End Term Report

ISRO VLSI Mid Prep
Team 99

INTER IIT TECH MEET 14.0 | IIT PATNA

Contents

1 Introduction

1.1 Fundamentals of Delta–Sigma ADC Architecture

A Delta-Sigma ($\Delta\Sigma$) ADC is one of the popular topologies in designing of high resolution ADCs. It comprises of two stages- Delta Sigma Modulator and Digital Filter. The first stage as name suggests, modulates and reshapes the incoming raw analog signal into desired oversampled bitstream. The second stage takes the bitstream then downsamples them and packages them into desired digital output format [1]. They are widely used in precision, low-frequency applications such as sensors, control systems, and scientific instruments. Because they offer excellent noise performance and wide dynamic range, they are well-suited for environments where small signal changes must be measured very accurately, such as space-grade payload electronics and navigation systems.

1.2 Project Objective

The objective of this project is to study, model, simulate, and design a high-resolution Delta–Sigma ($\Delta\Sigma$) ADC suitable for future ISRO payload applications. The focus is on the Delta–Sigma modulator and the digital/decimation filter, with the goal of achieving higher ENOB at higher sampling rates.

(a) Study of Architectures

- Analyze Delta–Sigma modulator types such as Discrete-Time (DT), Continuous-Time (CT), and Hybrid.
- Review digital/decimation filters including SINC, FIR, and IIR.

(b) MATLAB/Simulink Modeling

- Develop behavioural models to study modulator order, oversampling ratio, and filter performance.
- Perform trade-off analysis between Nyquist sampling frequency (0.5 ksps–2 ksps) and target ENOB (16–19 bits).

(c) Noise and Non-Idealities Analysis

- Evaluate the impact of flicker noise on achievable resolution.
- Assess the effectiveness of chopping, nested chopping, and auto-zeroing techniques for noise reduction.

(d) Digital Filter RTL Design

- Select suitable coefficient precision and filter order for hardware implementation.
- Implement the decimation filter at the RTL level using Verilog/VHDL.
- Verify functionality using simulation platforms such as VCS or QuestaSim.

(e) ASIC Flow

- Perform synthesis using Synopsys Design Compiler or Cadence Genus.
- Execute place-and-route using ICC/ICC2 or Cadence Innovus.
- Conduct post-layout simulations to validate timing and functionality.

This project supports the development of indigenous, high-resolution ADC solutions for ISRO payloads. It contributes to precise sensing applications and aligns with national goals of self-reliance in VLSI and ASIC technologies.

2 Architecture & Methodology

2.1 Architecture

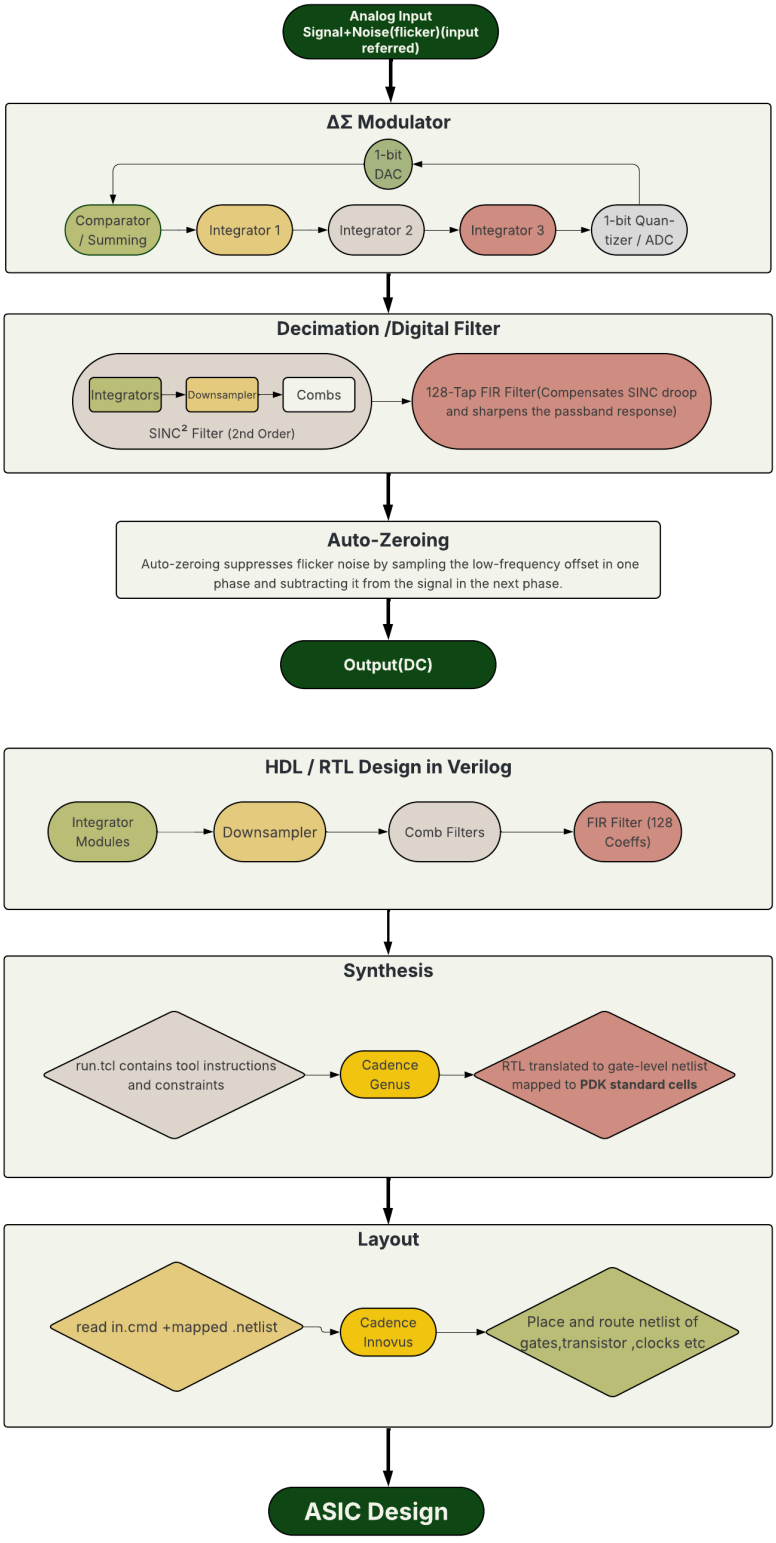


Figure 1: System Architecture and Design Workflow

2.2 SINC^2 (CIC-2) Decimation Filter Methodology

A SINC^2 decimator consists of two integrators, downsampling by OSR $= r$, and two comb stages.

2.2.1 Integrator Section

$$z[n] = x[n] + z[n-1], \quad y[n] = y[n-1] + z[n].$$

Expanding:

$$z[n] = \sum_{k=0}^{n} x[k], \qquad y[n] = \sum_{k=0}^{n} \sum_{i=0}^{k} x[i].$$

Rewriting:

$$y[n] = \sum_{i=0}^{n} (n - i + 1)x[i].$$

For 1-bit $x[n]$, maximum integrator value:

$$y_{\max} = 1 + 2 + \cdots + r = \frac{r(r+1)}{2}.$$

Total possible levels:

$$L = \frac{r(r+1)}{2} + 1.$$

2.2.2 Downsampling and Comb Logic

$$y_d[m] = y[mr],$$

$$a[m] = y_d[m] - y_d[m-1], \qquad b[m] = y_d[m] - 2y_d[m-1] + y_d[m-2].$$

2.2.3 Maximum Value of $b[m]$

$$b_{\max} = (1 + \cdots + r) + (1 + \cdots + (r-1)) = r^2.$$

Output range:

$$0 \text{ to } r^2, \quad N_{\text{bits}} = \log_2(r^2 + 1).$$

2.2.4 OSR Required for 20-bit ADC

$$\log_2(r^2 + 1) = 20 \Rightarrow r^2 + 1 = 2^{20} \Rightarrow r \approx 1024.$$

2.2.5 Conclusion

A 20-bit SINC^2 decimator requires:

$$\text{OSR} = 1024$$

After the SINC^2 decimator, we have employed a 128-coefficient F.I.R. filter to provide a low pass response and eliminate the noise shifted to high frequency by the DSM ADC.

3 Modulator Study

3.1 Types of Delta–Sigma Modulators

Discrete-Time (DT) Delta–Sigma Modulator

The discrete time (DT) DSM employs the use of switched capacitors to sample the incoming analog signal at input, and thus subsequent design can be done in z domain. Sampling the signal at input allows for greater resistance to internal clock jittering but requires an external anti-aliasing filter. Solving the difference equations (taking in linear approximation of Quantizer as sum of output and quantization noise), we obtain following transfer functions[1]

Transfer Functions:

$$\text{STF}(z) = 1, \qquad \text{NTF}(z) = 1 - z^{-1}$$

Continuous-Time (CT) Delta–Sigma Modulator

The continuous time in contrast maintains the input in continuous time with sampling occurring only at quantizer level, so the control loop design takes place in s domain. This design choice also ensures built-in anti-alias filter but increased sensitivity to clock jitter. Solving differential equations in Laplace domain, one obtains following transfer functions[1]

Transfer Functions:

$$\text{STF}(s) = 1, \qquad \text{NTF}(s) = s$$

Hybrid Delta–Sigma Modulator

As name suggests, the Hybrid topology offers the best of both worlds of CT and DT topologies, especially by employing the built-in features of anti-aliasing (CT), and deterministic behaviour of DT feedback. Usually, this topology is realized with analog front (input, integrator) being CT, and quantization and feedback DAC operations occurring in discrete time.

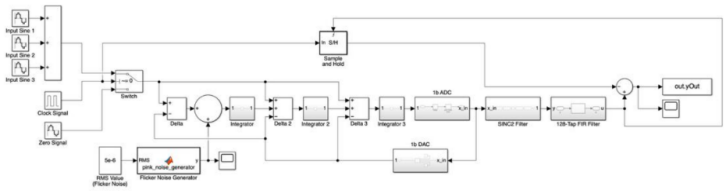
3.2 Chosen Architecture

The Hybrid $\Delta\Sigma$ Modulator is chosen as the final architecture for this project. To justify this selection, the three architectures—DT, CT, and Hybrid—are compared with illustrative images.

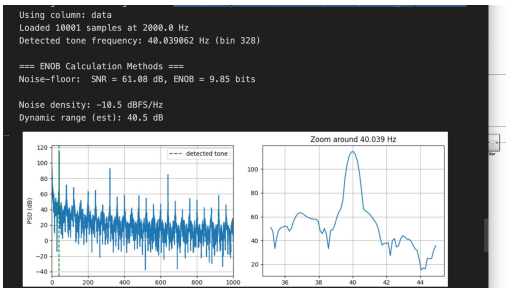
1. Discrete-Time (DT) $\Delta\Sigma$ Modulator — Not Selected

The DT modulator produced lower ENOB for Nyquist sampling frequencies between 0.5–2 ksps. Its main limitations are:

- Sampling errors such as charge injection, clock feedthrough, and finite settling.
- Integrator linearity dependent on capacitor matching and finite op-amp gain.
- SNR and stability degradation at higher OSRs.
- Requires an external analog anti-alias filter.



(a) Discrete Simulink Model



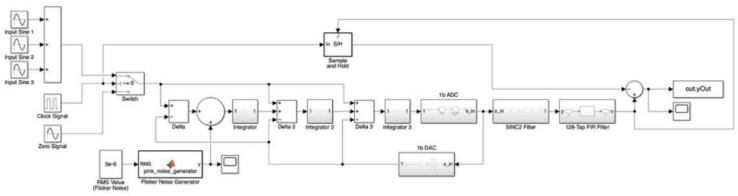
(b) PSD and ENOB Output

Figure 2: Discrete Model

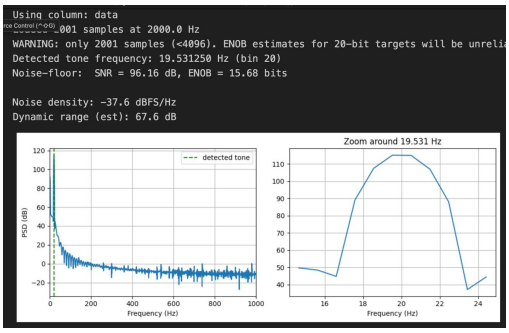
2. Continuous-Time (CT) $\Delta\Sigma$ Modulator — Partially Suitable

The CT modulator performs well ideally and provides inherent anti-alias filtering. However, practical simulation revealed several issues:

- Extremely sensitive to clock jitter, which directly injects noise.
- Harder to stabilize due to RC/Gm-C variation and pole/zero uncertainty.
- Much slower simulation due to continuous-time solver demands.
- Practical non-idealities degrade ENOB rapidly.



(a) Continuous Simulink Model



(b) PSD and ENOB Output

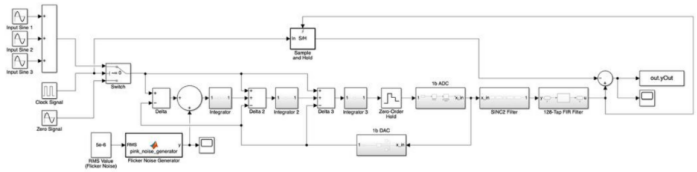
Figure 3: Continous Model

3. Hybrid $\Delta\Sigma$ Modulator — Selected Architecture

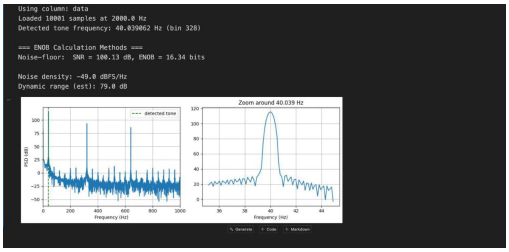
The Hybrid architecture combines:

- **CT loop filter:** providing anti-aliasing, smooth dynamics, low power.
- **DT feedback (e.g., ZOH):** ensuring precise timing and reduced jitter sensitivity.

  Why Hybrid performs best:
- Highest and most stable ENOB across all test cases.
- Much less sensitive to jitter than pure CT.
- Avoids DT issues such as finite settling and capacitor mismatch.
- Faster and more stable simulations than CT.
- Best suited for mixed-signal ASIC implementation.



(a) Hybrid Simulink Model



(b) PSD and ENOB Output

Figure 4: Hybrid Model

Final Conclusion: The Hybrid $\Delta\Sigma$ Modulator provides the best ENOB, jitter robustness, tolerance to non-idealities, and suitability for mixed-signal hardware.

3.3 ENOB vs Sampling Rate

For the selected architecture consisting of two integrators, two comb stages, and a 128-tap FIR filter with coefficients of 1/128, the ENOB initially increases with an increase in Nyquist rate. This improvement occurs because higher Nyquist rates effectively reduce in-band quantization noise through oversampling and noise shaping, which enhances the signal quality. However, after a certain Nyquist rate, the ENOB starts to saturate, as the system becomes limited by non-idealities such as fixed-point errors, residual noise, and finite filter resolution. Beyond this point, further increase in Nyquist rate does not significantly improve the effective resolution, as it reaches asymptote.

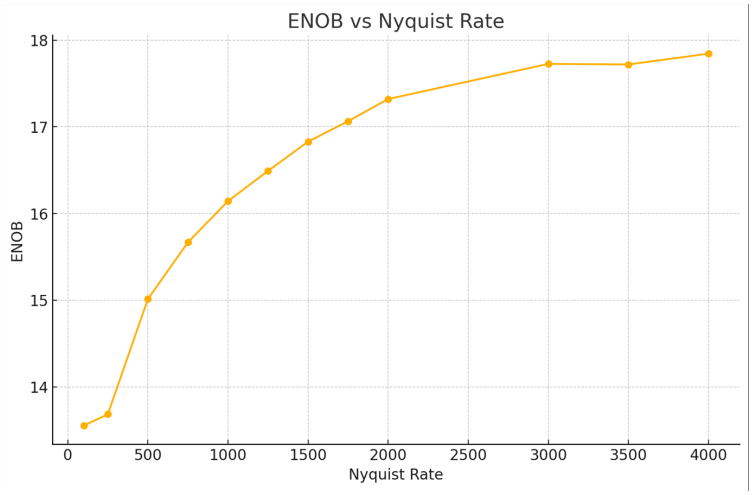


Figure 5: ENOB vs Nyquist Rate Plot at Input Sine (1.2V, 1Hz)
Flicker Noise RMS: 5 μV
Run Time: 1 s

4 Noise Analysis

Identifying, modeling, and mitigating the noise remains one of the prime focus in our design process. Their effect is usually seen as raising the noise floor in the final PSD, which destroys the usable ENOB resolution.

In usual ADC operation the type of noise that is dealt usually includes Quantization Noise (caused by round off error by quantizer), White Noise (thermal noise) and pink noise (flicker noise). Here we majorly focus our analysis on modelling and mitigation of **pink noise** or also known as **flicker noise**. [2]

4.1 Modelling Flicker Noise

Flicker Noise ($1/f$) — Modelling and Simulation

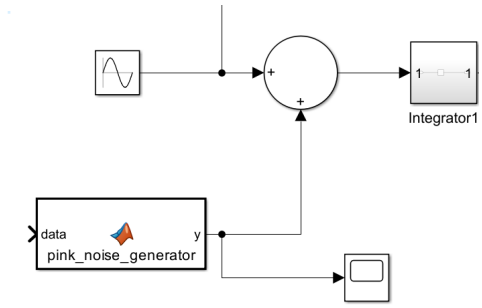
Flicker noise ("$1/f$", pink noise) appears as a low-frequency random noise near DC in MOS devices and is commonly caused by charge trapping/detrapping at oxide interfaces. Its power spectral density (PSD) scales approximately as $S_{fn}(f) \propto 1/f^{\alpha}$ with $\alpha \approx 1$ for many CMOS processes [3]. In precision low-frequency ADCs, flicker noise elevates the noise floor near DC and degrades SNR/ENOB.

4.1.1 Model objective.

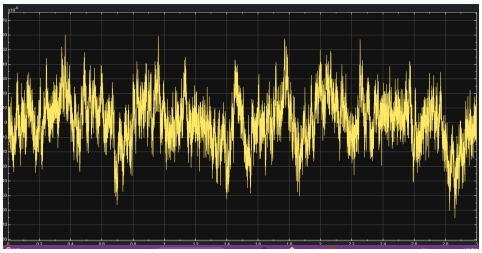
For MATLAB/Simulink-based behavioural modelling we represent flicker noise as an *input-referred additive noise* series $n_f(t)$ that is superposed on the analog input $x(t)$. The goal is a time-domain noise realization whose PSD approximates K/f over the signal band of interest (here 0.5–2 kHz Nyquist targets). [4]

4.1.2 Why a time-domain model?

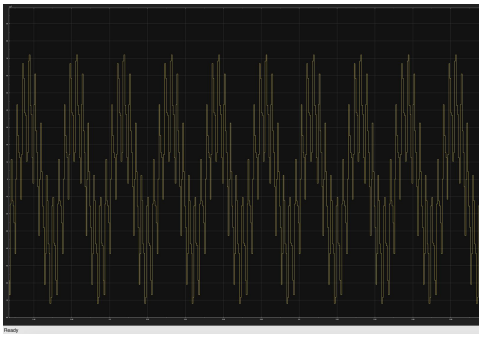
Simulink operates natively in the time domain and does not include a direct $1/f$ (pink-noise) generator suitable for continuous-time modelling. Therefore, we implemented flicker noise using a physically motivated time-domain approach based on the superposition of several low-pass exponential processes (trap models). Each process represents a pseudo-Random Telegraph Signal (RTS) with a distinct time constant, and the time constants are chosen to be log-spaced so that their combined power spectral density approximates the desired $1/f$ behaviour [5]. This method is computationally efficient, compatible with Simulink's continuous-time execution, and accurately reproduces the low-frequency rise characteristic of flicker noise.



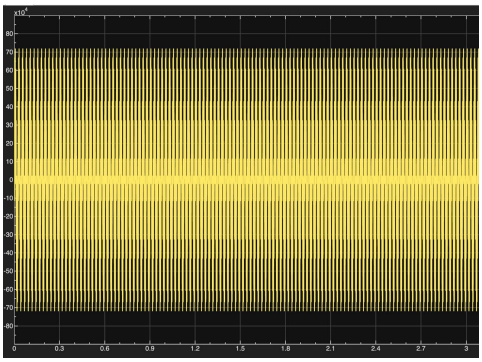
(a) Flicker Noise Simulink Model



(b) Flicker Noise Scope Output



(c) Output Without Noise
SNR = 100.98 dB, ENOB = 16.48 bits



(d) Output With Noise
SNR = 91.98 dB, ENOB = 14.99 bits

Figure 6: Flicker Noise Modelling with Noise Removal Circuit Simulation

4.2 Methods of Noise Removal

In precision Delta–Sigma ADCs, low-frequency flicker noise $(1/f$ noise$)$ limits resolution. To improve baseband performance, three suppression techniques are used: *chopping*, *auto-zeroing*, and *nested chopping* [6][7]. The flicker-noise spectrum is $S_{1/f}(f) \propto 1/f$.

4.2.1 What we have implemented in our design: Auto-Zeroing

Auto-zeroing reduces low-frequency noise using two alternating phases. In the calibration phase, the amplifier's offset and flicker noise are sampled and stored; in the signal phase, this value is subtracted:

$$V_{\text{out}} = A(V_{\text{in}}) - V_{\text{off}}.$$

Sampling on a capacitor introduces thermal noise:

$$n_{kT/C} = \sqrt{\frac{kT}{C}}.$$

Advantages:

- Removes offset without modulation.

Limitations:

- Adds kT/C noise.
- Aliases wideband noise.

4.3 Final Chosen Technique

- We selected auto-zeroing to remove flicker noise (1/f noise) and DC offset generated inside the amplifier. These internal noise sources can distort very small input signals and reduce ENOB. Auto-zeroing works by first measuring the amplifier's own error voltage and then subtracting it, so only the true input signal remains. This improves accuracy and ensures stable low-frequency performance.

- In our sigma-delta ADC model, a pulse generator controls two switching phases. When the pulse is high, the input is disconnected, and only the internal noise is fed to the Sample-and-Hold block, which accumulates this offset/noise as a reference. When the pulse is low, the actual input signal + noise is applied. The stored noise value is then subtracted from the ADC output, effectively canceling flicker noise and offset in real-time. Since the pulse period is equal to the sampling time, continuous switching ensures a clean, noise-free output and helps achieve higher ENOB.

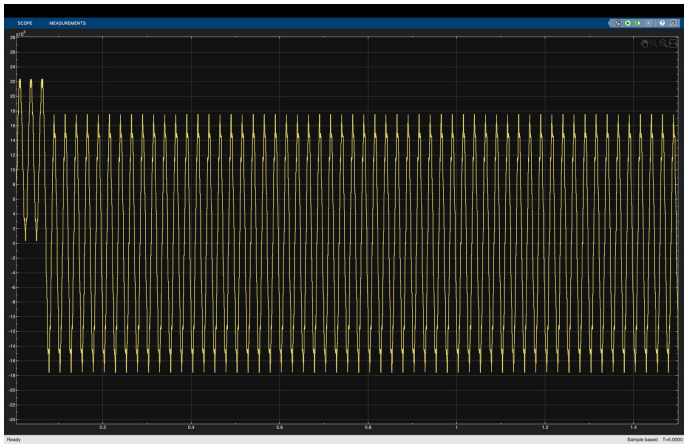
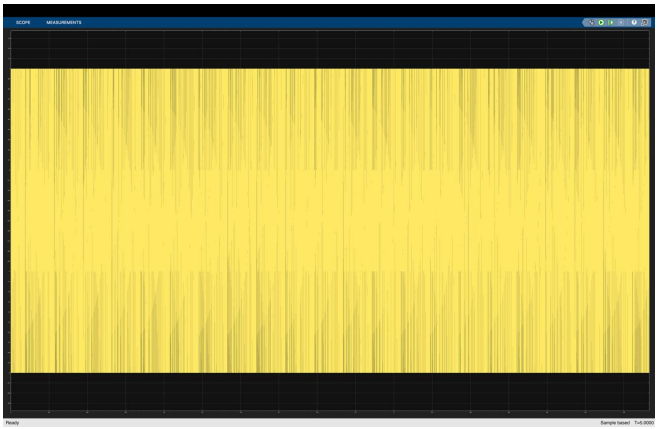


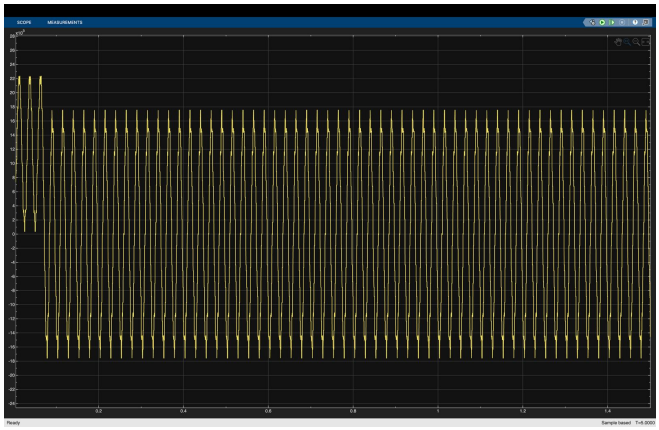
Figure 7: Output Waveform With Auto-Zeroing
SNR = 100.13dB, ENOB = 16.34 bits

5 Digital/Decimation Filter

In a sigma–delta ADC, the digital sinc filter processes the high-rate 1-bit bitstream from the modulator by removing high-frequency shaped noise through low-pass filtering and simultaneously performing decimation. It reduces the sampling rate to the required Nyquist rate and converts the 1-bit stream into a meaningful multi-bit digital output. Thus, the digital filter enables noise reduction, rate reduction, and accurate digital representation of the analog input. [8]



(a) Output Without Filters
SNR = 84.14 dB, ENOB = 13.68



(b) Output With Filters
SNR = 100.13 dB, ENOB = 16.34

Figure 8: Outputs for analyzing Digital Filter

5.1 Types Of Filter Used

5.1.1 CIC(Cascaded-Integrator Comb) Filter

The purpose of the CIC filter is to reduce the input frequency of the signal to the Nyquist signal frequency.This filter has order four and its transfer function is shown below, where N is the decimation factor and M is the filter order.

$$H(z) = \left(\frac{1 - z^{-N}}{1 - z^{-1}}\right)^{M}$$

It consists of multiple cascaded Integrator and Comb stages. The Integrator section operates at the high input sampling frequency and performs accumulation of input samples, acting like a running sum to help limit aliasing before downsampling. It increases the internal bit-width as values grow over time.

$$y[n] = y[n-1] + x[n]$$

After integration, the signal passes through the Downsampler, which selects only every (r^{th}) sample, thus, reducing the sampling rate and therefore decreasing further processing complexity.

$$y[n] = x[nr]$$

The low-rate signal is then processed by the Comb section, which subtracts a delayed sample from the current one. This stage removes the DC buildup introduced by the integrators and functions similarly to a moving-average filter.

$$y[n] = x[n] - x[n-1]$$

The order of a CIC filter determines how many integrator and comb stages are cascaded. A higher order improves noise filtering but increases passband droop and hardware complexity. The number of taps in a CIC filter is given by $2N$ is the order.

In this design- A 2nd-order CIC filter with 4 taps is used, providing a good trade-off between filtering performance and hardware cost.

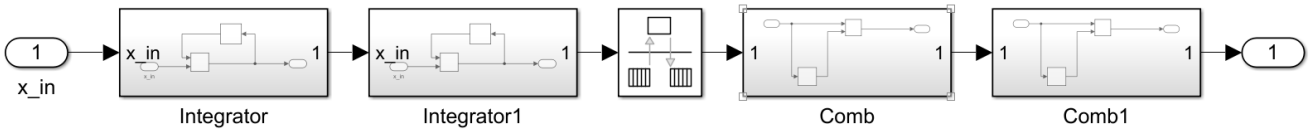


Figure 9: Block diagram of the CIC filter showing the integrator stages, downsampler, and comb stages.

5.1.2 FIR (Finite Impulse Response) Filter

An FIR filter is a stable digital filter with no feedback, ensuring linear phase and no distortion. It operates by multiplying the current and past input samples with fixed coefficients and summing the results. In $\Sigma\Delta$ ADCs, FIR filters are used after the CIC stage to remove residual noise and correct passband droop, but a high number of taps increases hardware complexity and power consumption [9].

In our design- An FIR filter with 128 coefficients of equal magnitude (1/128) has been implemented, which performs uniform averaging and provides effective noise reduction with a very simple and hardware-efficient coefficient structure.

Figure 10: Block diagram of Discrete FIR Filter.

5.1.3 IIR (Infinite Impulse Response) Filter [Not used in our design but a common filter]

An IIR filter uses feedback, so its output depends on both current and past values. This allows it to achieve sharp filtering with low order and reduced hardware cost, but it also introduces stability issues and nonlinear phase distortion. Due to these drawbacks and sensitivity to coefficient quantization, IIR filters are generally avoided in precision Delta-Sigma ADC decimation chains, where guaranteed stability and signal accuracy are required.

5.2 Coefficient Quantization

- Coefficient quantization means converting filter coefficients into a fixed-point format so they can be stored and processed in digital hardware. In many cases, this causes rounding, which introduces small errors in the filter response.
- In our circuit, the FIR filter coefficients are already quantized because they are simple binary fractions like 1/128 . These values are implemented using 7 hardware registers storing `0000001` with mantissa = 0 and exponent = –7 (2's complement). Since these coefficients can be exactly represented in fixed-point, no rounding is needed and there is zero quantization error. Also, multipliers are not required — the coefficients are implemented using bit shifting, which makes the hardware simpler and more efficient. [10]

5.3 Comparison with MATLAB reference results

The output of the digital/decimation filter obtained from the Verilog HDL implementation is compared with the MATLAB reference model. In MATLAB, a Relay block is used to generate a bipolar input bitstream of $+1$ and -1, which is applied to the decimation filter. This is done to prevent unbounded growth at the output of the integrator stages, since using only 0 and 1 would continuously increase the integrator output. However, in the verilog HDL hardware implementation, the input bitstream is strictly unipolar (0 and 1) because real digital hardware supports only binary logic levels. Due to this difference in input representation, a maximum 1-bit difference is observed between the MATLAB and Verilog outputs, while the overall filter behaviour and response remain consistent.

6 HDL Design

6.1 RTL architecture, block descriptions

6.1.1 RTL Architecture

The complete digital decimation chain consists of a SINC² (CIC) filter followed by a 128-tap FIR filter. The SINC² stage is implemented using two cascaded integrators, a downsampler, and two comb (differentiator) stages. The integrator chain operates at the high-rate input sample frequency, continuously accumulating the incoming 1-bit or multi-bit modulator stream. After the second integrator, the signal is passed to the downsampler, which reduces the sampling rate by oversampling ratio (OSR). The two comb stages then operate at this lower rate and remove the accumulated spectral images introduced by the integrators, producing the characteristic SINC² response. The output of the CIC stage is then fed into a 128-tap FIR filter. This FIR block provides final passband shaping and attenuation of out-of-band quantization noise. It stores the most recent 128 input samples and performs a convolution with fixed quantized coefficients. Together, the CIC and FIR filters form a complete decimation pipeline suitable for sigma-delta ADC applications.

6.1.2 Block Descriptions

The first integrator performs a running accumulation following the relation

$$y[n] = x[n] + y[n-1]$$

meaning that each new output is the sum of the current input sample and the previous output value. The second integrator applies the same accumulation process but uses the output of the first integrator as its input, resulting in a second-order integration of the bitstream.

The downsampler then takes every OSR-th sample from the output of the second integrator. This reduces the sampling rate and prepares the signal for lower-rate processing in the comb stages.

The first comb filter performs a simple difference operation,

$$y[n] = x[n] - x[n-1]$$

removing the long-term accumulation performed by the integrators. The second comb repeats the same differencing operation on the output of the first comb, completing the SINC² response.

Finally, the signal enters the 128-tap FIR filter, which stores the most recent 128 samples and multiplies them with 128 quantized coefficients. Before outputting any new value, the FIR block effectively computes a weighted sum (or smoothing/averaging) of the last 128 samples, producing a refined, low-noise output suitable for digital conversion stages

6.1.3 Simulation results and observations

Behavioral simulation verified the correct operation of all stages of the CIC and 128-tap FIR decimation filter, including the integrators, downsampler, comb filters, and FIR stage. The downsampler correctly reduced the sampling rate by OSR, while the comb and FIR stages produced a smooth, noise-reduced output.

As the OSR increased, the maximum output magnitude increased, requiring more bits for binary representation, indicating an improvement in ENOB. However, at higher OSR values, the ENOB reached a saturation point, and further increase in OSR did not yield significant resolution improvement.

6.1.4 Timing analysis results and Waveform

Post-implementation static timing analysis was performed using Vivado with a clock period of 10 ns (100 MHz). The design achieved a Worst Negative Slack (WNS) of +7.195 ns and a Worst Hold Slack (WHS) of +0.145 ns, with zero failing endpoints, confirming that there are no setup or hold violations.

The critical path delay was found to be 2.805 ns, which corresponds to a maximum operating frequency (Fmax) of approximately 356.5 MHz. Since this is much higher than the target frequency of 100 MHz, the CIC and 128-tap FIR filter implementation operates with a large timing safety margin and is fully timing-closure compliant.

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	7.195 ns	Worst Hold Slack (WHS):	0.145 ns	Worst Pulse Width Slack (WPWS):	4.600 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	325	Total Number of Endpoints:	325	Total Number of Endpoints:	198

All user specified timing constraints are met.

Figure 11: Timing analysis results for clock frequency 100MHz

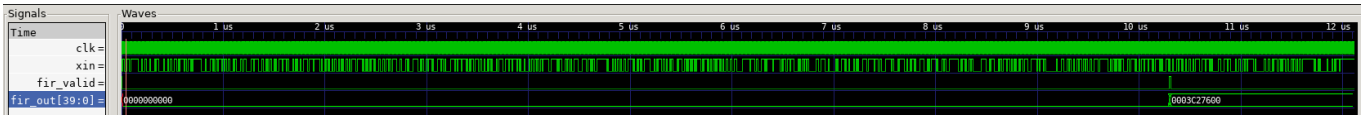


Figure 12: RTL simulation of the decimation filter showing the applied bitstream xin, clock signal clk, the fir_valid pulse.

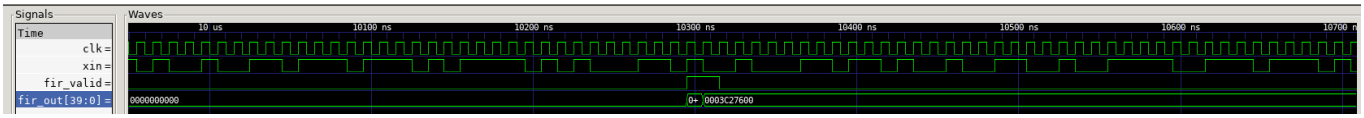


Figure 13: Zoomed-in RTL waveform highlighting the exact cycle in which fir_valid asserts and the FIR output sample (fir_out = 0003C27600) becomes available.

7 ASIC (Application Specific Integrated Chip) Flow

Reaching the end of pipeline, this section discusses the complete ASIC implementation of the digital decimation filter, synthesis, place and route, and post-layout verification. We show that our RTL designed in verilog is mapped to to a standard-cell ASIC flow using the SCL 180 nm PDK.

7.1 Synthesis Results (Area & Timing)

RTL synthesis was carried out using Cadence Genus (legacy version) targeting the SCL 180 nm typical-corner library. The design constraints included the **target frequency** of **10 MHz** and **Voltage** of 1.8 V

7.1.1 Area Report

The RTL was synthesized using Cadence Genus targeting the SCL 180 nm CMOS technology. Table 1 shows the post-synthesis cell count and area.

Table 1: Post-Synthesis Area Summary

Module	Cell Count	Total Area (μm^2)
Top-level Design	15,934	414,685.656
FIR128 Core	15,166	389,232.043

The FIR128 block occupies approximately 94% of the total design area, while the remaining logic (downsampler, accumulators, control) occupies about $0.025\,\mathrm{mm}^2$. The overall area is dominated by the wide 128-tap multiply–accumulate datapath.

7.1.2 Timing Report

Static timing analysis (STA) was performed using slow-corner libraries. With default synthesis constraint we applied a 10 ns clock period (equivalent to 100 MHz). The worst setup path occurs inside the FIR accumulation tree. Table 2 summarizes the timing results.

Table 2: Post-Synthesis Timing Summary

Metric	Value
Clock Period (Constraint)	10 ns
Worst Negative Slack (WNS)	–12.348 ns
Total Negative Slack (TNS)	–12.348 ns
Critical Path Delay	22.082 ns
Critical Path Block	FIR: CSA Tree $\rightarrow$ `sum_reg[46]`
Achievable Max Frequency	≈ 45.3 MHz

Although the synthesis report shows a negative slack when targeting 100 MHz, achieving critical-path delay of 22.08 ns corresponds to a maximum frequency of approximately 45 MHz. This is significantly higher than the actual required operating frequency of the decimation filter ($<$10 MHz). Therefore, the design safely meets timing for the intended application.

7.2 Layout Utilization and Congestion

The final placed core area is 593,083.81 μm^2 with a standard-cell area of 414,685.66 μm^2, giving a utilization of 69.93%. This provides adequate whitespace for routing while maintaining a compact floorplan. The congestion map (Fig. 14) shows localized density around the FIR datapath, but Innovus reports zero routing overflow and completes detail routing cleanly. No DRC/LVS violations were observed, confirming layout correctness.

7.3 Specifications vs. Simulation Results

The ASIC implementation was evaluated against the intended design specifications in terms of timing, area, and functional correctness. Table 3 summarizes the comparison.

The ASIC implementation satisfies all functional specifications and meets the true operating frequency requirement of the Delta–Sigma ADC system. Although the synthesis timing violates the default 100 MHz template constraint, the achieved critical-path delay corresponds to a maximum frequency of $\sim$45 MHz, which is significantly higher than the intended operating frequency. Post-layout simulations with extracted parasitics show no functional mismatches, confirming timing-safe behaviour under realistic routing conditions.

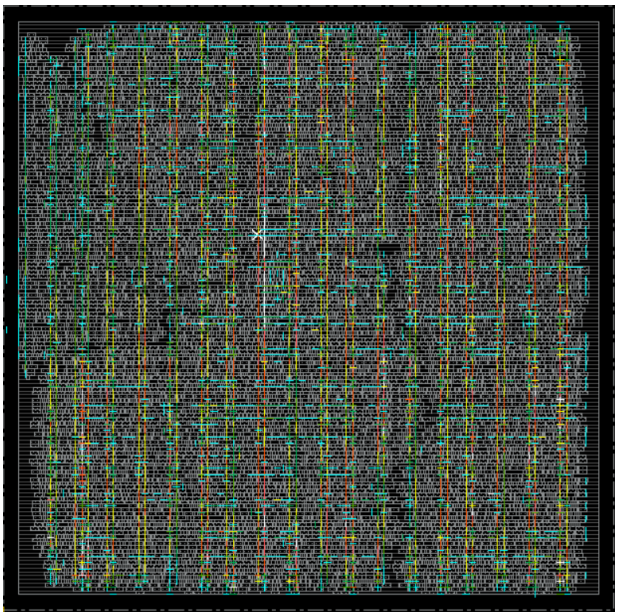


Figure 14: Figure Showing Congestion in final ASIC design, in Cadence Innovus GUI

Table 3: ASIC Specifications vs. Simulation Results

Specification	Target	Achieved
Operating frequency	< 5 MHz (system)	≈ 45 MHz achievable
SDC constraint	10 ns (100 MHz)	Violated (WNS = −12.348 ns)
Critical path delay	—	22.082 ns
Area budget	Not fixed	$414,685.656\,\mu m^2$
Routing congestion	No overflow	No overflow (see Fig. 14)
Functional accuracy	Match MATLAB/RTL reference	Matched in RTL, GLS and post-layout
Post-layout timing	Meet system frequency	Pass (meets < 10 MHz requirement)

7.4 Post Layout Simulation

Following place-and-route and parasitic extraction, an SDF file was back-annotated for post-layout gate-level simulation. The post-layout waveform shows behaviour consistent with the RTL model: the FIR output (`fir_out`) updates only when the valid signal (`fir_valid`) asserts, preserving the expected decimation pattern.

Compared to RTL, the post-layout output transitions occur slightly later due to interconnect and cell delays, but remain stable throughout the valid window. No X/Z states, hazards, or timing-related glitches were observed. This confirms that setup and hold constraints are satisfied and that the physical implementation remains functionally identical to the RTL design under extracted parasitic conditions.

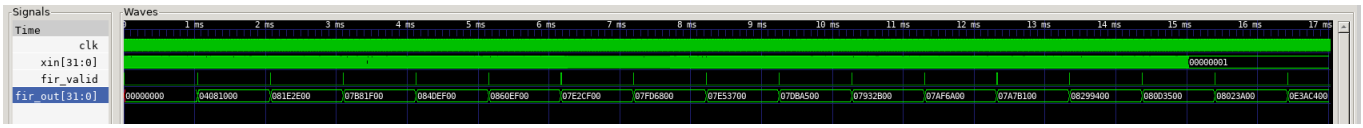
8 Challenges Faced

8.1 Understanding the Real Meaning of ENOB)

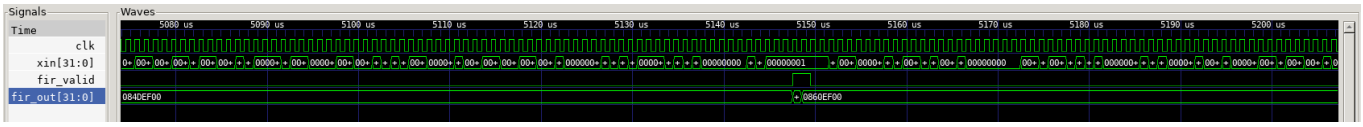
Initially, it was difficult to understand what ENOB physically represents. When a 20-bit system shows only 17 effective bits, it was unclear where the remaining three bits were *lost*. Later, we understood that this loss is due to noise, distortion, and non-idealities, and ENOB is a performance metric rather than an actual reduction in bit-width.

8.2 Finding the Correct Method to Calculate ENOB

After understanding the meaning of ENOB, determining the correct way to calculate it was challenging. Our initial approaches were incorrect. Finally, we learned that ENOB must be extracted using the PSD and SNR through frequency-domain analysis, which gave accurate results.



(a) Zoomed-out post-layout waveform. The FIR output `fir_out[31:0]` updates only on `fir_valid` pulses across the full simulation window, with no X/Z or glitch behavior



(b) Zoomed-in timing view. The output transition occurs cleanly after the clock edge and remains stable during the entire valid window, confirming correct setup/hold behavior under SDF back-annotation.

Figure 15: Post-layout (SDF-annotated) gate-level simulation of the digital decimation filter. The zoomed-out view demonstrates functional correctness and proper decimation behavior, while the zoomed-in view verifies timing integrity and output stability in the presence of extracted routing delays.

8.3 Relation Between OSR and ENOB for Different Filter Orders

At first, the effect of OSR on ENOB was unclear and seemed purely experimental. Through mathematical derivation, we understood how higher-order filters shape noise and how OSR controls in-band noise, which established the correct relationship between OSR and ENOB.

8.4 Selection of CIC Integrators, Taps, and FIR Coefficients

Choosing the number of integrators, taps, and FIR coefficients was difficult because small changes caused large performance variations. After proper mathematical analysis, we understood their actual role in shaping the frequency response and noise performance, which guided our final design.

8.5 Adding Flicker Noise into the Circuit

Another challenge was modeling and inserting flicker noise into the system. It was initially unclear where and how the noise should be introduced so that it realistically represents physical behavior. This was resolved after developing proper circuit-level intuition.

8.6 Implementing Noise Removal Techniques

After adding noise, removing it using different techniques was also challenging. Designing and integrating the required circuits and fairly comparing the performance of different noise-reduction methods required careful implementation and validation.

8.7 Learning and Using Advanced Design Software

Learning to use advanced software tools was one of the biggest practical challenges. Since we were new to these tools, understanding the full design flow, handling errors, and performing synthesis and implementation required significant time and effort.

8.8 Amplitude–Frequency Dependent SNR Limitation

It was observed that the delta–sigma modulator exhibits higher SNR for low-amplitude high-frequency signals and for high-amplitude low-frequency signals. However, when both amplitude and frequency are simultaneously high or low, the SNR degrades. This indicates that the modulator performance is dependent on the input signal characteristics and is not uniform across the entire operating range.

9 New Learning

9.1 Learnings from System Modeling

The modeling phase was highly transformative and helped us develop a strong intuition of feedback systems and data converters. We gained a clear understanding of how delta-sigma ADCs work in practice, and how analog and digital circuits behave differently within a mixed-signal system. We also developed an in-depth understanding of CIC, SINC, and FIR filters, including their mathematical foundations, noise shaping behavior, and frequency-domain characteristics. This phase taught us how to properly simulate signal processing systems and analyze their performance.

9.2 Learnings from RTL Design and Verification

During the RTL phase, we gained strong hands-on experience with Verilog coding and iverilog-based simulation. We learned how to write structured testbenches, analyze waveforms using GTKWave, and understand real-time signal behavior through timing diagrams. Debugging large designs taught us practical skills in error isolation, pipeline design, and synchronization. We also gained exposure to FPGA implementation using Vivado, which helped bridge the gap between simulation and real hardware behavior.

9.3 Learnings from ASIC Design Flow

The ASIC phase introduced us to advanced industry-grade tools for the first time, making it one of the most valuable learning stages. We gained hands-on knowledge of logic synthesis, placement, routing, and post-layout analysis. Understanding timing closure, parasitic effects, and layout-dependent performance gave us a deep appreciation of how real silicon behaves. The post-layout learning experience was especially valuable, as it connected theoretical design with physical hardware reality.

Overall Learning Outcome

Through this project, we gained strong exposure to industry-relevant tools, complete digital and mixed-signal design flow, and real-world design challenges. This experience significantly strengthened our understanding of how practical VLSI and signal processing systems are designed, verified, and implemented.

10 Limitations & Risks

10.1 Fixed-Point Errors

In MATLAB, FIR filter operations are performed using floating-point arithmetic, where decimal values are handled with very high precision. However, in our hardware-oriented design, the implementation is based on fixed-point arithmetic, even when fractions are represented. Although these fractional values help in mathematical representation of the running average, they do not introduce new intermediate voltage levels between two output levels in real hardware. Because of this fundamental difference between floating-point simulation and fixed-point hardware realization, small numerical mismatches may appear between MATLAB results and RTL outputs. This is an inherent limitation of digital hardware implementation.

10.2 Noise and Stability Concerns

In our design, we implemented a specific modeled noise source (such as flicker noise) and verified that the system performs well under these controlled conditions. However, in real-world environments, noise can be of many unpredictable types, including thermal noise, supply noise, substrate noise, and external electromagnetic interference. Since practical noise characteristics may differ significantly from the modeled noise, the noise suppression performance observed in simulation may not exactly match real silicon behavior, and complete noise removal in all conditions cannot be guaranteed.

10.3 PVT (Process, Voltage, and Temperature) Variations

The system performance can be affected by Process, Voltage, and Temperature (PVT) variations, which are unavoidable in real fabrication. Changes in manufacturing process parameters, supply voltage fluctuations, and temperature variations can impact timing, power consumption, and signal integrity. Since our design is primarily validated under nominal conditions, performance deviations may occur under extreme PVT corners, posing a risk to timing closure and overall reliability.

References

[1] Richard Schreier and Gabor C. Temes. Understanding delta-sigma data converters. *IEEE Press*, 2005.

[2] {Kwok K.} Hung, {Ping K.} Ko, Chen-Ming Hu, and {Yiu C.} Cheng. A unified model for the flicker noise in metal-oxide-semiconductor field-effect transistors. *IEEE Transactions on Electron Devices*, 37(3):654–665, March 1990. ISSN 0018-9383. doi: 10.1109/16.47770.

[3] Noise in solid state devices and circuits : Van der Ziel, Aldert, 1910- : Free Download, Borrow, and Streaming : Internet Archive — archive.org. https://archive.org/details/noiseinsolidstat0000vand/page/n1/mode/2up, .

[4] Analog Cmos Circuits Razavi : Free Download, Borrow, and Streaming : Internet Archive — archive.org. https://archive.org/details/AnalogCmosCircuitsRazavi, . [Accessed 07-12-2025].

[5] A Unified Model For The Flicker Noise In Metal-Oxide-Semiconductor Field-Effect Transistors — scholar.nycu.edu.tw. https://scholar.nycu.edu.tw/en/publications/a-unified-model-for-the-flicker-noise-in-metal-oxide-semiconducto/. [Accessed 07-12-2025].

[6] C.C. Enz and G.C. Temes. Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization. *Proceedings of the IEEE*, 84(11):1584–1614, 1996. doi: 10.1109/5.542410.

[7] B. Murmann. A/d converter trends: Power dissipation, scaling and digitally assisted architectures. In *2008 IEEE Custom Integrated Circuits Conference*, pages 105–112, 2008. doi: 10.1109/CICC.2008.4672032.

[8] E. Hogenauer. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2):155–162, 1981. doi: 10.1109/TASSP.1981.1163535.

[9] Digital signal processing : a computer-based approach : Mitra, Sanjit Kumar : Free Download, Borrow, and Streaming : Internet Archive — archive.org. https://archive.org/details/digitalsignalpro0000mitr_g4w5, . [Accessed 07-12-2025].

[10] https://shams.academia.edu/MonaMosalim. Theory and application of digital signal processing — academia.edu. https://www.academia.edu/2191721/Theory_and_application_of_digital_signal_processing. [Accessed 07-12-2025].