INTER IIT TECH MEET 14.0

ISRO – VLSI Mid-Prep

End-Term Report

High-Resolution Delta–Sigma ($\Delta\Sigma$) ADC Design

Team ID: Team_82

December 7, 2025

Contents

Abstract

This work presents the complete system-level design, modeling, and hardware implementation of a high-resolution $\Delta\Sigma$ analog-to-digital converter (ADC) targeted for low-bandwidth sensor applications in the 0.5–2 kS/s range. The study explores multiple modulator topologies (2^{nd}–4^{th} order) and identifies an optimal fourth-order discrete-time, single-bit modulator that achieves an effective number of bits (ENOB) above 16 bits. A behavioral noise analysis confirms that flicker noise dominates at very low frequencies; thus, architectural desensitization is employed instead of hardware chopping to maintain high ENOB without additional circuit overhead.

A multi-stage digital decimation chain, consisting of CIC, halfband, and FIR stages, is designed to suppress shaped quantization noise and reduce sampling rate while preserving signal integrity. Fixed-point simulation and HDL verification demonstrate close agreement with MATLAB reference results, with post-layout FPGA implementation confirming robustness. The final synthesized decimation filter, utilizes 5997 LUTs, exhibits a critical path delay of 13.05 ns, and reports zero negative slack after routing, indicating timing closure and DRC-clean implementation. The post-layout implementation reports a total power consumption of 2.57 mW and an occupied silicon area of 0.16677 mm^2, confirming the design's high energy efficiency and compact footprint suitable for ASIC integration.

The overall $\Delta\Sigma$ ADC system achieves an ENOB of 16–18 bits under ideal conditions and 15–17 bits with injected flicker noise, validating the design's robustness and power-efficient architecture for future ASIC integration.

1 Introduction

This work studies the design space of a high–resolution delta–sigma ($\Delta\Sigma$) analog–to–digital converter (ADC) targeting an effective number of bits (ENOB) of 16–20 bits at Nyquist–rate output data rates in the 0.5–2 kS/s range. Delta–sigma converters achieve such resolutions for low–bandwidth signals by combining large oversampling ratios with noise–shaped quantization in a feedback loop and subsequent digital decimation filtering, thereby relaxing the requirements on the quantizer and analog front–end compared with Nyquist–rate ADCs [1][2][3]

The core analog block is a $\Delta\Sigma$ modulator, implemented in discrete–time (switched–capacitor), continuous–time, or hybrid form, whose loop order, topology, and quantizer resolution determine the noise–transfer function (NTF), in–band quantization noise, stability margins and sensitivity to excess loop delay and jitter[1][4][5] . Multi–bit modulators with dynamic element matching (DEM) further trade quantization noise against DAC linearity and digital complexity[5][6] . On the digital side, multistage decimation chains based on CIC and FIR stages are used to remove out–of–band shaped noise and reduce the sampling rate to the target Nyquist rate while meeting passband–ripple and stopband–attenuation specifications [7][1]

.

Performance is quantified using signal–to–noise–and–distortion ratio (SNDR) and ENOB. For a sinusoidal test input, the widely used relation

$$\mathrm{ENOB} \approx \frac{\mathrm{SNDR}_{\mathsf{dB}} - 1.76}{6.02}$$

is obtained by inverting the ideal quantization–noise formula $\mathrm{SNR}_{\mathsf{ideal}} \approx 6.02N + 1.76\ \mathrm{dB}$ for an N–bit Nyquist–rate converter[8]. In this project, these models and metrics are used to compare alternative modulator and decimation–filter architectures over the specified 0.5–2 kS/s Nyquist range and to identify design points suitable for integration in an ASIC implementation flow.

2 Architecture and Methodology

The $\Delta\Sigma$ ADC considered in this work is split into two main parts: an analog $\Delta\Sigma$ modulator and a digital decimation filter. The modulator oversamples a low-frequency sensor signal at sampling frequency f_s and produces a 1-bit or few-bit noise-shaped bitstream. The digital decimation filter removes out-of-band quantization and residual flicker noise and reduces the sampling rate down to the desired Nyquist output rate $f_{\mathrm{Nyq,out}}$ (0.5–2 ksps and beyond).

At system level, the design is driven by a small set of parameters: maximum signal bandwidth, f_s, oversampling ratio (OSR), target ENOB and total decimation factor $D = f_s/f_{\mathrm{Nyq,out}}$. The chosen modulator order and OSR are those that can meet the ENOB target with reasonable analog complexity; the decimation filter is then designed to complete the noise shaping and decimation in the digital domain.

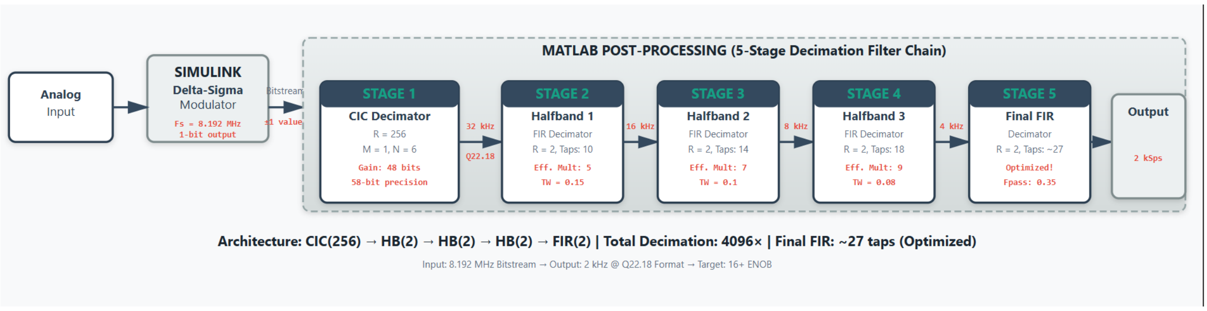

Figure 1: Top Level Architecture for Implemented Sigma Delta ADC

3 Design Journey and Toolchain Experiments

This section summarises the main practical steps taken during the project, from early Simulink prototyping to fixed-point, RTL and noise-aware behavioural analysis.

3.1 Exploring the PS and Simulink Toolboxes

We started by reading the PS and building a quick end-to-end prototype in Simulink using library blocks: a $\Delta\Sigma$ modulator from the mixed-signal toolboxes and CIC/FIR decimators from the DSP toolboxes. This provided a working modulator–CIC–FIR chain to verify basic behaviour (noise shaping, decimation, ENOB trends). However, the imported $\Delta\Sigma$ block is essentially a black box with no explicit access to internal noise sources, so it was used only for initial functional validation before moving to an explicit behavioural model controlled entirely from MATLAB.

3.2 MATLAB Engine + DSM Toolbox Parameter Optimisation

We then wrapped the DSM toolbox models with MATLAB scripts that sweep key parameters (modulator order, OSR, quantizer resolution, coarse decimation settings). For each configuration, the Simulink model was run automatically and SNR/SNDR/ENOB were logged. This allowed us to identify (order, OSR) pairs that achieve > 16 bits ENOB at 0.5–2 ksps while keeping OSR values realistic for the analog front-end; these operating points serve as references in the rest of the report.

3.3 From Toolbox Filters to Custom Multi-Stage Filters

At higher OSR and tighter ENOB targets, default toolbox CIC/FIR settings proved suboptimal: some presets attenuated part of the useful band at low Nyquist rates, whereas others provided insufficient stopband attenuation, causing ENOB loss even with a good modulator. We therefore computed passband/stopband edges directly from $f_{\mathrm{Nyq,out}}$ and OSR, partitioned the overall decimation factor into multiple stages (e.g. CIC, intermediate stages, final sharp FIR and optional halfbands), and designed custom multi-stage filters whose combined response meets the bandwidth and ENOB requirements. This structure was iterated jointly with the modulator operating point until > 16 bits ENOB was preserved in the system-level simulations.

3.4 HDL Coder and Fully Parallel Filter Implementation

After validating the fixed-point decimation filter, we used **HDL Coder** to generate synthesizable RTL. To simplify timing closure and keep the performance analysis clean, the critical FIR stages were implemented in a fully parallel style (one multiplier per tap), while halfband filters were inserted where they significantly reduced logic cost. The number and placement of halfband and compensation stages were tuned using synthesis reports and ENOB measurements, so that the RTL remains faithful to the MATLAB model yet realistic from an ASIC perspective, as detailed in Sections 7.

3.5 Noise Analysis with Coloured Noise Injection

Finally, we extended the ideal quantization-noise simulations to include low-frequency noise. A pink-noise source was injected at the modulator input in Simulink to emulate front-end flicker-like behaviour, and the complete modulator + decimation chain was re-simulated with ENOB recomputed. These experiments quantified the ENOB margin available once realistic low-frequency noise is present and provided a consistency check for the more detailed noise discussion in Section 5.

4 Modulator Study

This section briefly documents the $\Delta\Sigma$ modulator options considered, the simulation setup used to evaluate them, and the final architecture chosen for the rest of the work. The main figure of merit is the effective number of bits (ENOB) at Nyquist sampling frequencies in the 0.5–2 ksps range.

4.1 Modulator Architectures

Several modulator architectures were evaluated at system level using behavioural models (ideal integrators, quantizers and feedback DACs). Table 1 summarizes the main candidates.

Table 1: Summary of candidate $\Delta\Sigma$ modulators.

ID	Type	Order	ENOBs
M1	DT	2nd	16.43
M2	DT	3rd	16.84
M3	DT	4th	17.92

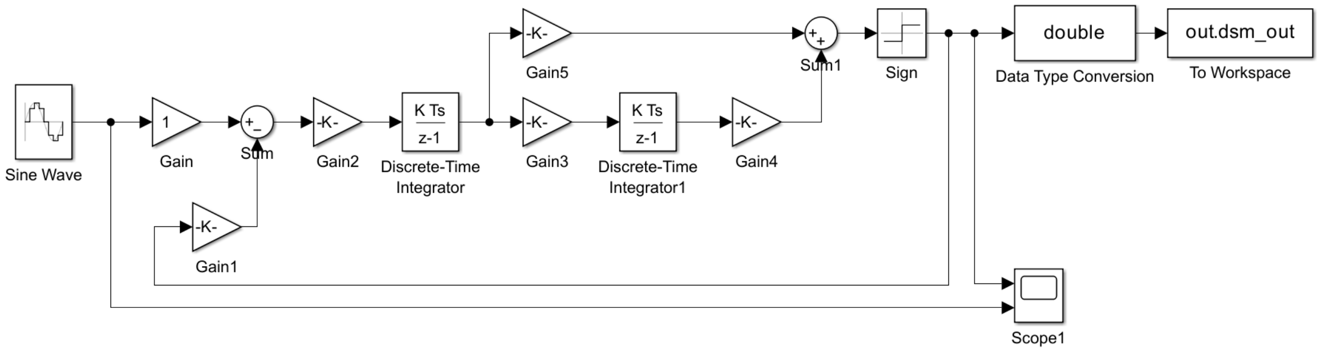

Figure 2: Matlab Architecture for 2nd Order Sigma Delta

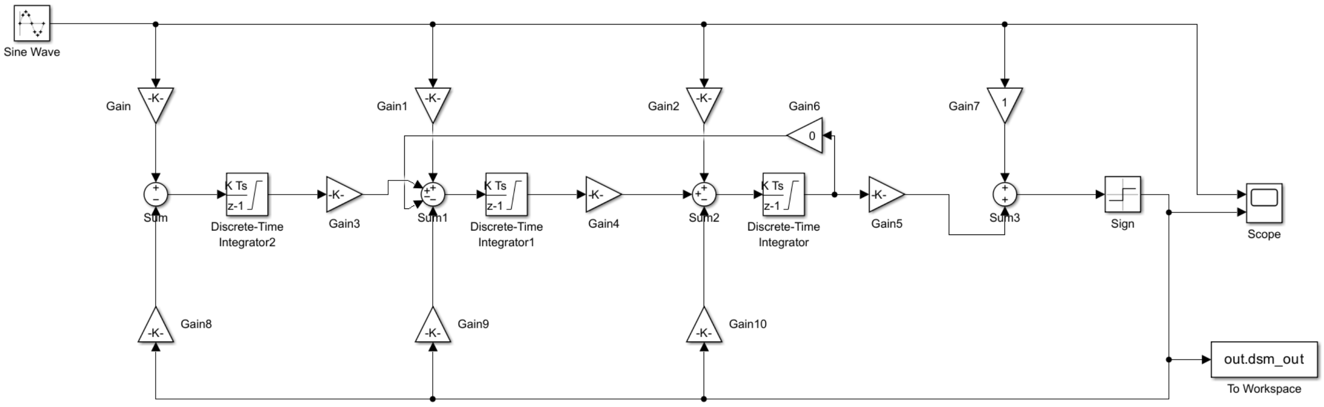

Figure 3: Matlab Architecture for 3rd Order Sigma Delta

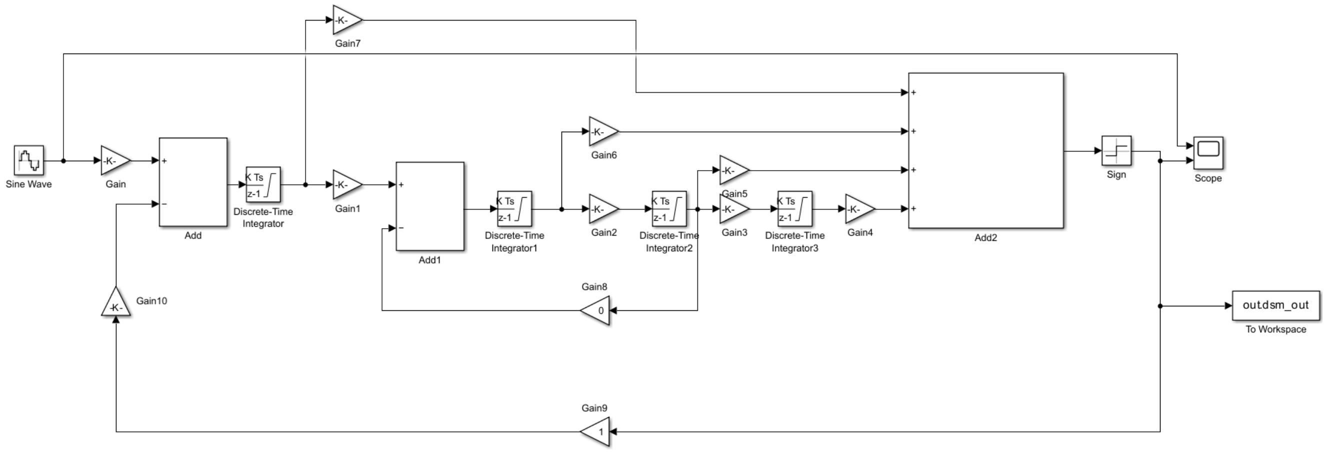

Figure 4: Matlab Architecture for 4th Order Sigma Delta

Behavioural simulations were then run to estimate ENOB for different Nyquist sampling frequencies in the target band. The 2nd- and 4th-order modulators demonstrated the strongest noise-shaping and thus offer the lowest noise performance among the candidates. Loop-filter coefficients were tuned for stability, and behavioural simulations were performed to estimate ENOB across different Nyquist sampling frequencies in the target band.

4.2 Simulation Setup

All modulators were evaluated under a common setup to allow fair comparison. A near full-scale single-tone input was applied within the signal band (typically around $0.4\, f_{\mathrm{Nyq,out}}$), the internal sampling frequency f_{s} was set by the chosen OSR, and the Nyquist output rate $f_{\mathrm{Nyq,out}}$ was swept over 0.5, 1.0 and 2.0 ksps. The resulting bitstreams were passed through an ideal or reference decimation filter consistent with the later digital design, and ENOB was computed from the output SNDR using the relation in Section 1. Additional noise-only runs (zero input) were used to verify that in-band noise levels matched theoretical expectations for each order and OSR.

4.3 ENOB versus Nyquist Sampling Frequency

Table 2 illustrates the comparison of Sampling Frequency Vs ENOBs for Order 2 Modulator.

Sampling Rate (ksps)	OSR	SNDR (dB)	ENOB (bits)
0.5	16384	75.38	12.23
1.0	8192	89.81	14.63
2	4096	100.72	16.44

Table 2: ENOB vs sampling rate in the 0.5–2 ksps range.

4.4 Selected Modulator and Rationale

Based on the ENOB results and implementation considerations, a single reference modulator is selected for the remainder of the work, summarised in Table 3.

Table 3: Chosen $\Delta\Sigma$ modulator for further design and analysis.

Attribute	Value / choice
Type	DT 4th order
Quantizer	1-bit
Nominal OSR	4096
Target ENOB	≥ 16 at 0.5–2 ksps

Justification. The selected modulator is the 4th-order discrete-time, 1-bit architecture, which achieves ENOB above 16 bits over the 0.5–2 ksps range even in the presence of input-referred flicker noise, while retaining comfortable stability margins at the chosen OSR. A 1-bit quantiser avoids multi-bit DAC linearity and DEM overhead, keeping the loop and digital calibration simple. The 2nd-order option shows a smaller but acceptable ENOB degradation with noise, whereas the 3rd-order candidate fails to maintain its nominal SNR under realistic non-idealities and was therefore discarded. Continuous-time and hybrid alternatives were not pursued further due to their higher sensitivity to clock jitter and analog implementation details, which lie outside the scope of this behavioural, digitally focused study.

5 Noise Analysis

Since ENOB is ultimately set by the in-band SNDR (cf. Section 1), only noise mechanisms that materially contribute within the signal band are retained[9]. For the low-frequency operation considered here, quantization and thermal noise can be pushed well below the target noise floor by suitable choice of modulator order, OSR and capacitor sizing, whereas device flicker ($1/f$) noise from the front-end dominates the input-referred noise budget[10]. In line with common practice, this total $1/f$ contribution is lumped into a single input-referred source in the behavioural model, with circuit-level mitigation techniques such as chopping or autozeroing implicitly captured by adjusting its power spectral density[11, 12, 13].

5.1 Flicker-Noise Mitigation Options

At circuit level, three standard techniques are considered to suppress in-band $1/f$ noise and offset in precision front-ends, together with an architecture-level option that reduces the sensitivity of ENOB to residual flicker noise. The key trade-offs for these options are summarised in Table 4; in this work their effect is captured at system level by scaling the effective in-band noise power according to the chosen technique and operating frequency.

Table 4: High-level comparison of flicker-noise mitigation techniques.

Technique	Principle	Advantage	Disadvantage
Chopping	Modulate signal up, process, then demodulate	$1/f$ and offset reduction	Extra switching, possible chopping ripple
Nested	Apply chopping in more than one stage	Strong $1/f$ and offset suppression	Higher complexity, area and power
Auto-zero	Sample and subtract offset / low-frequency noise	Simple concept, good offset trimming	Adds sampling noise, can limit bandwidth
Architectural	Choose/modify modulator architecture so that flicker-noise contribution is below the shaped quantization/thermal noise floor	Reduces ENOB sensitivity to device $1/f$ noise at system level	May require higher order/OSR, different topology, or increased power/area

5.2 Introduction Of Noise In The Circuit

In the behavioural noise simulations, only an equivalent input-referred flicker-noise source is injected at the modulator input, while separate $1/f$ contributions from internal integrators are neglected. For low-frequency $\Delta\Sigma$ modulators, the total input-referred $1/f$ noise is known to be dominated by the first integrator/front-end, so that device flicker noise can be accurately lumped into a single input source[14, 15]. This modelling approach is consistent with device-noise simulation methodologies that extract an equivalent input-referred noise power spectral density and apply it as the sole noise source in behavioural $\Delta\Sigma$ simulations[16].

Figure 5 shows the implementation of flicker noise with the addition of a colored noise block. For further analysis, number of samples per output channel was kept as 1.

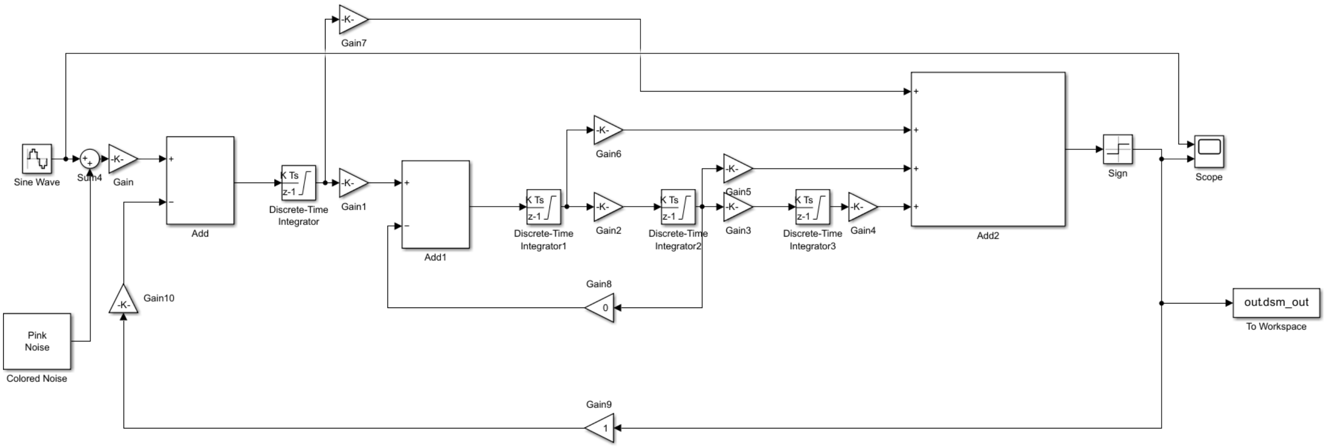

Figure 5: Matlab Architecture with Input Flicker Noise

5.3 Chosen Strategy and Impact on ENOB

Among the options in Table 4, this work focuses on *architectural* desensitization of flicker noise rather than explicit offset-cancellation hardware. By appropriately choosing the modulator topology, loop order and OSR, the input-referred $1/f$ contribution can be kept below the shaped quantization and thermal noise floor, so that ENOB is effectively limited by the architecture rather than by device flicker parameters[5, 4]. Compared to chopping or auto-zeroing, this approach avoids additional switches, clocks and calibration logic, does not introduce chopping ripple or extra sampling noise, and therefore achieves nearly the same in-band noise performance without increasing front-end hardware complexity in the ASIC implementation[6].

DSM Order	**SNR (dB)**	**ENOB (bits)**	**SNR with Noise (dB)**	**ENOB with Noise (bits)**
2nd	100.66	16.43	92.60	15.09
3rd	103.13	16.84	26.3818	4.09
4th	109.63	17.92	102.77	16.70

Table 5: Comparison of SNR and ENOB for different DSM orders, with and without added noise.

Table 5 compares the SNR and ENOB for different modulator orders, with and without the injected input-referred flicker noise. For the 2nd-order design, ENOB drops from 16.43 bits to 15.09 bits in the presence of noise, indicating a noticeable but not catastrophic degradation: the architecture still preserves most of its ideal performance. In contrast, the 4th-order modulator maintains an ENOB of 16.70 bits with noise (versus 17.92 bits ideally), comfortably satisfying the 16-bit target even under realistic conditions.The 3rd-order topology fails to sustain its nominal SNR once noise is introduced, highlighting its higher sensitivity to non-idealities. Overall, these results support the choice of architecture-level desensitization as the preferred mitigation strategy: a suitably chosen modulator order and OSR can limit the impact of flicker noise on ENOB without resorting to additional chopping or auto-zero hardware.

6 Digital / Decimation Filter

The digital decimation filter converts the high-rate bitstream from the $\Delta\Sigma$ modulator into a low-rate, high-resolution output at the desired Nyquist sampling frequencies (0.5–2 ksps). It has two main functions: (i) suppress out-of-band shaped quantization noise and (ii) decimate the sampling rate from the internal modulator frequency f_s down to the target output rate $f_\mathrm{Nyq,out}$ defined in Section 2. The filter must satisfy both spectral requirements (passband ripple, stopband attenuation) and implementation constraints (word length, area, maximum clock frequency).

6.1 Design Goals

At system level, the decimation filter is specified by: (i) a signal band equal to the sensor bandwidth used in the modulator study (Section 4); (ii) a stopband edge sufficiently below $f_\mathrm{s}/2$ to reject shaped quantization noise; (iii) passband ripple and stopband attenuation chosen such that the filter's in-band distortion and noise do not reduce ENOB below the target; and (iv) a total decimation factor $D = f_\mathrm{s}/f_\mathrm{Nyq,out}$, which may be realised in one or more stages.

6.2 Filter Architecture

To keep hardware complexity reasonable, a multi-stage architecture is assumed, where early stages perform coarse decimation and later stages provide fine shaping of the passband and stopband. A typical split is illustrated in Table 6; actual numbers should be filled from the implemented design.

Table 6: Example decimation-filter stage breakdown (5-stage version).

Stage	Type	Decimation factor
S1	CIC / SINC	*256*
S2	FIR	*2*
S3	FIR	*2*
S4	FIR	*2*
S5	FIR	*2*

Early CIC/SINC stages exploit the fact that the modulator bitstream is highly oversampled and dominated by out-of-band noise, making simple integrator–comb structures attractive. A final FIR stage can then be used to correct passband droop and to achieve the desired stopband attenuation with modest tap count.

Figure 6 summarises the chosen multi-stage decimation filter architecture, showing each stage (CIC and subsequent halfband/FIR stages) together with its decimation factor, number of taps and effective multiplier count.

6.3 Fixed-Point Format and Word Lengths

Once the floating-point decimation architecture meets the system-level specifications, it is mapped to a fixed-point implementation. The main design variables are the input format (1-bit modulator

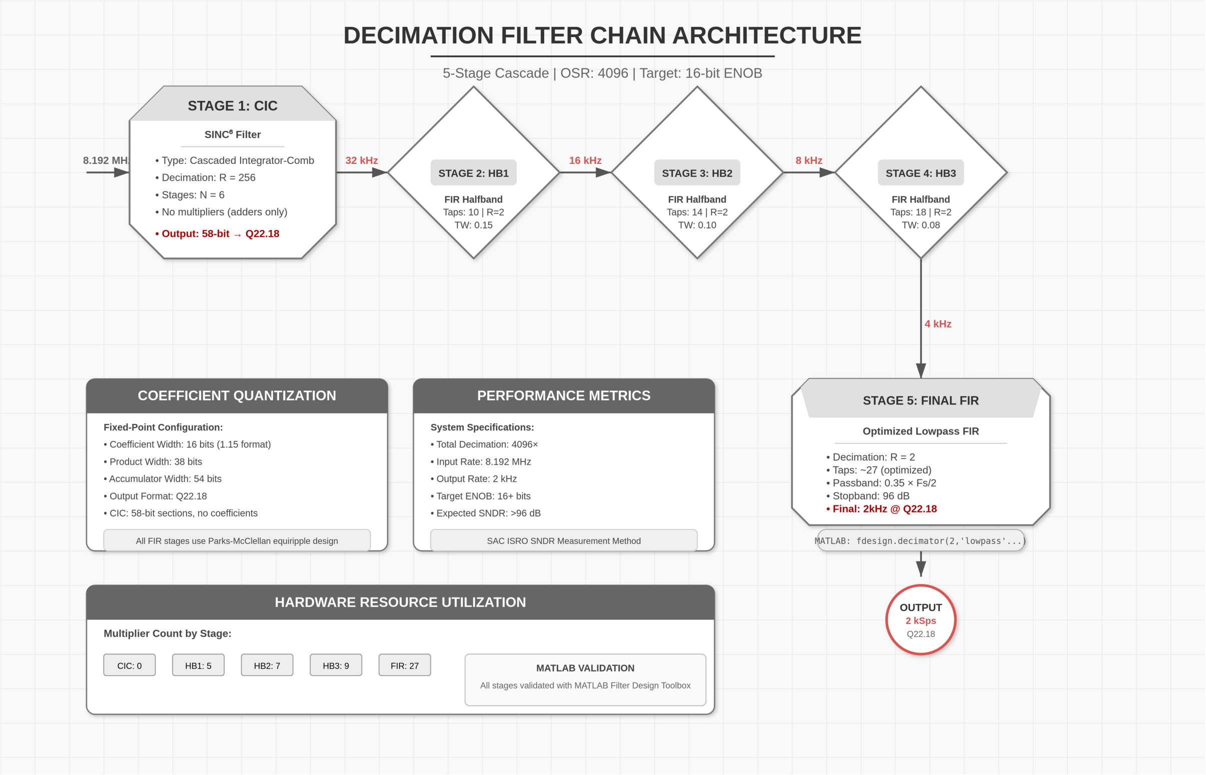

Figure 6: Filter Architecture

output), internal word lengths per stage (integrators, combs, FIR accumulators) and the output word length, chosen slightly above the target ENOB. In practice, wider words are used in the early integrator stages, where signal growth over D samples is largest, and widths are reduced in later stages after decimation. Fixed-point simulations are then used to verify that (i) no overflow occurs for full-scale input and (ii) the additional quantization noise due to truncation/rounding does not degrade ENOB below the target.

6.4 Validation Against MATLAB Reference

The fixed-point model of the decimation filter is validated in two steps:

1. **Model-level check:** apply the same modulator bitstreams to both the floating-point and fixed-point filter models in MATLAB and compute:

    - maximum absolute difference per sample,
    - RMS difference, and
    - ENOB / SNDR at the output.

    The differences should remain small enough that ENOB degradation is within an acceptable margin (e.g., less than 0.2–0.3 bits).

2. **RTL-level check:** implement the fixed-point filter in HDL and reuse the same test vectors. The RTL output is compared against the fixed-point MATLAB reference sample-by-sample, or via ENOB/SNDR measurements, to confirm functional equivalence.

In the subsequent sections (HDL design and ASIC flow), only this validated fixed-point architecture is carried forward, and its impact on timing, area and overall ENOB is discussed with respect to the system-level targets defined earlier.

7 HDL Design

The fixed-point decimation chain described in Section 6 is implemented as a synthesizable and ASIC-oriented HDL architecture. The objective is to translate the system-level filter stages—coarse decimation, halfband filtering, and final passband-correction FIR—into an area-efficient RTL hierarchy that preserves the numerical behavior observed in MATLAB simulations.

The complete decimator is packaged as a single top-level HDL block with clearly defined submodules corresponding directly to the algorithmic stages. This one-to-one mapping simplifies verification and keeps the RTL aligned with the fixed-point reference model throughout the design cycle.

RTL Structure and Module Hierarchy

The RTL hierarchy is intentionally kept shallow and modular to ease synthesis, timing closure, and integration into the ASIC flow. The following modules constitute the full decimation chain:

- `top_decim`: top-level wrapper that instantiates all stages and exposes the external interface.
- `cic_stage`: implements the coarse decimation using a single-stage CIC/SINC filter with an integrator–comb structure.
- `hb_fir_stage1`, `hb_fir_stage2`, `hb_fir_stage3`: three cascaded halfband FIR filters used for progressive bandwidth reduction, exploiting symmetric coefficients and zero taps to minimize multipliers.
- `fir_stage_A` and `fir_stage_B`: split implementation of the final compensation FIR filter, used to avoid excessive logic density and improve place-and-route convergence in ASIC.
- `ctrl_logic`: generates enable signals for the multi-rate stages and asserts `dout_valid` at the final output.

The top-level interface follows a synchronous streaming protocol:

- `clk`, `rst`: global clock and synchronous reset,
- `din`, `din_valid`: high-rate modulator bitstream input,
- `dout`, `dout_valid`: low-rate decimated output and its valid strobe.

All RTL is fully synchronous with a single clock domain and avoids gated clocks or inferred latches, ensuring synthesis portability and predictable timing behavior.

Implementation Considerations

The decimation chain is optimized for ASIC by (i) exploiting multiplier–free or multiplier–light structures (CIC and halfband filters) and splitting the final FIR to avoid large adder trees, (ii) matching internal word lengths to the MATLAB fixed-point model with explicit truncation/rounding and saturation to control bit growth, (iii) inserting pipeline registers only where required for timing, and (iv) keeping each FIR/HB block as a standalone, synthesizable module for early QoR checks.

Verification Methodology

Verification mirrors the system-level fixed-point flow. First, a fixed modulator bitstream and its decimated outputs are generated in MATLAB as reference. The bitstream is then exported to a Verilog-readable memory and used as deterministic stimulus in the HDL testbench. On each asserted `dout_valid`, the RTL output is captured and compared against the MATLAB reference (sample-by-sample and via ENOB/SNDR metrics); any deviations beyond the allowed error budget are traced back to rounding/truncation effects and used to refine the RTL arithmetic. This keeps the HDL implementation tightly aligned with the behavioural model and reduces risk at ASIC handoff.

Results

Table 7: Implementation summary for decimation filter RTL.

Metric	**Result**
LUTs	5997
Critical path delay	13.049 ns
Worst negative slack (WNS)	0 ns
Implementation state	Implemented and routed successfully
DRC status (post-route)	0 errors (DRC clean)
Routing status	0 failed nets, 0 unrouted, 0 partially routed nets
Global routing utilisation	Vertical: 4.46%, Horizontal: 6.90%
Congestion	No congested regions in any direction
Router mode	Resource-optimisation mode (no timing violations)

8 ASIC Implementation Results

Following RTL verification in MATLAB, the `top_decim` digital decimation filter was synthesized and implemented using industry-standard ASIC design tools and the UMC 90nm process design kit.

8.1 Design Flow

The complete ASIC implementation flow consisted of the following stages:

1. **RTL Design and Verification:** The decimation filter architecture was developed in Verilog HDL and functionally verified through MATLAB co-simulations to ensure correct frequency response and decimation behavior.

2. **Logic Synthesis:** The RTL design was synthesized using Cadence Genus with the UMC 90nm standard-cell library. Timing constraints were derived from the required sampling frequency f_{s}, with appropriate input/output delays to model the interface to the sigma-delta modulator and downstream digital processing blocks.

3. **Post-Synthesis Analysis:** Static timing analysis (STA) was performed to verify timing closure, and power analysis was conducted to estimate dynamic and static power consumption at the target operating frequency.

4. **Place and Route:** The synthesized gate-level netlist was taken through physical implementation using Cadence Innovus, including floorplanning, standard-cell placement, clock tree synthesis, and detailed routing.

5. **Post-Layout Verification:** Final timing analysis with extracted parasitics confirmed timing closure after accounting for interconnect delays. Post-layout simulations validated functional correctness of the final design.

8.2 Implementation Metrics

Table 8 presents the key metrics obtained from the ASIC implementation flow. The synthesized design successfully meets timing requirements with adequate margin, demonstrating the feasibility of the proposed decimation filter architecture in the target technology.

Table 8: ASIC implementation metrics for the digital decimation filter in UMC 90 nm technology.

Metric	**Achieved**
Technology node	UMC 90 nm
Required clock frequency	*8.192 MHz*
Post-Synthesis Results	
Total cell area	*0.167 mm*^{2}
Transistor count	*10,947 gates*
Total power @ f_{s}	*2.57 mW*

8.3 Discussion

The successful physical implementation of the decimation filter in UMC 90nm technology validates the proposed architecture.

9 Challenges Faced

Only the main practical difficulties are listed here, focusing on how they influenced the final architecture and implementation.

- **Finding optimal filter architecture for feasible ASIC synthesis while maintaining ENOB:** Aggressive filter designs achieving high ENOB in simulation often resulted in excessive area and timing violations during synthesis. Iterative refinement was necessary to balance signal quality with hardware constraints.

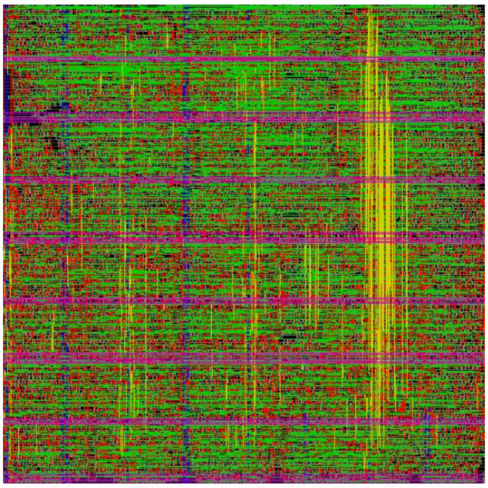

Figure 7: Post-layout view of the design after place-and-route, showing the final floorplan and routing of the decimation filter.

- **ENOB versus sampling rate trade-off:** Higher sampling rates improved ENOB but placed severe timing constraints on the digital filter. Finding the optimal operating point between signal quality and achievable clock rates proved iterative and technology-dependent.
- **Optimization of filter parameters for ENOB:** Filter order, number of stages, and coefficient precision all significantly impacted ENOB. Parameter sweeps were necessary to maximize effective resolution while remaining synthesizable.
- **Chopper circuit integration:** Integrating chopper modulation and demodulation to mitigate flicker noise introduced additional complexity. Synchronizing the chopper clock with the sigma-delta modulator and ensuring proper digital demodulation required careful timing analysis across the analog-digital interface.
- **Matching MATLAB and RTL behaviour:** Small differences in rounding and truncation between MATLAB and HDL initially led to ENOB mismatches; the arithmetic representation had to be aligned carefully.
- **Tool and flow constraints:** Limited access to full sign-off tools meant that only basic synthesis, timing and, at best, simplified layout flows could be run, leaving some uncertainty on power and detailed physical effects.

10 New Learning

The project helped connect theory and practice across several levels of the $\Delta\Sigma$ ADC design flow. Key learnings include:

10.1 System-Level Design

- The interplay between modulator order, OSR, and decimation-filter design in determining ENOB at low Nyquist sampling rates

- Low-noise architecture techniques beyond chopper stabilization: correlated double sampling (CDS), auto-zeroing, dynamic element matching (DEM), and kT/C noise considerations
- $\Delta\Sigma$ modulator topologies (CIFB, CIFF, CRFF), loop stability analysis, and quantizer resolution trade-offs

10.2 Digital Implementation

- Disciplined fixed-point design where every extra bit has clear cost in area and timing
- Impact of architectural choices (long accumulators, narrow margins, pipelining) on synthesis results and achievable clock frequency

10.3 Physical Design Tools

- **Cadence Genus**: RTL synthesis, constraint development, timing closure iteration, and critical path optimization
- **Cadence Innovus**: Floorplanning, power grid design, clock tree synthesis, and place-and-route optimization
- Iterative nature of design closure and when to modify RTL versus adjusting constraints or physical strategies

10.4 Cross-Domain Integration

Understanding the complete signal chain from analog circuit noise sources through quantization effects to digital filtering, and how design decisions propagate across domains.

11 Limitations and Risks

The current work should be viewed as a digital-centric proof of concept rather than a production-ready space-grade solution. The main limitations and risks are:

- **Behavioural analog model only:** The $\Delta\Sigma$ modulator is modelled with ideal or simplified non-idealities; a real analog front-end may achieve lower ENOB than predicted, especially under PVT variations.
- **Simplified noise and operating conditions:** Flicker and thermal noise are treated with high-level models and a limited set of input tones and conditions. Full PVT and statistical coverage is not performed.
- **Digital-only ASIC flow:** Only the decimation filter is taken through synthesis (and, if available, a basic layout). Mixed-signal integration, detailed power analysis and full sign-off checks remain open tasks for future work.
- **Potential specification changes:** If future system requirements change (bandwidth, power, target ENOB or sampling rates), the chosen combination of modulator architecture and decimation-filter structure may need to be revisited.

12 References

References

[1] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, 2004.

[2] J. C. Candy and G. C. Temes, “Oversampling methods for A/D and D/A conversion,” in *Oversampling Delta-Sigma Data Converters: Theory, Design, and Simulation*. IEEE Press, 1992.

[3] P. M. Aziz, H. V. Sorensen, and J. Van der Spiegel, “An overview of sigma-delta converters,” *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 61–84, 1996.

[4] F. Ortmanns and F. Gerfers, *Continuous-Time Sigma-Delta A/D Conversion: Fundamentals, Performance Limits and Robust Implementations*. Springer, 2006.

[5] Y. Geerts, M. Steyaert, and W. M. Sansen, *Design of Multi-Bit Delta-Sigma A/D Converters*. Kluwer Academic Publishers, 2002.

[6] I. Galton, “Delta-sigma data conversion in wireless transceivers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 1, pp. 302–315, 2002.

[7] E. B. Hogenauer, “An economical class of digital filters for decimation and interpolation,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 2, pp. 155–162, 1981.

[8] W. Kester, “Taking the mystery out of the infamous formula ‘SNR = 6.02N + 1.76 dB’,” Analog Devices, Application Note MT-001, 2005.

[9] J. Steensgaard, “Noise sources in switched-capacitor delta-sigma modulators,” in *Practical Approach to Delta-Sigma Design*, EPFL Short Course, 2004.

[10] R. Schreier, “Design-oriented estimation of thermal noise in switched-capacitor delta-sigma modulators,” *IEEE Trans. Circuits Syst. I*, vol. 52, no. 11, pp. 2358–2368, 2005.

[11] C. C. Enz and G. C. Temes, “Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling, and chopper stabilization,” *Proc. IEEE*, vol. 84, no. 11, pp. 1584–1614, 1996.

[12] H.-L. Chen, P.-S. Chen, and J.-S. Chiang, “A low-offset low-noise sigma-delta modulator with pseudorandom chopper-stabilization technique,” *IEEE Trans. Circuits Syst. I*, vol. 56, no. 12, pp. 2533–2545, 2009.

[13] Y. Han *et al.*, “A wide dynamic range sigma-delta modulator for EEG acquisition with chopper-stabilized amplifier,” *Sensors*, vol. 22, no. 3, 2022.

[14] 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, no. 6, pp. 1298–1308, Dec. 1988.

[15] S. Roy, *Design of a Power Efficient 16-bit Discrete Time Delta-Sigma Modulator for Audio Applications*. M.Tech. thesis, Dept. of Electrical Engineering, IIT Madras, Chennai, India, 2013.

[16] M. Terrovitis and K. S. Kundert, “Device noise simulation of delta-sigma modulators,” *The Designer’s Guide Community*, Aug. 1999, rev. 2006.