations. [69]

In conclusion, the power consumption of the majority-voting scheme with 5 samples is similar to the power consumption of the equivalent analog-scaling method , while the error probability is similar in both cases too. Also note that in an actual SAR ADC, the majority voting does not only reduce the impact of the noise from the comparator, but also reduces the impact of the noise from the feedback DAC. However, this work focuses on the comparator noise, which is more critical for the overall performance.

B. Data-driven Operation

So far, the majority-voting procedure does not yet have a major impact on the achievable ADC performance. However, it offers flexibility as the amount of noise reduction can be dynamically adjusted simply by changing the number of samples used to determine the majority decision. This can be used advantageously as the noise requirement for the comparator is not identical for each bit-decision in the SAR algorithm: only in one comparison cycle is the amplitude of the input signal Vin to the comparator less than 0.5 LSB. Similarly, there is only one cycle in which |VIN| is between 0.5 LSB and 1 LSB, one case where it is between 1 LSB and 2LSB, and so on. Thanks to this known variation of the amplitude of Vin , the noise requirement for the comparator changes correspondingly: the requirement is most critical in one case only, while it becomes more and more relaxed for the other cases. This knowledge suggests that one could save power by adjusting the

comparator noise performance depending on the amplitude of Vin . Thus, a technique is needed to provide flexible noise-performance, and another technique is needed to estimate the amplitude of Vin [69] .

Flexible noise-performance can be implemented conveniently based on majority voting [69] : the voting scheme can be instantaneously enabled in the noise critical case(s) to reduce the noise, whereas the voting scheme can be disabled during the remaining cases to save power. To estimate the potential power savings of such a scheme, a 12-bit SAR ADC is considered as an example. To reduce the noise level of the comparator in such an ADC by 2 , permanent analog scaling would need 4 scaling of the comparator, leading to a power-consumption increase of 300%. With a flexible majority-voting system, 5 voting will be sufficient to get a similar noise reduction. If the voting is enabled only in one cycle, the total number of comparisons is instead of 12. Thus, the power increase is only 33% as opposed to the 300% of the conventional analog-scaling method.

The second requisite to implement the proposed method [69] is a technique to estimate the amplitude of Vin , such that the most noise-critical cycle can be detected, which is not known beforehand. To automatically detect this, the comparator decision-time can be observed, as this decision time is inversely related to the amplitude of the input signal [M.Pelgrom 2017]. If the decision-time is longer than a certain threshold-time Tmv , this implies that the input signal is smaller than a certain threshold voltage Vmv. In this way, the method can detect whether a decision is noise-critical or not. The threshold for being noise-critical can be set by tuning the delay Tmv to achieve a desired Vmv.

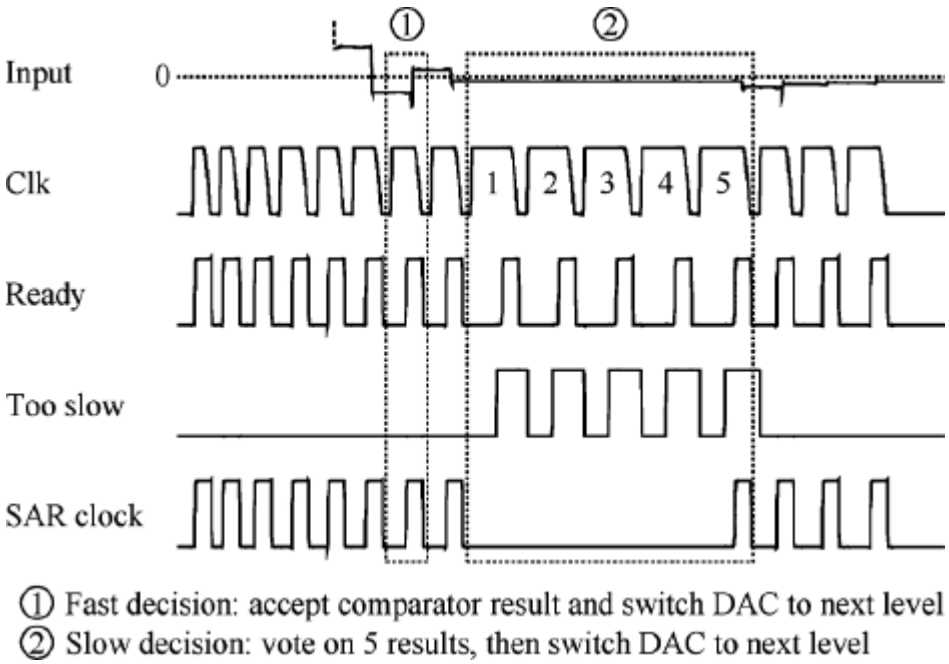


Figure 5-118 Timing diagram of a 12-bit SAR conversion with Data-Driven Noise Reduction (DDNR). [69]

The proposed method in [69] is denoted by the term Data-Driven Noise Reduction (DDNR), as it selectively applies noise reduction by means of majority voting, dependent on the input data (i.e., the amplitude information of Vin). The timing diagram of a 12-bit SAR conversion with DDNR is illustrated in Fig. 5-118: the comparator *Clk* initiates each comparison. Together with the result of the comparison, the comparator also generates a *Ready* indication as soon as the decision is made. As shown by situation 1 in Fig. 5-118, if the *Ready* indication occurs fast enough, the comparator output is immediately accepted without majority voting as the input signal is apparently sufficiently large. In case the circuitry detects that the comparator decision is *too slow* (situation 2 in the figure), the decision may be corrupted by noise. **Thus, the decision is postponed**, 4 additional comparisons are performed, and majority voting is applied to the 5 samples to obtain the majority decision while suppressing the noise. The clock for the SAR logic normally follows the comparator *Ready* signal to control the bit-cycling operations. However, in the case of voting, the SAR clock is suspended until the majority decision is reached and the next bit cycle can be activated.

C. DDNR Behavioral modelling simulations

The author in [69] made behavioral modelling analysis for the DDNR technique, as the threshold level is moved up, more bits are subjected to operating the Majority voting logic, for instance when the threshold was less than LSB, this means that the majority voting would operate in the LSB cycle only, but if the threshold was more than LSB, the DDNR would operate in more bits than just the LSB, of course higher threshold level will indicate more comparisons and thus more power consumption and thus less input referred noise (IRN), refer to [69] for further knowledge.

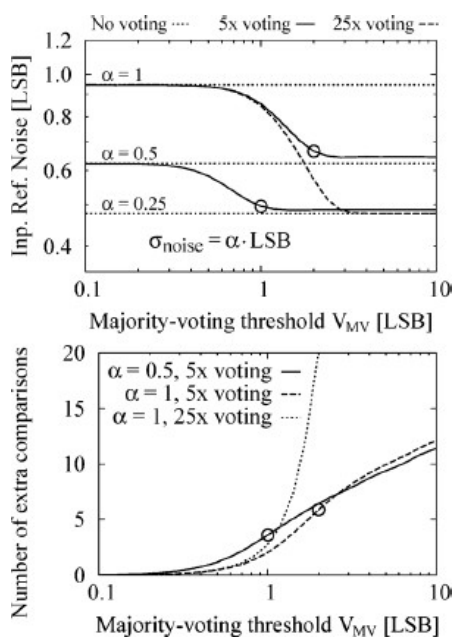


Figure 5-119 Simulated ADC input-referred noise and number of extra comparisons versus majority-voting threshold for various amounts of comparator noise ($\sigma_{noise} = \alpha$.LSB). [69]

As we can see in Fig 5-119, the IRN-plot, the horizontal lines without voting indicate the performance of a conventional ADC without DDNR. When voting is applied, the IRN will become dependent on the selected threshold level, as the threshold decides when to apply voting or not. For low thresholds (i.e., almost no voting will be applied), the DDNR ADC logically performs identical to a conventional ADC with the same comparator noise. When the threshold is increased, the IRN will gradually go down until it reaches the limit as expected based on the number of samples in the voting scheme: with 5 or 25 samples, the noise reduces equivalent to 4x or 16x analog scaling, respectively. This is confirmed from the figure as the DDNR ADC with e.g. α=1, approximates the performance of a conventional ADC with α=0.5 when 5 votes are used and when the threshold setting is sufficiently high. Thus, it indeed performs similar to analog design scaled by 4 . When 25 votes are used, the curve approximates a conventional ADC with α=0.25 , equivalent to 16x analog scaling. The circles in the figure indicate the points from where maximum noise reduction is obtained. The second plot in Fig. 5-119 illustrates the average number of additional comparisons per conversion (on top of the 12 nominal ones), which logically increases as the threshold is moved up.

The circles in this plot (corresponding to the circles in the IRN plot) highlight that for 5x voting and for α=0.5 or 1, about 4 or 6 additional comparisons are sufficient to get the maximum possible IRN-improvement. The number of additional comparisons is increasing with higher comparator noise (i.e., higher α),which can be understood from the fact that more decisions will become noise-critical when there is more comparator noise. Thus, the voting threshold needs to be moved up to allow proper noise suppression in case of increased comparator noise. Nonetheless, the 4 or 6 additional comparisons lead to an increase of about 30% or 50% of the overall power consumption only, which is substantially below the +300% required by conventional analog scaling. For the situation with voting on 25 samples, the power consumption (at maximum attainable IRN) increases by a factor of 5, which is at least 3x more efficient than the equivalent 16x analog scaling.

We proposed hardware implementation in the paper is shown in Figure 5-120, and then we will discuss our own proposal.

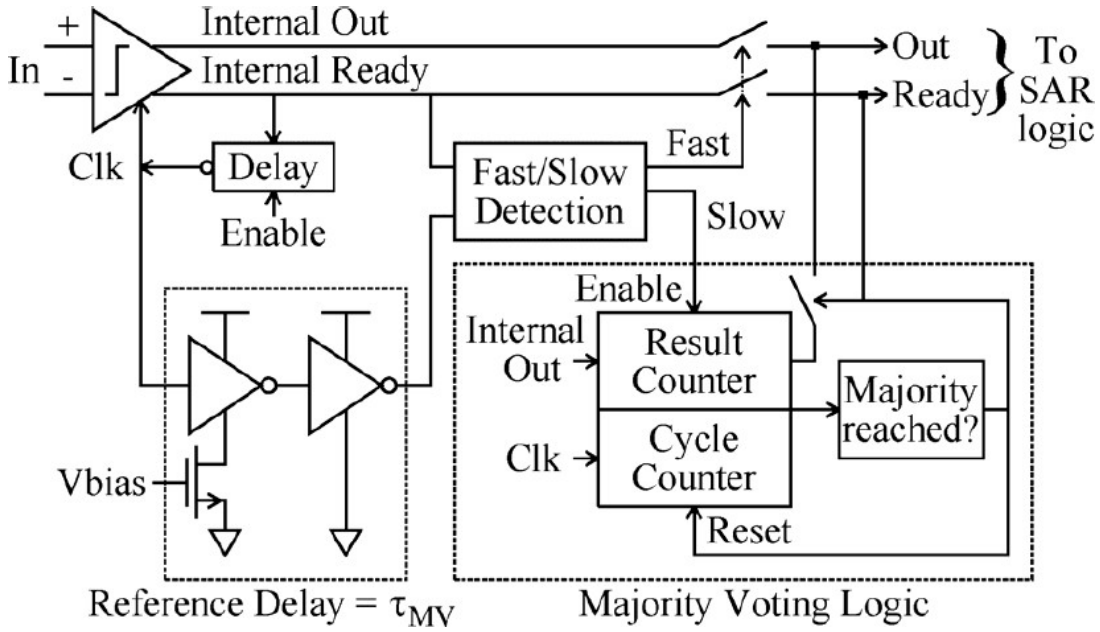


Figure 5-120 Circuit implementation of the Data-Driven Noise-Reduction method (DDNR).[69]

Our proposed hardware implementation :

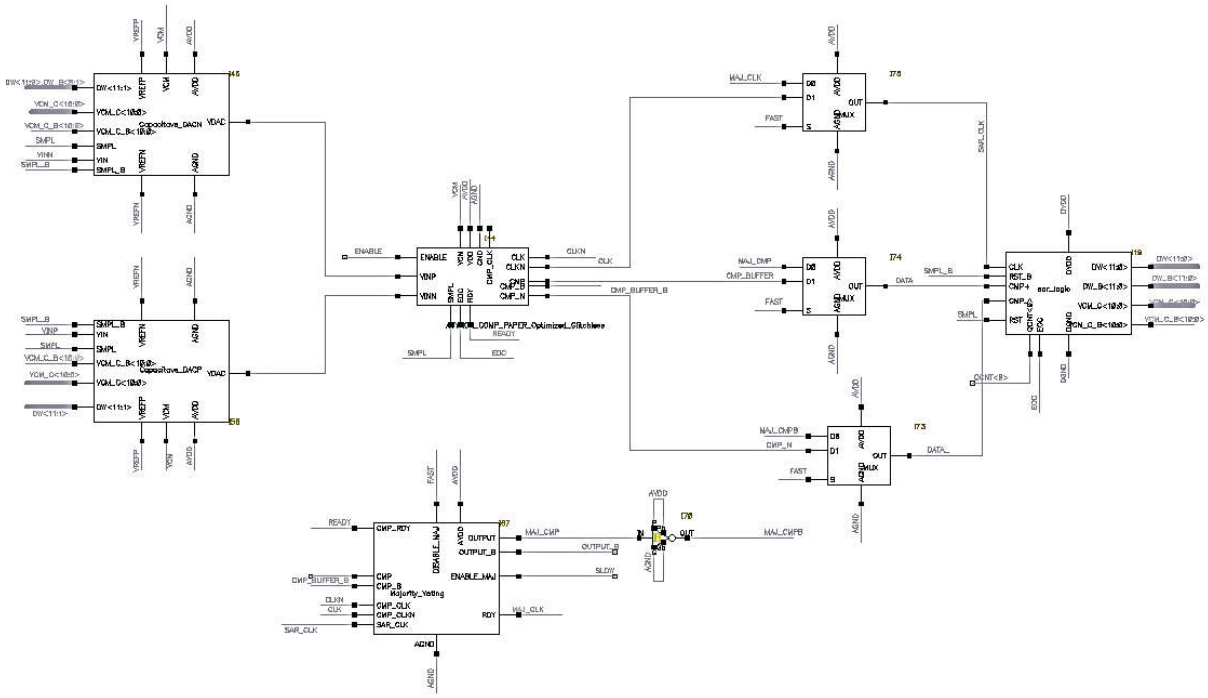


Figure 5-121 Circuit implementation of the Full SAR-ADC-Majority block .

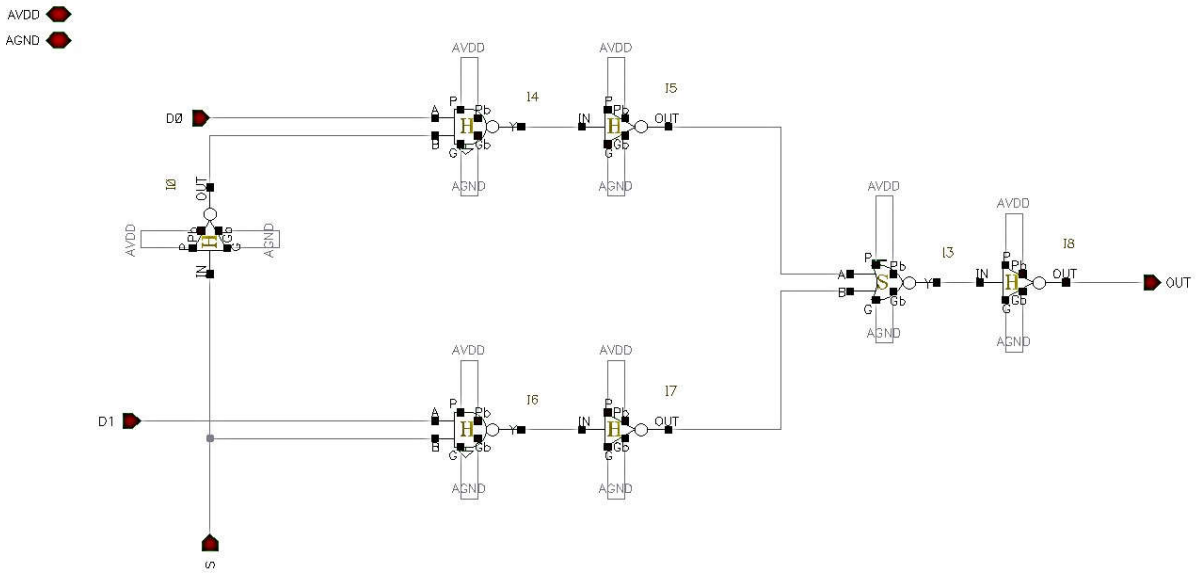


Figure 5-122 Circuit implementation of the Multiplexer block.

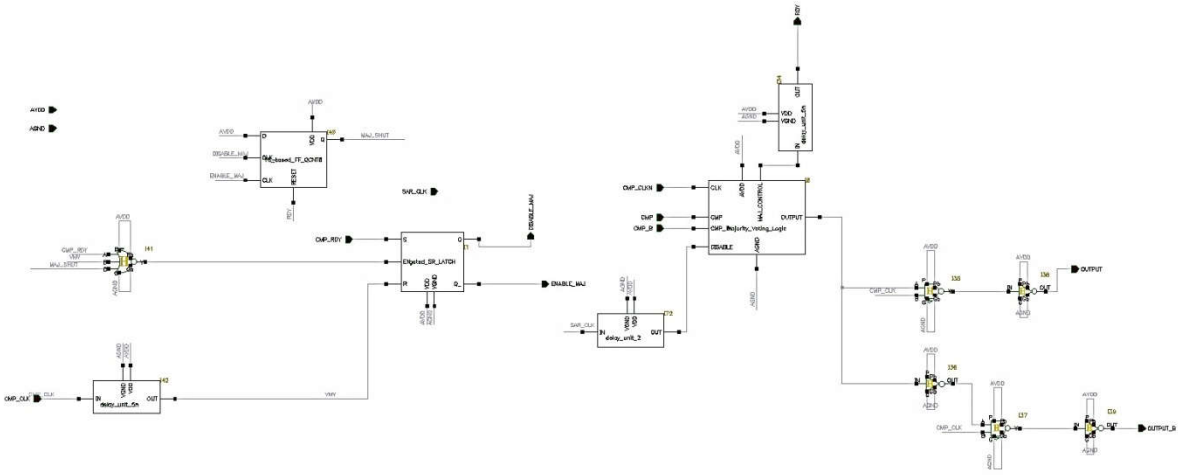


Figure 5-123 Circuit implementation of the Majority voting block.

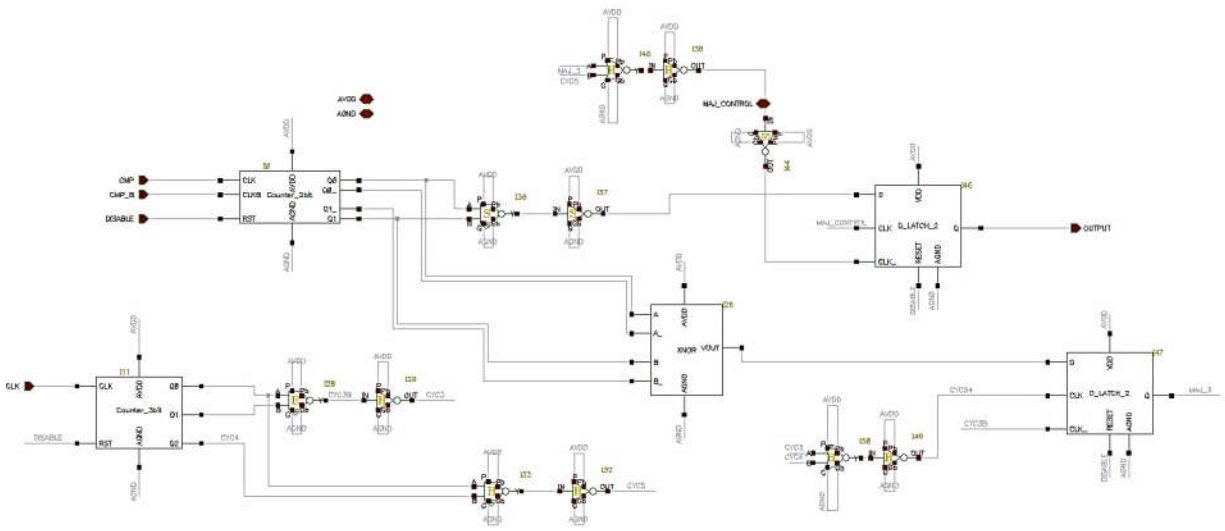


Figure 5-124 Circuit implementation of the Majority voting logic block.

In Figure 5-123, we can see the circuit implementation of the majority voting block, as we know, we will identify the Vin at which we will operate the majority voting based on the delay the comparator takes to resolve the output, so instead of using current starved inverters which required certain stable PVT insensitive voltage input, we used an inverter chain to delay the CMP-CLK to a certain value that corresponds to our required VMV, if the CMP-READY first it will SET the SR-Latch and thus disabling the comparator which means that Vin is above the threshold value, but if CMP-CLK which ,still after the delay, came earlier than CMP-READY this triggers the SR-Latch to enable the majority voting logic block.

When the majority voting is enabled, this postpones the decision as if it isolates the comparator with the majority voting block where it operates either for 3 or 5 times depending on the output decision and the majority of votes, the Multiplexers in Figure 5-121 isolate the comparator outputs which are the CLK, CMP,CMPN, these outputs now go to the majority voting block and during this time, the SAR-LOGIC is at rest waiting for the Majority voting block to give it the CLK, Maj-CMP and Maj-CMPN to store after the comparator has run extra 3 or 5 times to get the final decision. As we discussed before, each comparator run produces an output that is un-correlated to the other runs and other outputs giving a complete statistical independence from the other outputs.

The majority voting block is designed so as to count from 0 to 5 using a 3-bit counter, in the meantime, there is a two bit counter that is counting the number of ones that the comparator gives during these extra cycles, at the third cycle, the logic checks if the comparator gave all three

successive ones, and thus the final output is one, and the comparator does not continue the next two comparisons because the answer has already been set, also at the fourth cycle it checks for three ones and if so, it gives the output as HIGH without running that extra cycle, this saves a lot of power, After that we have a viable output either LOW or HIGH depending on the majority of votes, this is set by the 2-bit counter, and also if the decision came at the third, fourth or finally at the fifth cycle, a trigger is generated which resembles the SAR-CLK that would pass on to the logic to push the newly achieved value either HIGH or LOW to the LOGIC, and then the Majority voting block is disabled, the Multiplexers re-establishes the direct connection between the SAR-LOGIC and the comparator, and at every new cycle the condition is re-checked on whether to operate the majority block or not.

Since this is a 3-or-5 logic block, we decided to do a little extra enhancement, since we already have a 3-bit counter which can count from 0-7, we decided to make it a 4-7 logic block as in Figure 5-125, first value to get 4 votes gets selected or at the end of the cycle number 7, we check the votes, in this variation we see that at cycles 4,5,6,7 we check if the third bit is 1, this means that we have 4 ones, so once we achieve that we report it and disable the majority, on the other hand if we achieve 4 zeros at cycle 4,5,6 we report it as well, the output is LOW and the majority is disabled once the output is achieved, a SAR-CLK edge is generated to write the output value down in the SAR-LOGIC. In both topologies the counters are reset right after the value is achieved.

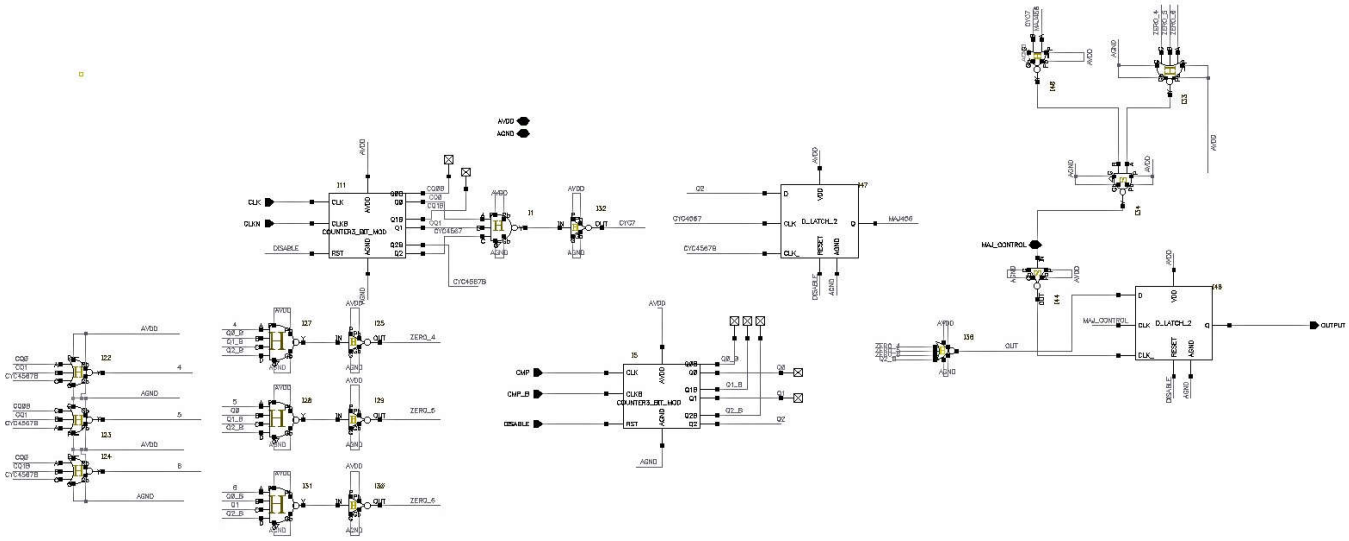


Figure 5-125 Circuit implementation of the 4-7 Majority voting logic block.

In figure 5-126, we will report the Waveforms showing the operation, and the simulation results will be layout in the final chapter i.e. simulation results.

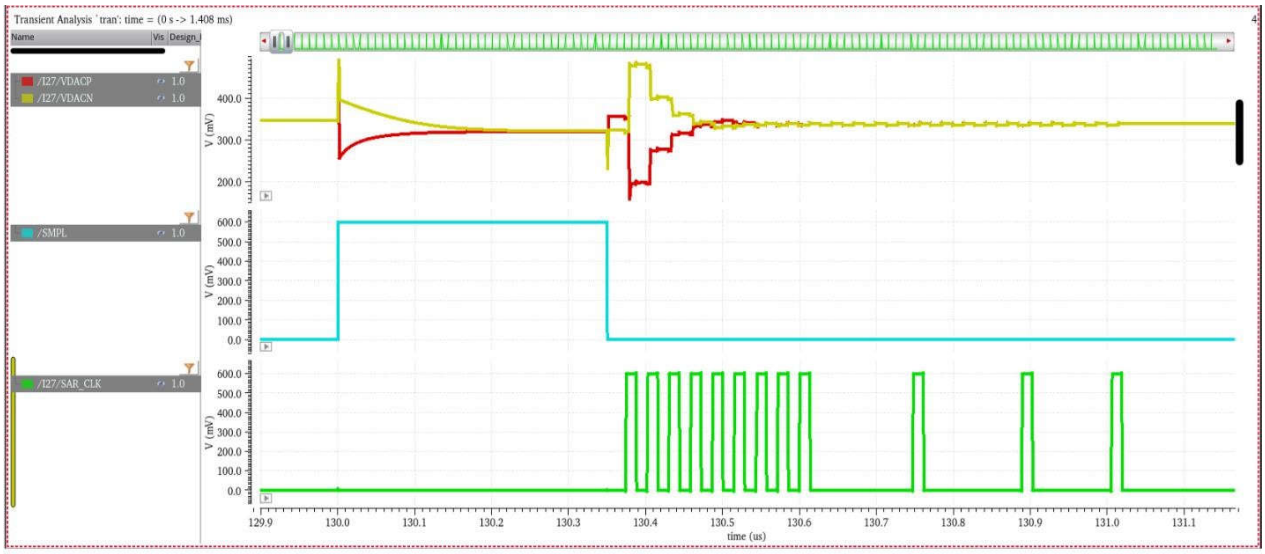


Figure 5-126 Waveforms showing how the SAR-CLK was postponed for some time during the operation of the majority voting.

You should note that the Majority voting block can be operated once per sample, if were able to set the delay time to a point where it points out the required Vin, but here we reduced the delay to a point where the last 3 LSBs can use the majority voting block to their advantage and reduce the noise spec. on these last LSBs.

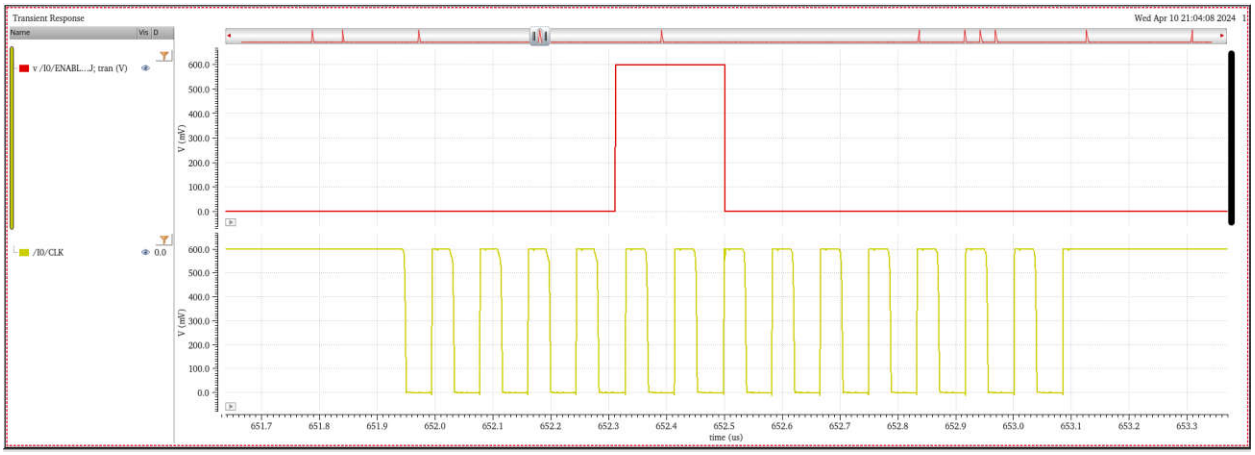


Figure 5-127 Waveforms showing how the CMP-CLK operates only three times as the comparator has resolved the required outputs in the first three cycles.

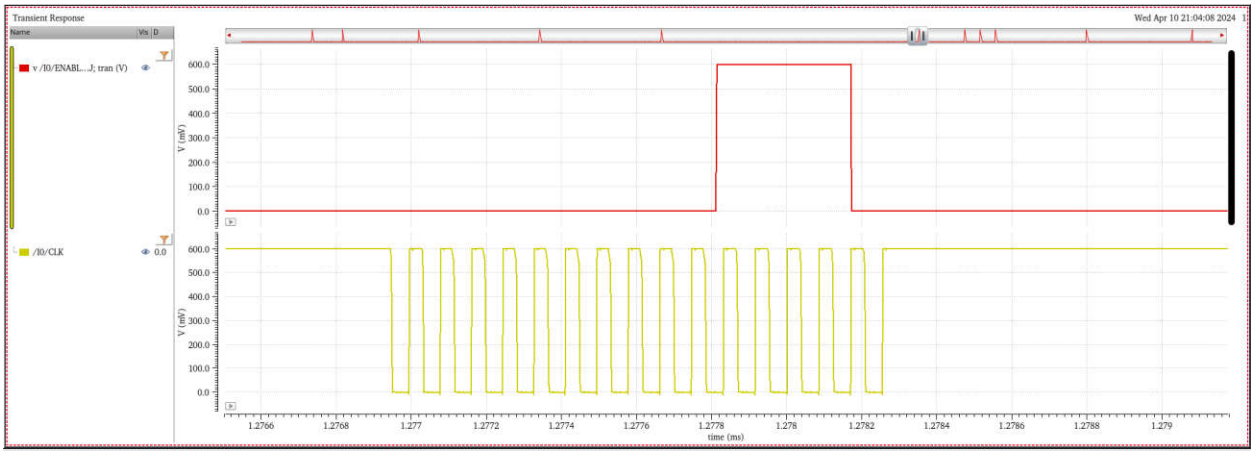


Figure 5-128 Waveforms showing how the CMP-CLK operates five times as the comparator has resolved the required outputs during the complete five cycles.

In Figure 5-128 and Figure 5-127, we show the Enabling of the majority voting block once during each sample to perform its function of reducing the IRN.

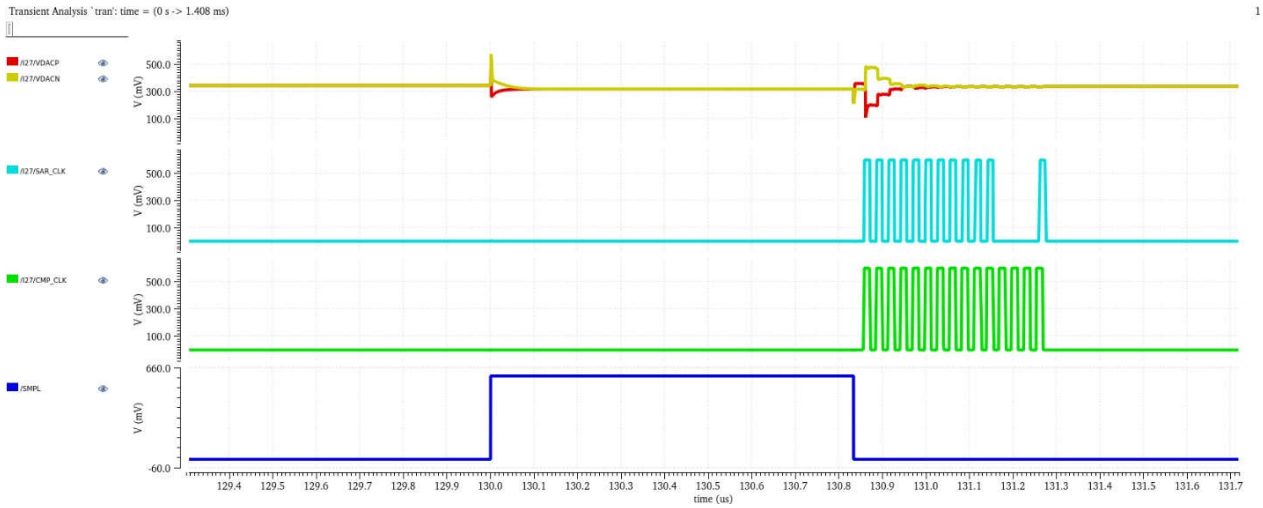


Figure 5-129 Waveforms showing how the CMP-CLK operates extra 3 times but it is isolated from the SAR-CLK, where the SAR-CLK gives the 12 cycles that are desired.

5.18 Conclusion

In conclusion, in this chapter we have studied the full cycle of the comparator, starting with the concepts, passing by the design, mathematical equations describing the block, and then the circuit implementations, we showed that the design flow should start with behavioral modelling and then we go for the schematics for the single block, and then do the integration process with the entire system, and re-iterate the design, then we go for the Layout, and during the Layout there are multiple iterations in the design, but finally we got there and completed the design of the full comparator with its asynchronous loop and most of the required tests regarding the comparator block.

References

[1] Low-Power SAR ADC Design: Overview and Survey of State-of-the-Art Techniques

[2] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. A. M. Klumperink, and B. Nauta, "A 10-bit charge-redistribution ADC consuming 1.9 μW at 1 MS/s," IEEE J. Solid-State Circuits, vol. 45, no. 5, pp. 1007–1015, May 2010.

[3] D. Schinkel, E. Mensink, E. Klumperink, E. Van Tuijl, and B. Nauta, "A double-tail latch-type voltage sense amplifier with 18 ps setup+ hold time," in IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2007, pp. 314–605.

[4] H. S. Bindra, C. E. Lokin, D. Schinkel, A.-J. Annema, and B. Nauta, "A 1.2-V dynamic bias latch-type comparator in 65-nm CMOS with 0.4-mV input noise," IEEE J. Solid-State Circuits, vol. 53, no. 7, pp. 1902–1912, Jul. 2018.

[5] M. Liu, K. Pelzers, R. van Dommele, A. van Roermund, and P. Harpe, "A 106 nW 10 b 80 kS/s SAR ADC with duty-cycled reference generation in 65 nm CMOS," IEEE J. Solid-State Circuits, vol. 51, no. 10, pp. 2435–2445, Oct. 2015.

[6] X. Tang, B. Kasap, L. Shen, X. Yang, W. Shi, and N. Sun, "An energyefficient comparator with dynamic floating inverter pre-amplifier," in Proc. IEEE Symp. VLSI Circuits (VLSI), Jun. 2019, pp. C140–C141.

[7] P. Nuzzo, F. De Bernardinis, P. Terreni, and G. Van der Plas, "Noise analysis of regenerative comparators for reconfigurable ADC architectures," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 6, pp. 1441–1454, Jul. 2008.

[8] L. Chen, A. Sanyal, J. Ma, X. Tang, and N. Sun, "Comparator commonmode variation effects analysis and its application in SAR ADCs," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 2014–2017.

[9] B. Razavi, “The StrongARM latch [A circuit for all seasons],” IEEE Solid-State Circuits Mag., vol. 7, no. 2, pp. 12–17, Spring 2015.

[10] T. Sepke, P. Holloway, C. G. Sodini, and H.-S. Lee, “Noise analysis for comparator-based circuits,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 3, pp. 541–553, Mar. 2009.

[11] B. Øksendal, Stochastic Differential Equations. Berlin, Germany: Springer-Verlag, 1998

[12] B. Razavi, “The StrongARM latch [A circuit for all seasons],” IEEE Solid-State Circuits Mag., vol. 7, no. 2, pp. 12–17, Spring 2015.

[13] P. Nuzzo, F. De Bernardinis, P. Terreni, and G. Van der Plas, “Noise analysis of regenerative comparators for reconfigurable ADC architectures,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 6, pp. 1441–1454, Jul. 2008.

[14] T. Sepke, P. Holloway, C. G. Sodini, and H.-S. Lee, “Noise analysis for comparator-based circuits,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 3, pp. 541–553, Mar. 2009.

[15] D. Schinkel, E. Mensink, E. Klumperink, E. van Tuijl, and B. Nauta, “A double-tail latch-type voltage sense amplifier with 18ps setup+hold time,” in IEEE ISSCC Dig. Tech. Papers, Feb. 2007, pp. 314–605.

[16] K. R. Laker and W. M. C. Sansen, Design of Analog Integrated Circuits and Systems, McGraw-Hill, 1994.

[17] J.-T. Wu and K.-L. Chang, “MOS Charge Pumps for Low-Voltage Operation,” IEEE Journal of Solid-State Circuits, vol. 33, pp. 592–597, April 1998.

[18] K. Bult, “Analog Design in Deep Sub-Micron CMOS,” Proc. of the IEEE European Solid-State Circuits Conf., pp. 11–17, 2000.

[19] J. Cherry and W. Snelgrove, “Clock Jitter and Quantizer Metastability in Continuous-Time Delta–Sigma Modulators,” IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing, vol. 46, pp. 661–676, June 1999.

[20] S. Luschas and H. S. Lee, “High-Speed __ Modulators with Reduced Timing Jitter Sensitivity,” IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing, vol. 49,pp. 712–720, November 2002.

[21] O. Oliaei, “Design of Continuous-Time Sigma-Delta Modulators with Arbitrary Feedback Waveform,” IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing, vol. 50, pp. 437–444, August 2003.

[22] O. Oliaei, “Sigma-Delta Modulator with Spectrally Shaped Feedback,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 50, pp. 518–530, September 2003.

[23] L. Hern´andez, A. Wiesbauer, S. Pat´on, and A. DiGiandomenico, “Modelling and Optimization of Low Pass Continuous-Time Sigma-Delta Modulators for Clock Jitter Noise Reduction,” Proc. of the IEEE Intl. Symp. on Circuits and Systems, pp. 1072–1075, 2004.

[24] L. Hern´andez, P. Rombouts, E. Prefasi, S. Pat´on, and C. L. M. Garc´ıa, “A Jitter Insensitive Continuous- Time __ Modulator Using Transmission Lines,” Proc. of the IEEE Intl. Conf. on Electronics, Circuits and Systems, pp. 109–112, 2004.

[25] R. H. van Veldhoven and A. H. M. van Roermund, Robust Sigma Delta Converters, Springer, 2011.

[26] P. G. R. Silva and J. H. Huijsing, High Resolution IF-to-Baseband __ ADC for Car Radios, Springer, 2008.

[27] M. Miyahara, Y. Asada, D. Paik, and A. Matsuzawa, “A low-noise self-calibrating dynamic comparator for high-speed ADCs,” in Proc. IEEE Asian Solid-State Circuits Conf., Nov. 2008, pp. 269–272.

[28] M. van Elzakker, E. van Tuijl, P. Geraedts, D. Schinkel, E. A. M. Klumperink, and B. Nauta, "A 10-bit charge-redistribution ADC consuming 1.9 μW at 1 MS/s," IEEE J. Solid-State Circuits, vol. 45, no. 5, pp. 1007–1015, May 2010.

[29] S.-E. Hsieh and C.-C. Hsieh, "A 0.44-fJ/conversion-step 11-bit 600- kS/s SAR ADC with semi-resting DAC," IEEE J. Solid-State Circuits, vol. 53, no. 9, pp. 2595–2603, Sep. 2018.

[30] M. Liu, K. Pelzers, R. van Dommele, A. van Roermund, and P. Harpe, "A 106nW 10-b 80 kS/s SAR ADC with duty-cycled reference generation in 65 nm CMOS," IEEE J. Solid-State Circuits, vol. 51, no. 10, pp. 2435–2445, Oct. 2015.

[31] X. Tang, L. Chen, J. Song, and N. Sun, "A 1.5fJ/conv-step 10b 100kS/s SAR ADC with gain-boosted dynamic comparator," in Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC), Nov. 2017, pp. 229–232.

[32] F. van der Goes et al., "11.4 A 1.5 mW 68dB SNDR 80MS/s 2× interleaved SAR assisted pipelined ADC in 28nm CMOS," in IEEE ISSCC Dig. Tech. Papers, Feb. 2014, pp. 200–201.

[33] H. S. Bindra, C. E. Lokin, D. Schinkel, A.-J. Annema, and B. Nauta, "A 1.2-V dynamic bias latch-type comparator in 65-nm CMOS with 0.4-mV input noise," IEEE J. Solid-State Circuits, vol. 53, no. 7, pp. 1902–1912, Jul. 2018.

[34] X. Tang et al., "An energy-efficient comparator with dynamic floating inverter amplifier," IEEE J. Solid-State Circuits, vol. 55, no. 4, pp. 1011–1022, Apr. 2020.

[35] P. G. R. Silva and J. H. Huijsing, High Resolution IF-to-Baseband __ ADC for Car Radios, Springer, 2008.

[36] A. Agnes, E. Bonizzoni, P. Malcovati, and F. Maloberti, “A 9.4-ENOB 1V 3.8 µW100kS/s SAR ADC with time-domain comparator,” in IEEE ISSCC Dig. Tech. Papers. IEEE, 2008, pp. 246–610.

[37] S.-K. Lee, S.-J. Park, H.-J. Park, and J.-Y. Sim, “A 21 fJ/conversionstep 100 kS/s 10-bit ADC with a low-noise time-domain comparator for low-power sensor interface,” IEEE J. Solid-State Circuits, vol. 46, no. 3, pp. 651–659, Mar. 2011.

[38] J. P. Mathew, L. Kong, and B. Razavi, “A 12-bit 200-MS/s 3.4-mW CMOS ADC with 0.85-V supply,” in Proc. Symp. VLSI Circuits (VLSI Circuits), Jun. 2015, pp. C66 C67.

[39] J. Jin, Y. Gao, and E. S.- Sinencio, “An energy-efficient time-domain asynchronous 2 b/step SAR ADC with a hybrid R-2R/C-3C DAC structure,” IEEE J. Solid-State Circuits, vol. 49, no. 6, pp. 1383–1396, Jun. 2014.

[40] K. Yoshioka and H. Ishikuro, “A 13b SAR ADC with eye-opening VCO based comparator,” in Proc. 40th Eur. Solid State Circuits Conf. (ESSCIRC), Sep. 2014, pp. 411–414.

[41] VAN ELZAKKER et al.: A 10-bit CHARGE-REDISTRIBUTION ADC CONSUMING 1.9 W AT 1 MS/s

[42][TANG et al.: ENERGY-EFFICIENT COMPARATOR WITH DYNAMIC FIA]

[43] Methods for Fast Characterization of Noise and Offset in Dynamic Comparators

[44] Spectre ® Circuit Simulator Reference, Product Version 19.1, Cadence Design Systems, 2020.

[45] T. Rabuske and J. Fernandes, “Noise-aware simulation-based sizing and optimization of clocked comparators,” Analog Integrated Circuits and Signal Processing, vol. 81, no. 3, pp. 723–728, 2014, issn:15731979. doi: 10.1007/s10470-014-0428-4.

[46] J. Kim, B. S. Leibowitz, J. Ren, and C. J. Madden, "Simulation and analysis of random decision errors in clocked comparators," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 8, pp. 1844–1857, 2009. doi: 10.1109/TCSI.2009.2028449.

[47] behad.razavi The Design of a Comparator, tutorials of the analog mind

[48] B. Razavi, "The StrongARM Latch [a circuit for all seasons]," IEEE Solid State Circuits Mag., vol. 7, no. 2, pp. 12–17, Spring 2015. doi: 10.1109/MSSC.2015.2418155.

[49] Fast and accurate technique for comparator offset voltage simulation

[50] F. Op't Eynde *et al.*, "A CMOS Fourth-Order 14 b 500 k-Sample/s Sigma-Delta ADC Converter," *IEEE ISSCC Digest of Technical Papers*, pp. 62–63, February 1991.

[51] H.-B. Le *et al.*, "A Regulator-Free 84 dB DR Audio-Band ADC for Compact Digital Microphones," *Proc. of the IEEE Asian Solid-State Circuits Conf.*, 2010.

[52] B. E. Boser and B. A. Wooley, "The Design of Sigma-Delta Modulation Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, vol. **23**, pp. 1298–1308, December 1988.

[53] S. Nadeem *et al.*, "16-Channel Oversampled Analog-to-Digital Converter," *IEEE Journal of Solid-State Circuits*, vol. **29**, pp. 1077–1085, September 1994.

[54] T. V. Burmas *et al.*, "A Second-Order Double-Sampled Delta-Sigma Modulator Using Additive-Error Switching," *IEEE Journal of Solid-State Circuits*, vol. **31**, pp. 284–293, March 1996.

[55] A. Pena-Perez *et al.*, "A 84 dB SNDR 100 kHz Bandwidth Low-Power Single Op-Amp Third-Order. Modulator Consuming 140µW," *IEEE ISSCC Digest of Technical Papers*, pp. 478–479, February 2011.

[56] Y.L. Wong, M.H. Cohen, P.A. Abshire, A floating-gate comparator with automatic offset adaptation for 10-bit data conversion, IEEE Trans. Circuit. Syst. I: Regular Paper. 52 (7) (July 2005) 1316–1325.

[57] H. Omran, A. Alhoshany, H. Alahmadi, K.N. Salama, A 33fJ/step SAR capacitance-to-digital converter using a chain of inverter-based amplifiers, IEEE Trans. Circuit. Syst. I: Regular Paper. 64 (2) (Feb 2017) 310–321.

[58] A. Graupner, A Methodology for the Offset-Simulation of Comparators, The Designers Guide Community, 2005.

[59] Cadence Design Systems, ADC Verification Workshop, 2014.

[60] M.J. Pelgrom, Analog-to-Digital Conversion, Springer, 2017.

[61] H. Park et al., "A 0.7-V 870-μW Digital-Audio CMOS Sigma-Delta Modulator," IEEE Journal of Solid-State Circuits, vol. 44, pp. 1078–1088, April 2009.

[62] K. Lee et al., "A Noise-Coupled Time-Interleaved __ ADC with 4.2MHz BW, -98 dB THD, and 79 dB SNDR," IEEE ISSCC Digest of Technical Papers, pp. 494–495, February 2008.

[63] A.J. Gines, E. Peralias, G. Leger, A. Rueda, Closed-loop simulation method for evaluation of static offset in discrete-time comparators, in: Electronics, Circuits and Systems (ICECS), 2014 21st IEEE International Conference on. IEEE, 2014, pp. 538–541.

[64] Digital Background Self-Calibration Technique for Compensating Transition Offsets in Reference-less Flash ADCs

[65] R. Wu et al., Precision Instrumentation Amplifiers and Read-Out Integrated Circuits,

[66] [J. Xue et al. / INTEGRATION, the VLSI journal 43 (2010) 353–364]

[67] P. Harpe, E. Cantatore, and A. van Roermund, "11.1 An oversampled 12/14b SAR ADC with noise reduction and linearity enhancements achieving up to 79.1dB SNDR," in IEEE ISSCC Dig. Tech. Papers, Feb. 2014, pp. 194–195.

[68] A. Shrivastava, "12-bit non-calibrating noise-immune redundant SAR ADC for system-on-a-chip," in Proc. IEEE Int. Symp. Circuits Syst., May 2006, p. 4.

[69] HARPE et al.: A 10b/12b 40 kS/s SAR ADC WITH DATA-DRIVEN NOISE REDUCTION ACHIEVING UP TO 10.1b ENOB AT 2.2 fJ/CONVERSION-STEP Analog Circuits and Signal Processing, DOI: 10.1007/978-1-4614-3731-4_2, Springer Science+Business Media New York 2013

.