DESIGNCON® 2024
WHERE THE CHIP MEETS THE BOARD

Design Shift-Left: Enabling Early SerDes Mixed-Signal Design Beyond 200 Gb/s

Johnathan Adams, The MathWorks, Inc.
jadams@mathworks.com

David Halupka, SeriaLink Systems Ltd.

Aleksey Tyshchenko, SeriaLink Systems Ltd.

Marc Erickson, The MathWorks, Inc.

Pragati Tiwary, The MathWorks, Inc.

Richard Allred, The MathWorks, Inc.

Tripp Worrell, The MathWorks, Inc.

Barry Katz, The MathWorks, Inc.

Abstract

High-speed mixed-signal systems, such as 200+ Gb/s SerDes, use both analog and digital processing subsystems, often with complex interactions between them. The evolution of manufacturing processes and ever-increasing performance requirements for these systems drive increased system, digital/analog, and analog/digital interaction complexity. More complex interactions lead to longer design cycles and a higher probability of design issues.

In our previous paper, we showed that it is possible to left shift the validation effort, that is earlier in the workflow, by leveraging early architectural models to generate behavioral mixed-signal simulation models. In this paper, we show that the same automatic model generation capabilities can also be used to left shift a subset of the circuit design work earlier in the workflow. We use a SerDes system architectural model to generate an analog-to-digital converter (ADC) behavioral model, which we use instead of an actual circuit as the basis to design an ADC calibration scheme.

This way, the calibration system design no longer depends on a finalized ADC design. Instead, the calibration and ADC circuits design may be done in parallel, using the behavioral model, designing the two systems to work together. This workflow can potentially accelerate system design time by allowing interacting subsystems to be designed and validated in parallel.

Author Biographies

Johnathan Adams, Ph.D., is a software engineer at the MathWorks, Inc. He received his B.S. in 2017, M.S. in 2019, and Ph.D. in 2022 from Worcester Polytechnic Institute.

David Halupka, Ph.D., is a co-founder of SeriaLink Systems. David has over 20 years of experience in mixed-signal and embedded system design. He was with Kapik for 11 years, serving as a Senior System-Architect and Principal Engineer, and led the digital design team at Kapik. In 2018, he joined Intel's Mixed Signal-IP Group as Senior Systems Engineer, where he was responsible for adaptation algorithm development for the multi-standard SerDes. Dr. Halupka received his Ph.D, M.A.Sc., and B.A.Sc. from the University of Toronto.

Aleksey Tyshchenko, Ph.D., is a co-founder of SeriaLink Systems – a consulting team focusing on system modeling of high-speed serial links, IBIS-AMI modeling, model correlation, and system validation. SeriaLink Systems is working on building a configurable modeling flow to support SerDes projects through their entire life cycles, from architecture definition, through analog and digital design, to design validation. He has been working on behavioral modeling of high-speed SerDes systems, architecture analysis, adaptation, and signal integrity with multi-standard SerDes IP teams at V Semi and Intel. His Ph.D. research at the University of Toronto, Canada, focused on CDR systems for high-speed ADC-based receivers.

Marc Erickson has spent his career in the design, verification, and tool development for large ASICs and FPGAs at Intel, Teradyne, and as a consultant. For the last 15 years, he has been a technical lead at MathWorks, contributing to the HDL Verifier, SoC Blockset, SDR-Zynq, and Vision-Zynq products. Marc received his B.S.E.E. from Princeton University.

Pragati Tiwary is a principal software engineer at MathWorks, Inc., and also a team lead for the Mixed-Signal Blockset product. Prior to this he worked eight years at Cavium Inc. as a senior analog and mixed-signal design engineer. Pragati received his B.S. in Electronics and Communications Engineering from India's Birla Institute of Technology (2005) and his M.S. in Electrical and Computer Engineering from Carnegie Mellon University (2008).

Richard Allred, Ph.D., is a principal software engineer at MathWorks, Inc., with over a decade of signal integrity design experience at Intel, Inphi, and SiSoft. He has authored dozens of IBIS-AMI models and is currently developing signal integrity tools at MathWorks. Dr. Allred received his B.S./M.S. (2006) and Ph.D. (2021) from the University of Utah.

Tripp Worrell joined MathWorks in 2017 after spending three years at SiSoft managing the development of their award-winning EDA simulation software. Before this, he worked for seven years at Cisco Systems as a Signal Integrity and High Speed Design Engineer where he was responsible for design and verification of large enterprise networking linecards and backplanes. His current role as Development Manager overseeing SerDes Toolbox, Mixed-Signal Blockset, and Signal Integrity Toolbox leverages both his hardware and software experience to best support the needs of leading-edge clients at MathWorks. Tripp received his B.S. in Computer and Electrical Engineering (2005) and his M.S. in Computer Engineering (2007) from North Carolina State University.

Barry Katz, Director of Engineering, RF & AMS Products, leads the development teams responsible for RF, EM, Signal Integrity, SerDes, and Mixed-Signal modeling and simulation products at MathWorks. Before joining MathWorks, Barry served as President and CTO of SiSoft which he founded in 1995. At SiSoft, Barry was responsible for leading the definition and development of SiSoft's products. He has devoted much of his career to delivering a comprehensive design methodology, software tools, and expert consulting to solve the problems leading-edge high-speed systems designers face. Barry was the founding chairman of the IBIS Quality Committee. He received his B.S.E.E. from the University of Florida and his M.S.E.C.E. from Carnegie Mellon.

Introduction

High-speed analog-to-digital converters (ADCs) are now commonplace in SerDes receivers (112Gb/s and above), such as the one shown in Figure 1. In ADC-based SerDes receivers, the ADC is preceded by a CTLE that pre-conditions and partially equalizes the incoming signal. The ADC samples and quantizes the partially equalized analog signal into discrete digital samples at a rate of 1 sample per unit interval (UI). The ADC's baud-rate samples demultiplexed into frames of samples, to a sample rate that is suitable for digital signal processing. The digital feed-forward equalizer (FFE) and decision-feedback equalizer (DFE) fully equalize the received signal samples. The CDR uses a baud-rate phase detector to drive the VCO in feedback to achieve a 1 sample per UI sampling rate and appropriate sample phase alignment.

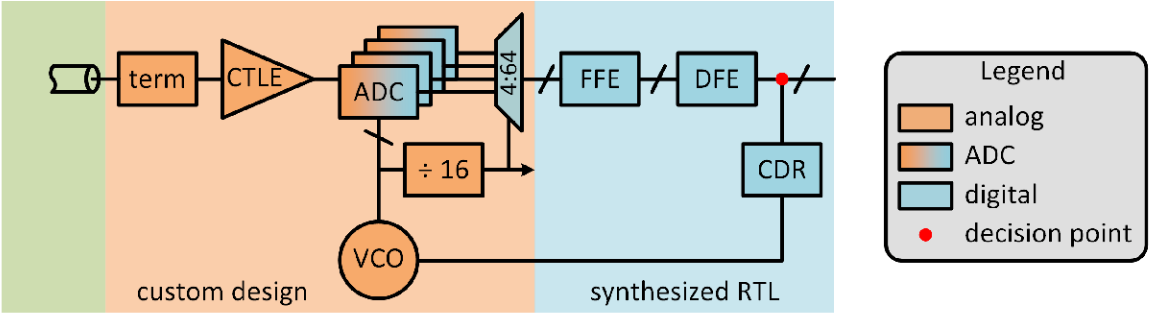

Figure 1. ADC based SerDes receiver system diagram.

The ADC samples the incoming analog data at 64GS/s or above, with an effective resolution of 5-8 bits [1],[2]. This sample rate exceeds the capabilities of an individual successive-approximation register (SAR) ADC and even an individual Flash ADC. Therefore, it is common to use time-interleaved ADCs, wherein a bank of individual ADCs are used in a round-robin fashion, to achieve the required aggregate sampling rate. This keeps the sampling rate of each individual ADC within a technologically achievable rate. To simplify analysis and maintain focus on the modeling approach rather than the design trade-offs, we will consider a four-way interleaved ADC. However, that is not to say that there is a specific limit to the ADC interleaving factor. There are a multitude of design decisions that need to be analyzed but are beyond the scope of this work.

Figure 2 shows a simplified block diagram of a four-way time-interleaved ADC, which utilizes four independent ADCs, ADC[0] to ADC[3], to sample and quantize the input signal. While each ADC might be an instance of the same design even down to the transistor level, each ADC will manifest imperfections differently due to manufacturing imperfections and random process variation. These manufacturing imperfections can be modeled, amongst other ways, as an input-referred offset and gain. To model the random manufacturing-induced differences, each ADC has independent gain and offset error parameter values. Unlike for a non-time-interleaved ADC, when left uncompensated, these errors reduce the overall effective number of bits (ENOB) of the time-interleaved ADC.

Offset and gain errors typically result from manufacturing induced mismatches between the various ADCs that make up a time interleaved ADC system [3]. Figure 3 shows the schematic of a single SAR ADC from the time-interleaved ADC presented by Luan et al. in [4]. Their full 64-way time-interleaved ADC design calls for 64 ADC cores to generate an 8-bit binary output. Other designs use Flash or other types of Nyquist-rate ADC architectures in similar interleaved converter systems [4],[5],**Error! Reference source not found.**.

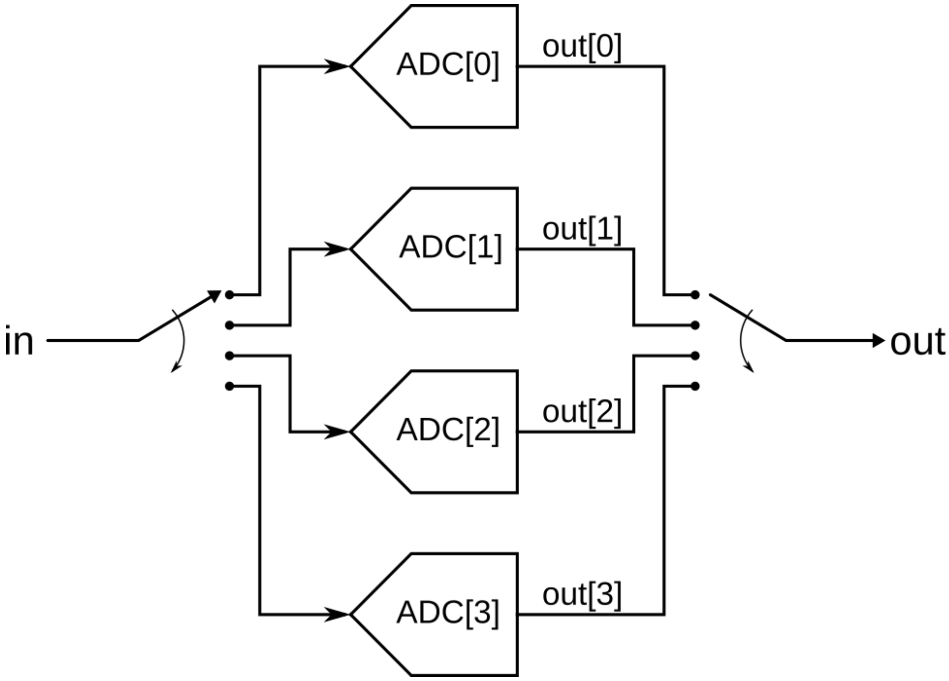

Figure 2. Simplified interleaved ADC block diagram.

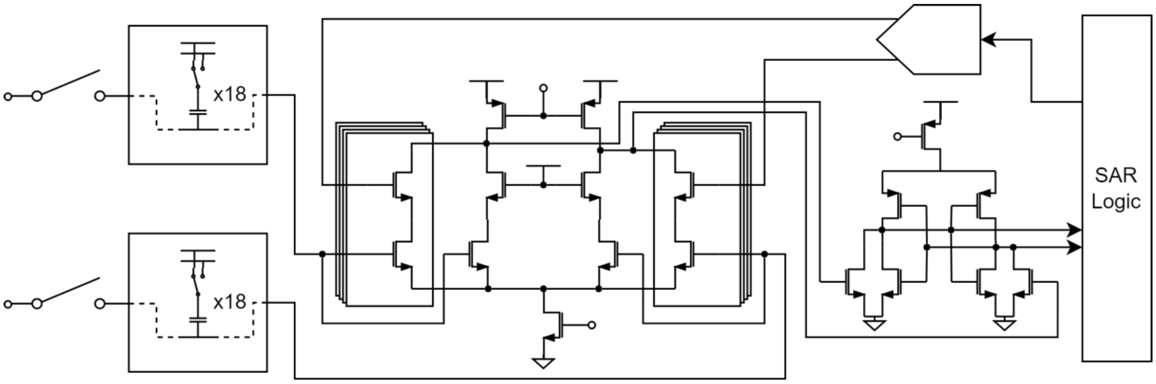

Figure 3. A schematic of a representative SAR ADC based on Luan et al. [4], wherein the full ADC interleaves 64 individual SAR ADCs.

In this work, we apply the "ABCs of SerDes Modeling" from [7] to synchronize the design of a mixed-signal feedback-based static-error-compensation system with the time-interleaved ADC it compensates. Architectural models, or A-models for short, model the system at a high level of abstraction and are used to develop requirements or evaluate different system architectures. Ideally, A-models are updated throughout the design process to better reflect the physical system they model. Behavioral models, or B-models for short, are used to fill gaps where more model detail is required than the A-model typically provides or to act as circuit model stand-ins when simulating the entire circuit may be prohibitive. Circuit models, or C-models, contain the most detail and complexity. A C-model gets closest to the performance of the implemented system, as it implements the design at hand, but is resource-intense to simulate compared to the other two types of models.

An architectural-level model (A-model) of the time-interleaved ADC is shown in Figure 4. This model contains static error impairments, which we discuss in more detail in Section 1. In Section 2, we choose a static error compensation algorithm and discuss its theory of operation. In Section 3, we generate a B-model of the ADC automatically, which we use in Section 4 to evaluate the performance of the static error compensation system before the ADC has been designed.

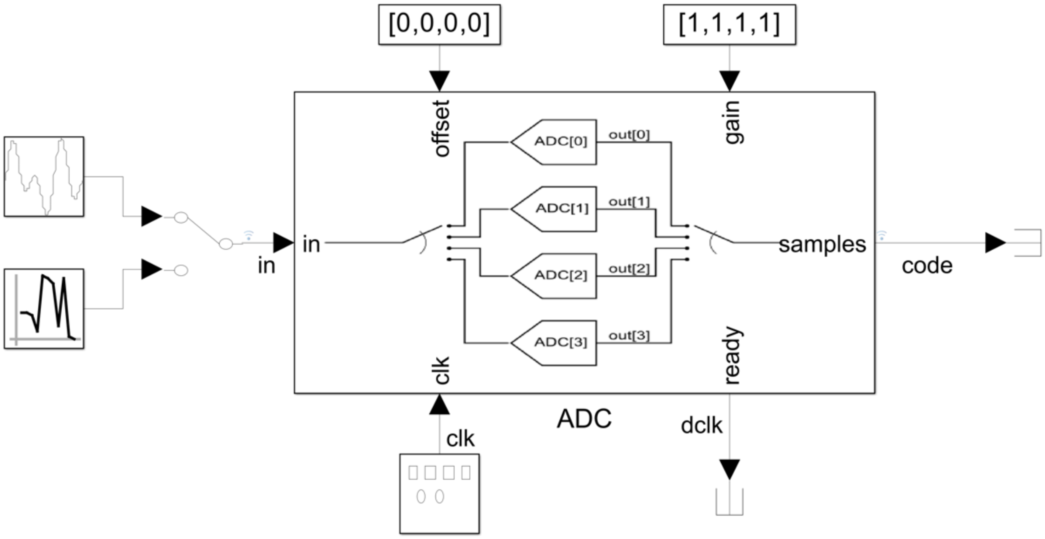

Figure 4. Architectural model of the time-interleaved ADC as implemented in Simulink®.

1 Modelling ADC and Impairments

ADCs have several performance characteristics that influence their signal-to-noise ratio (SNR) and effective number of bits (ENOB). However, this paper concentrates on offset error and gain error. We define these errors based on code transitions [8]. We typically define offset error as the difference between the actual and nominal levels associated with the first code transition. Subsequently, gain error is the difference between the actual and nominal levels associated with the last code transition[1] (after the offset error has been

[1] Sometimes, ADC gain error is expressed as a multiplicative factor, similar to operational amplifier gain error. This is another way to capture the same information.

removed). However, it makes more sense to use zero-scale error instead of offset error in the context of the differential signals of the SerDes receiver. Zero-scale error is identical to offset error except that it is measured from code 0[2], even if that code is in the middle of the converter's dynamic range [9]. This paper uses the term "offset error" to refer to the zero-scale error. Figure 5 shows how offset error and gain error can change the transfer curve of a stand-alone, non-time-interleaved 4-bit ADC.

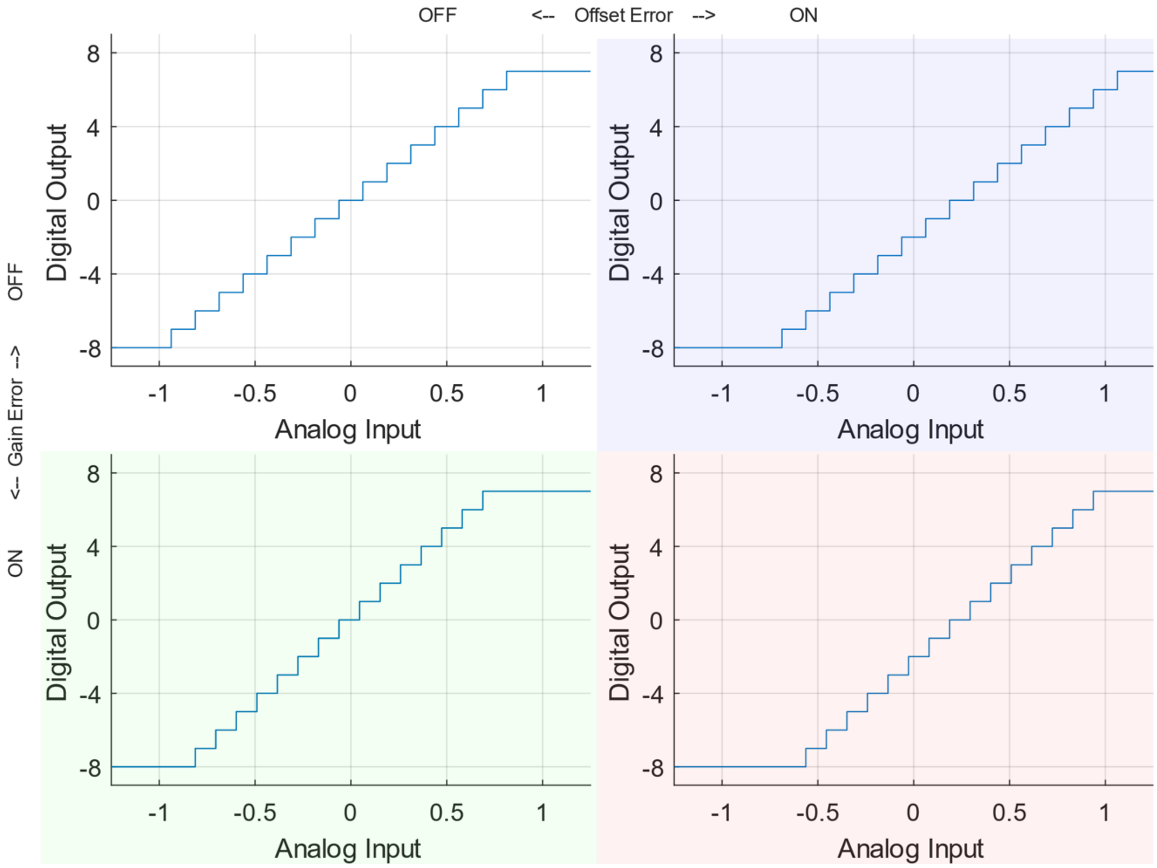

Figure 5. A 4-bit ADC's transfer curve showing (upper left) unimpaired, (upper-right) impaired with offset error, (lower left) impaired with gain error, and (lower right) impaired with both offset error and gain error.

To visualize how gain and offset mismatches between time-interleaved ADCs affect the linearity of the overall time-interleaved ADC, we compare four cases: an impairment-free, an offset mismatched, a gain mismatched, and an offset and gain mismatched time-interleaved ADC. The four cases are divided into four quadrants, as in Figure 5 and the subsequent plots shown in this paper; the quadrants are color-coded as follows:

- The impairment-free case (no offset and gain error) is plotted on a white background.

[2] Zero-scale error is measured from the transition between code 0 and code 1.

- The case with offset, but no gain error, is plotted on a blue background.
- The case with gain, but no offset error, is plotted on green background.
- The case with both gain and offset errors is plotted on a red background.

Figure 6 shows a period of a sinusoidal signal at the output of a time-interleaved ADC using the same four impairment configurations and randomly selected gain and offset error parameters. Figure 4 shows the Simulink® model that generates this data.

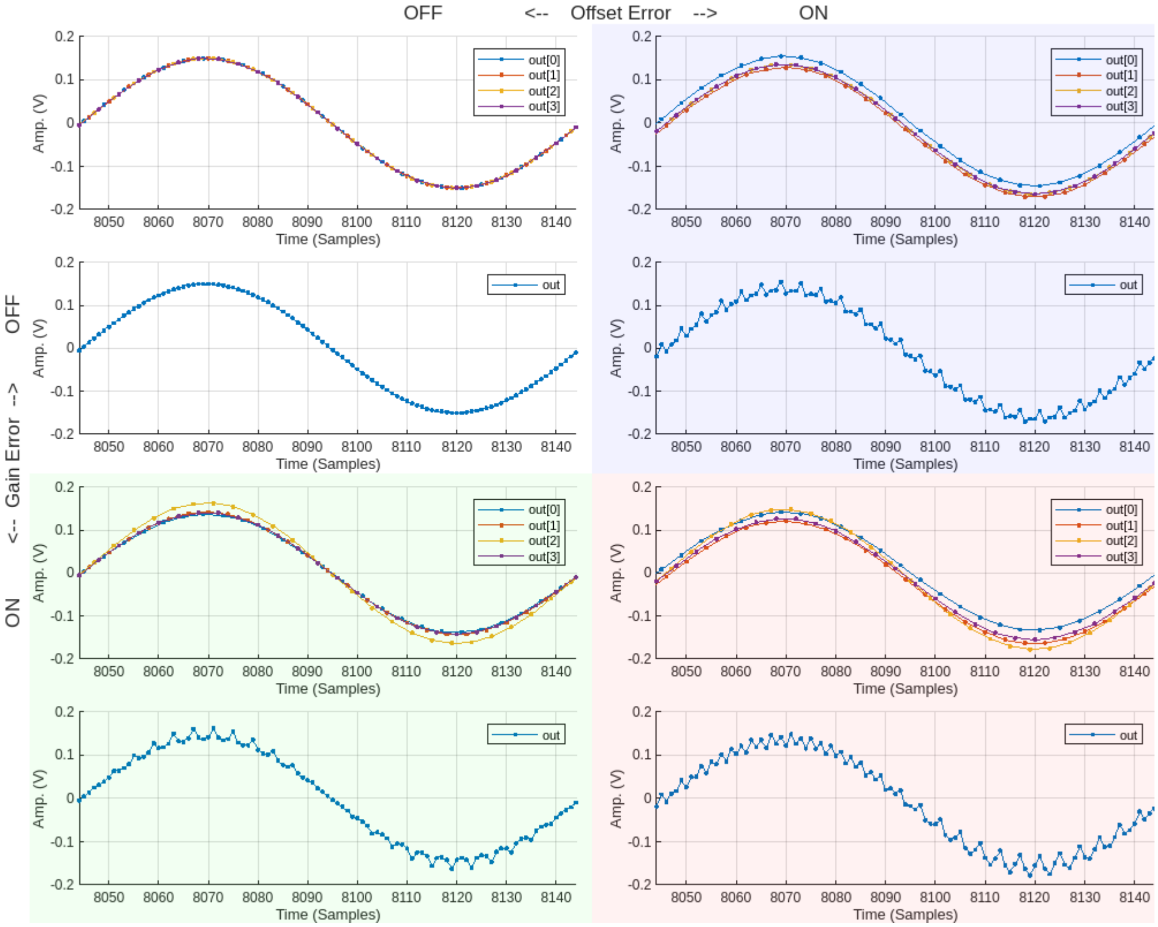

Figure 6. Comparison of impairment effects on interleaved ADC conversion results. Each quadrant has two graphs: the upper graph shows the outputs of all four ADC cores independently, and the lower one shows the result of interleaving those four cores. In all four quadrants, the input data is the same; the only difference is the presence or absence of the offset error and gain error impairments.

In Figure 6, the output waveform from each ADC, **out[0]** - **out[3]**, appears correct individually. However, when thse independent outputs are time-interleaved, they no longer result in a smooth waveform but have a distinct periodic signature: the signature differs for an offset-only mismatch compared to a gain-only mismatch. This interleaving-induced gain/offset signal deteriorates the signal-to-noise-and-distortion ratio (SNDR, as seen in Figure 7).

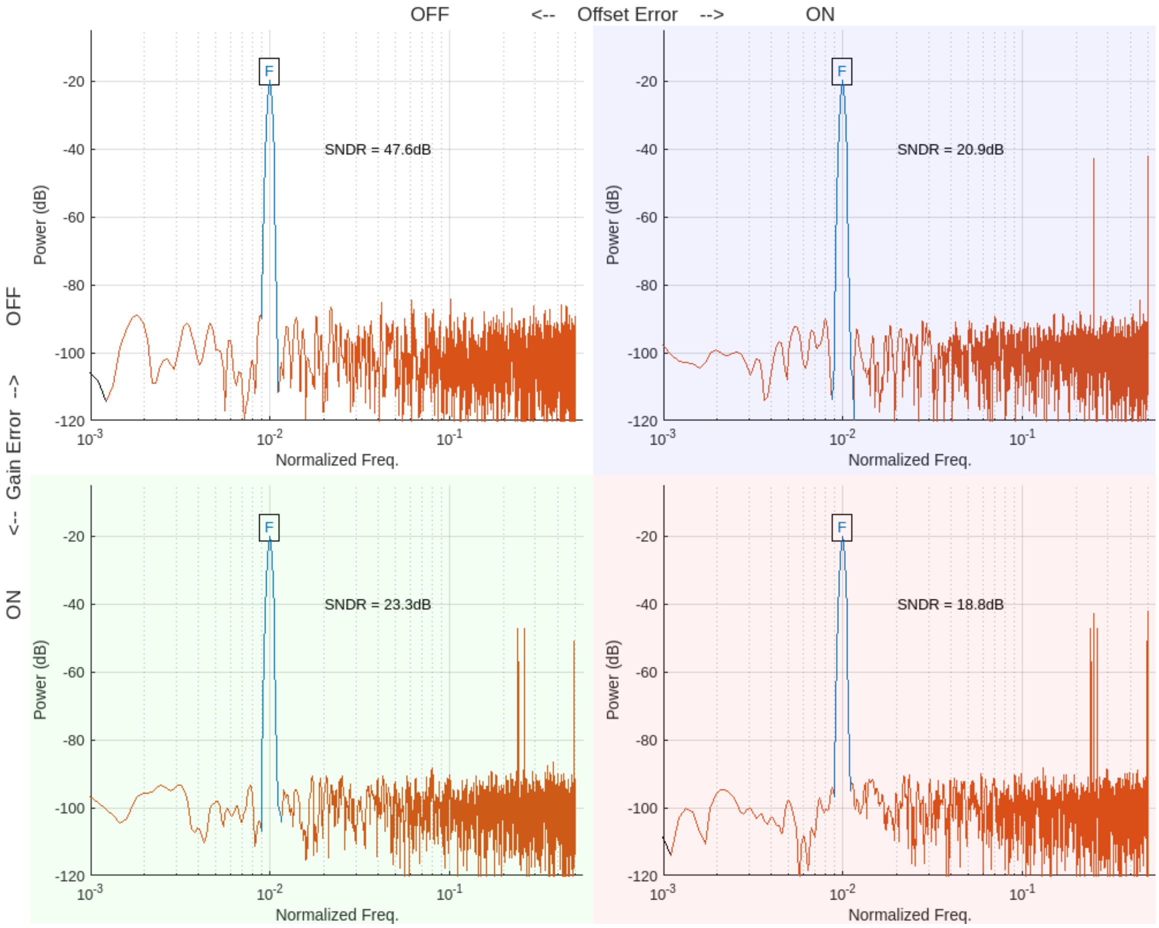

Figure 7. Comparison of impairment effects on interleaved ADC output SNDR. The signal is shown in blue, and noise/distortion in red. In all four quadrants, the input data is the same; the only difference is the presence or absence of the offset error and gain error impairments.

The spectrum for the time-interleaved output for the four different cases is plotted in Figure 7 using the aforementioned background coloring scheme. The mismatch-free case has a single tone -20 dB, corresponding to the input sinusoid, and a noise floor below -80 dB. The noise floor amplitude is directly tied to the quantization noise of the ADC, which is determined by the resolution of the ADC. The SNDR can be mapped to the ENOB of an ADC by using the following formula [10]:

$$ENOB = \frac{SNDR - 1.76\text{ dB}}{6.02\ \text{dB/b}}$$

The SNDR for the mismatch-free case is 47.6 dB, which maps to an ENOB of 7.6 b, which is shy of the expected 8 b. A significant contributor to this difference is the input signal, which is 75% of full scale (FS) or -2.5 dB of FS. Hence, the actual SNDR is closer to 50 dB (47.6 dB + 2.5 dB) and the expected 8 b ENOB.

Offset and gain mismatches cause ENOB degradation by introducing unwanted tones into the ADC output spectrum. The DC gain and offset errors are up converted by the interleaving rate. Therefore, the SNDR is 20.9 dB for the offset mismatched case, 23.3 dB for the gain mismatched case, and 18.8 dB for the gain and offset mismatched case, corresponding to ENOBs of 3.2 b, 3.6 b, and 2.8 b, respectively. Hence, the 8 b ADC performs like a 3 b ADC if the gain and offset mismatches present between the interleaved ADC paths are left uncompensated.

2 Design of the Compensation System

To eliminate mismatches, we first characterize the offset and gain errors mathematically. Then, we choose a compensation circuit architecture based on the system presented by Stepanovic in [11]. Finally, we implement the compensation system using Verilog-RTL.

2.1 Characterization of Offset and Gain Errors

One can model the offset and gain mismatch between ADC paths in a time-interleaved ADC as an input-referred offset error and an input-referred gain error, as shown in Figure 8, where **oe[0] - oe[3]** are the input-referred offset errors and **ge[0] - ge[3]** are the input-referred gain errors.

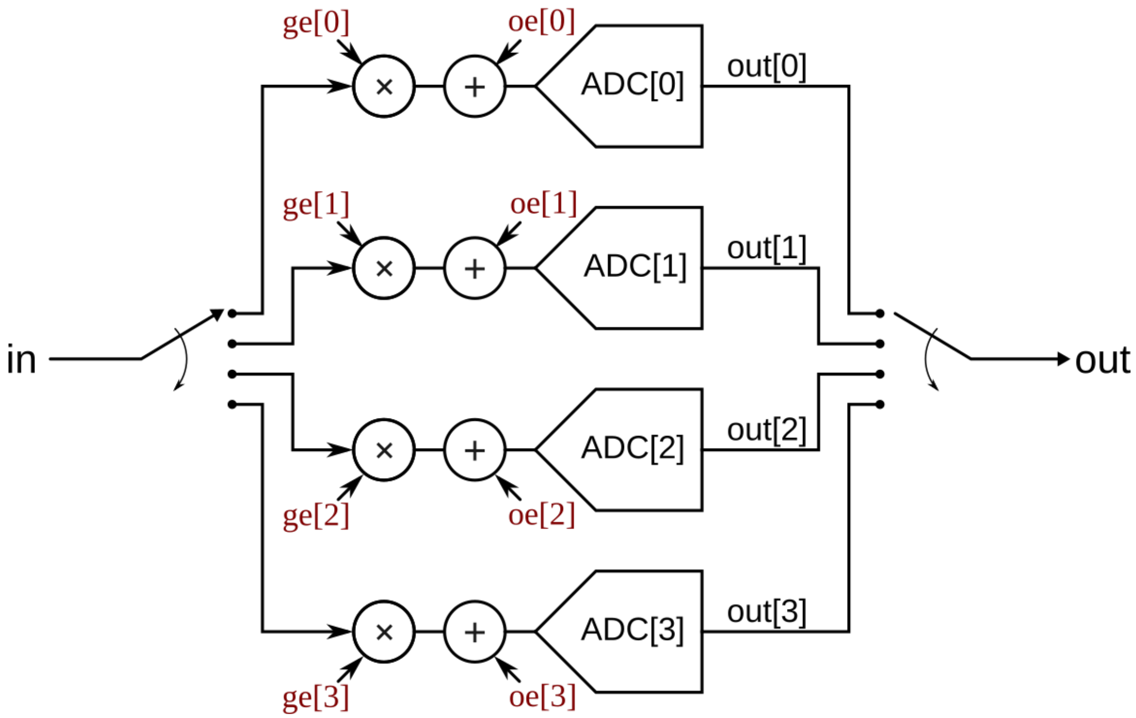

Figure 8. Interleaved ADC block diagram with impairment injection points. Gain error is multiplicative, and offset error is additive.

Sinusoidal tones are good inputs to measure the resulting signal distortion due to offset and gain mismatches. However, practical ADC inputs are not pure tones, especially for those in an ADC-based SerDes. The input signal has an amplitude distribution that can be modeled as a linear combination of normal distributions, depending on the severity of the channel impairments, such as inter-symbol interference, crosstalk, loss, and the number of signaling levels used. After compensating for offsets in components upstream from the ADC [12], the incoming signal ought to have a zero-mean distribution (0V signal). However, in the presence of path-dependent offset and gain mismatches, the code-word distributions, as seen by each ADC path, no longer have zero mean nor equal power.

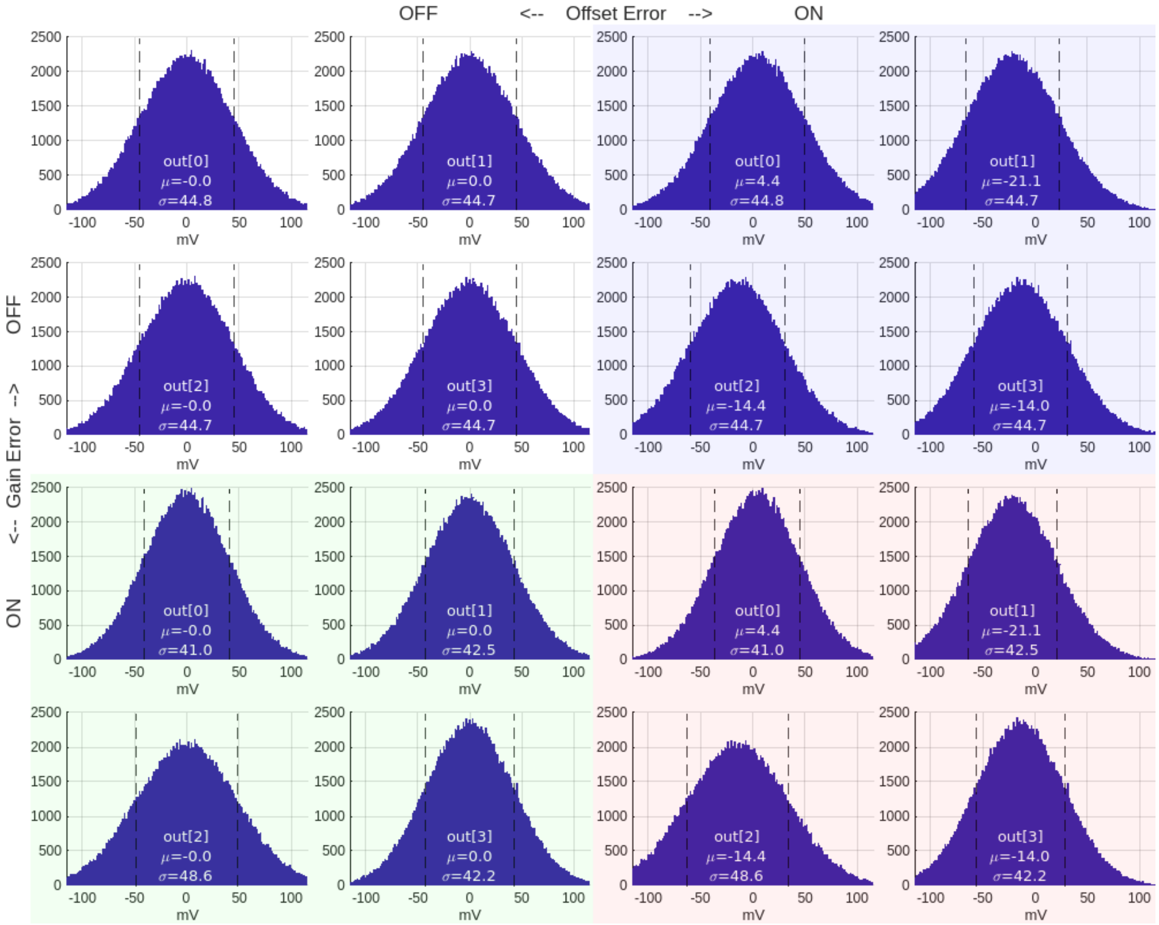

Figure 9. Histograms of individual ADC core outputs for a stimulus of a zero-mean, 44.7 mV variance normally distributed random variable. Each quadrant contains the four histograms from the individual ADC cores, and the difference between quadrants is the presence or absence of the offset error and gain error impairments.

The mean (μ) and standard deviation (σ) should be equal for the 4 ADCs (top-left four histograms) when there is no gain or offset mismatch between the ADCs because the same input is driving each ADC path. However, the means and/or standard deviations differ between the ADC paths for the other three cases, depending on whether the offsets and/or gains are mismatched. The mean (offset) and standard deviation (gain) discrepancies enable a way to quantify the path discrepancies and allow for compensation using a feedback loop.

2.2 Compensation System Architecture

Given this method of sensing multi-path gain and offset errors, we need a way to correct them: the actuator. This can be done in the analog domain before analog-to-digital conversion, as shown in Figure 10, or after conversion in the digital domain [3]. This example compensates for the gain and offset mismatches in the analog domain via path-independent controls for each ADC, as shown in Figure 10. Here, **oc[0:3]** are the offset corrections, and **gc[0:3]** are the gain correction terms that will be applied.

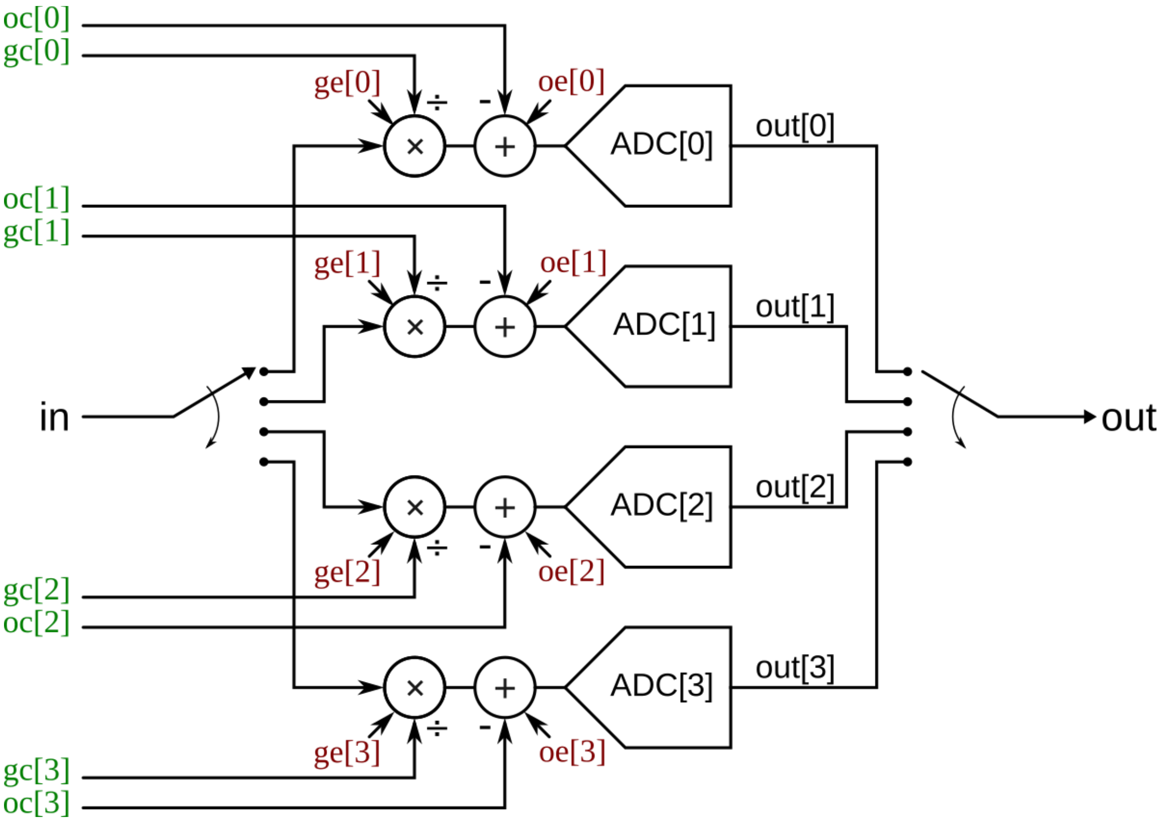

Figure 10. Interleaved ADC block diagram with impairment and compensation injection. Compensation terms are defined such that a given amount of offset or gain error is compensated by the same amount of offset or gain compensation, respectively.

Even though a practical ADC requires gain and offset calibration at power-up or periodically during mission mode, a SerDes system model may not necessarily model these mismatches because they are assumed to be calibrated or compensated before the system begins normal operation. However, a behavioral model for a time-interleaved ADC may be used to test the ability of the SerDes system to compensate for gain and offset errors. Afterall, the calibration algorithm is described as RTL and needs to be validated using a Verilog simulator[3], before and after synthesis and place and route.

2.3 Implementation of the Compensation System Using Verilog-RTL

The compensation algorithm is based on those discussed in [11]. The RTL code fragment that compensates for the offset is shown below.

```
if (offset_en)
    dc[gi] = dc[gi] + samples[gi];

offset[gi] = -dc[gi] / 2**atten;
```

When offset compensation is enabled, the signed sample value (**samples[gi]**) for each time-interleaved ADC path (**gi**) is integrated. The convergence of the offset applied to each ADC path (**offset[gi]**) is controlled by the bandwidth coefficient of the control feedback loop ($2^{-\mathbf{atten}}$). Hence, the offset is compensated by integrating the measured average sample value, until the average sample value becomes zero mean via feedback.

The RTL code fragment that compensates for the gain mismatch is shown below.

```
if (gain_en) begin
    if (samples[gi] < 0)
        pwr[gi] = pwr[gi] – (pwr[gi] + samples[gi]) / 2**atten;
    else
        pwr[gi] = pwr[gi] – (pwr[gi] - samples[gi]) / 2**atten ;

    gain[gi] = gain[gi] + (tgt - pwr[gi]) / 4096;
end
```

The gain mismatch compensation loop assumes that the power of the input signal is known. It measures the average power of the independent paths using the **pwr[gi]** signal, one for each path. The gain compensation applied to each path is the difference between the expected signal power (**tgt**) and the measured signal power according to the ADC path (**pwr[gi]**). The gain compensation converges when **pwr[gi]** equals the expected signal power (**tgt**).

Note that if the average signal power of the input signal is not known ahead of time, then it must be estimated. One method uses the average between the individual paths as the expected signal power, $\mathbf{tgt} = \frac{1}{N}\sum_{N} \mathbf{pwr}[\mathrm{gi}]$. While this approach is generally more flexible, it does have a downside: the time-interleaved ADC might not have an input-to-output gain of 1. Instead, the gain-adaptation loop converges to the average of the individual path gains.

[3] VHDL can be used instead of Verilog, without a loss of generality nor a change to the flow described.

3 Generate a SystemVerilog DPI Component of the ADC

Using HDL Verifier™, we generate the SystemVerilog-DPI component of the ADC with gain and offset errors as tunable parameters using the `slbuild` [12] MATLAB function. The goal is to generate an ADC B-model from the A-model, shown in Figure 4. This enables the ADC design and its compensation circuit to proceed in parallel. As we update the A-model to reflect the realities of the ADC circuit design, we can continue to re-generate new versions of the B-model to ensure that the compensation system remains robust to design changes. Halupka et al. discuss this process in detail in [7], including the requirements the A-model must meet and how the automatic B-model generation is accomplished. A downloadable and working example of this technique, as it applies to the ADC model shown in Figure 4, is available [13].

4 Evaluating Performance of the Compensation System

Once the ADC behavioral model has been generated, it can be used to evaluate compensation system performance. To exercise the ADC offset and gain compensation algorithm, we use the following two-step process:

1. First, the time-interleaved ADCs are stimulated using a signal generated from a Gaussian noise source. This input provides sufficient randomness to the input so that the ADC calibration engine can adapt the gain and offset of the independent time-interleaved ADC paths.
2. Second, the ADC input is switched to a tone generator so that the distortion resulting from the now-calibrated ADC paths can be measured.

The ADC gain and offset compensation control values are logged during the simulation. Time-domain traces or converged values for these are shown in Figure 11 through Figure 14 for the four different cases: mismatch free, offset mismatch, gain mismatch, and gain and offset mismatch. Figure 11 shows the convergence of the compensation system over time. Each quadrant shows two subplots, offset compensation control values on top and gain compensation control values on bottom.

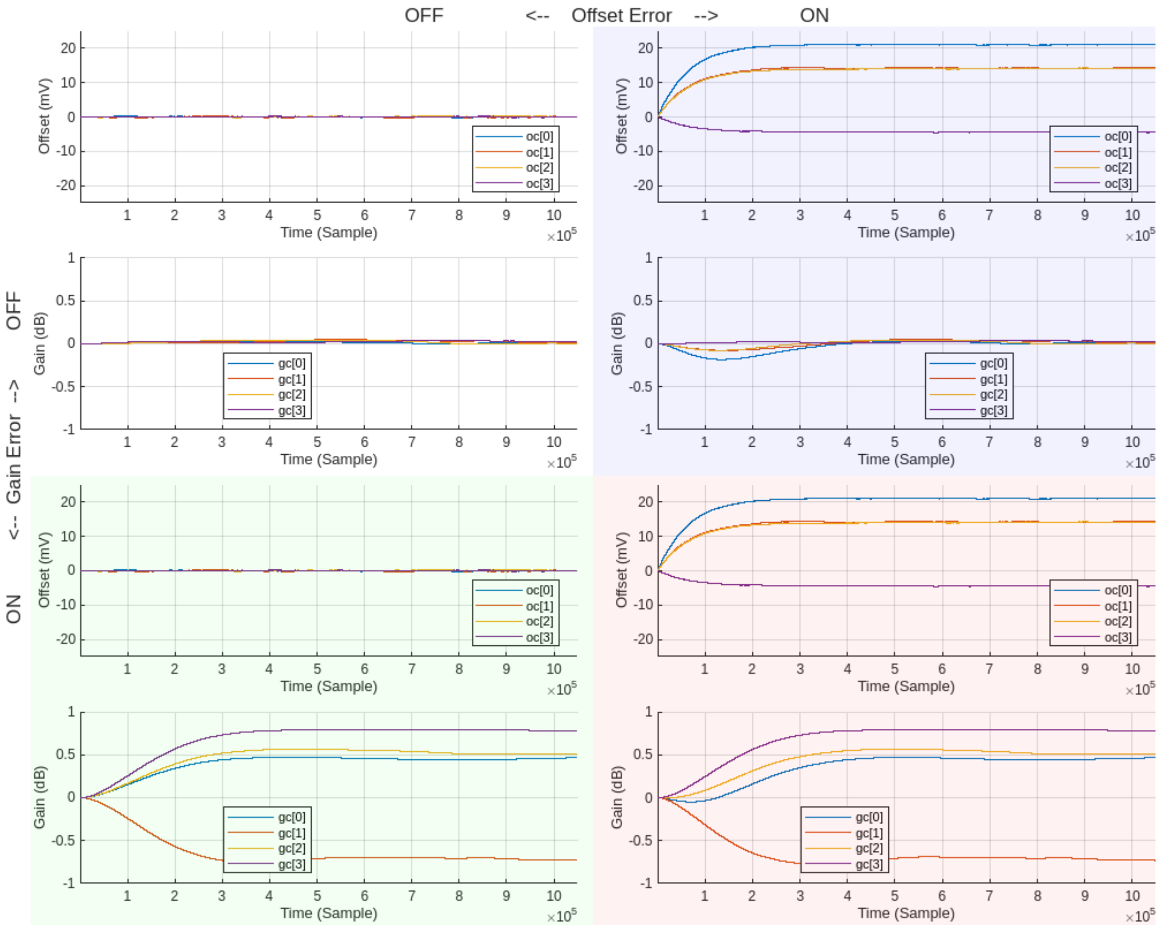

Figure 11. Offset and gain compensation system adaptation over time. Each quadrant contains two graphs, the upper plot showing the offset compensation term for each ADC core and the lower plot showing the gain compensation term for each ADC core. Again, the differences between quadrants are the presence or absence of offset and gain error impairments.

Figure 12 shows the ADC calibration convergence results, both for offset and gain calibration. The initial offset/gain error is shown using a cross marker ×, the compensation required is shown using a circle marker ○, and the residual error is shown using a square marker □. Similar to Figure 11, each quadrant contains two subplots, offset compensation results on top and gain compensation results on bottom. When the offset or gain compensation controls match their impairment, the residual error ought to be 0 mV for offset and unity (0 dB) for gain. The results presented in Figure 12 correspond to the final time step of the time-domain trace shown in Figure 11.

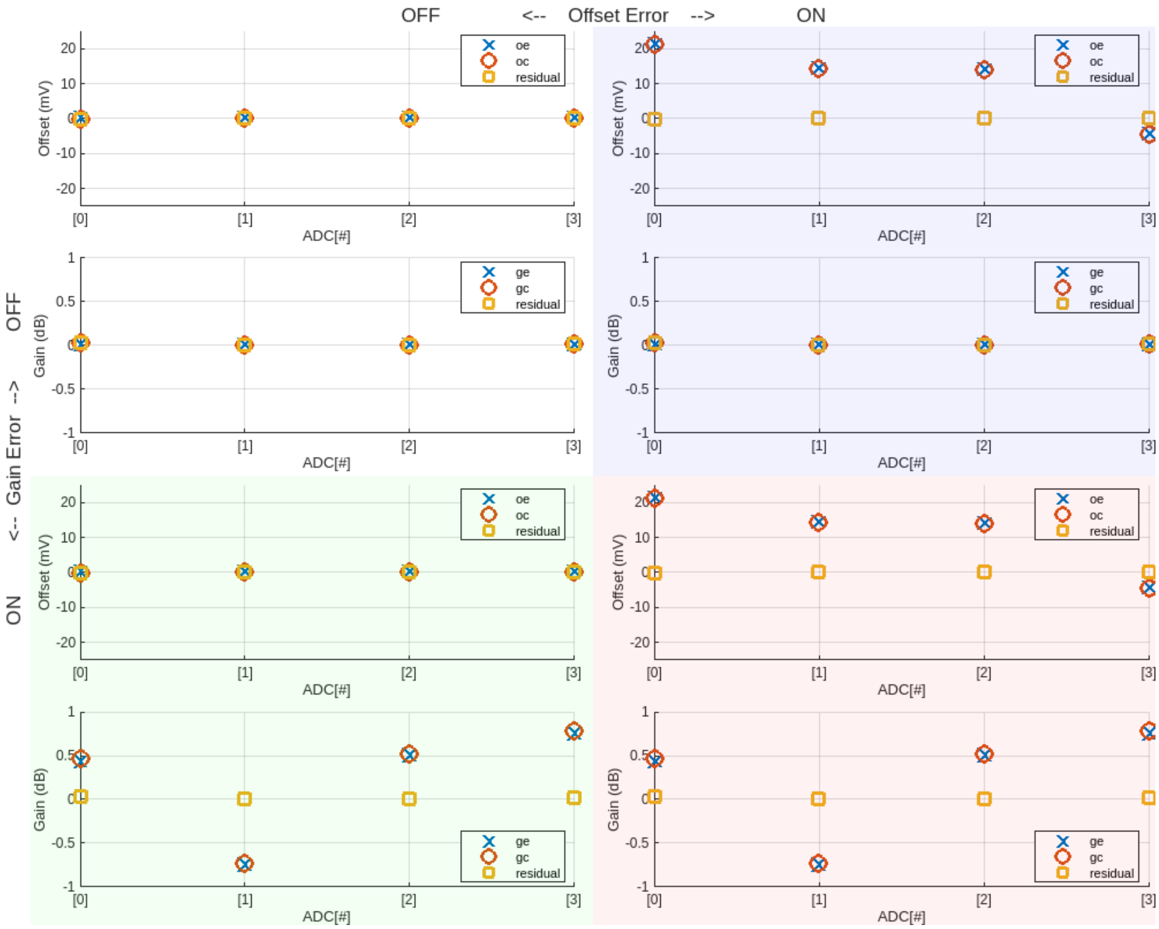

Figure 12. ADC error compensation results, at the end of the simulation, are plotted per core. In each quadrant, the upper axis shows the offset error, the applied offset compensation, and the residual offset error after compensation, while the lower axis shows the same terms for gain error and compensation.

Finally, we compare system performance using a tone waveform and its frequency spectrum, initially shown in Figure 6 and Figure 7. The corresponding post-calibration results are shown in Figure 13 and Figure 14. They show the resulting time-interleaved ADC ENOB, as measured after gain and offset calibration are completed. In the three cases where there were gain and offset errors, the ENOB is restored to the ideal case after gain/offset calibration.

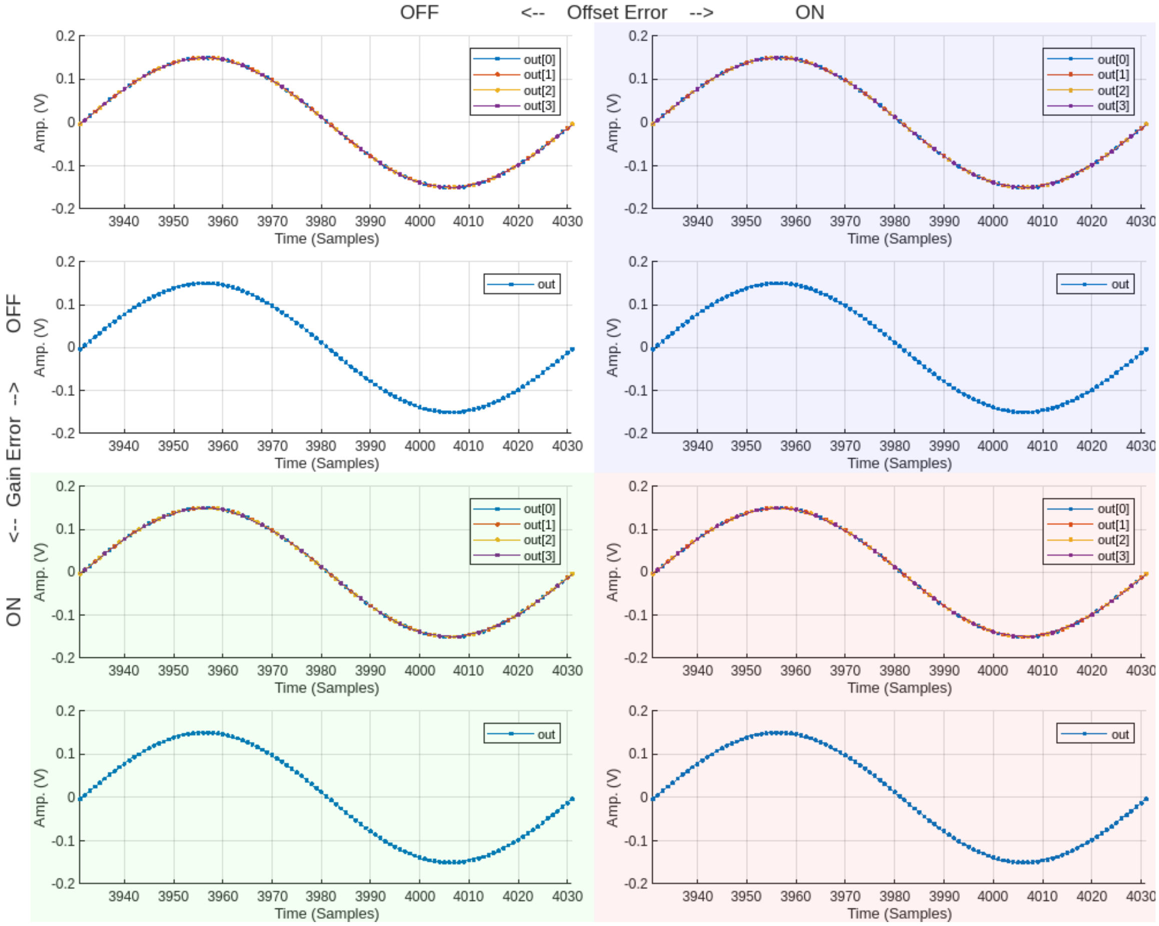

Figure 13. This figure is identical to Figure 6, except that the error compensation system is applied.

The output signal spectrums for the interleaved ADC show the SNDR improvement after gain and offset calibration is performed. In all three cases, the post-compensation SNDR is 47.2dB. This is equivalent to the offset-and-gain error free case, indicating that all interleaved paths are now in agreement.

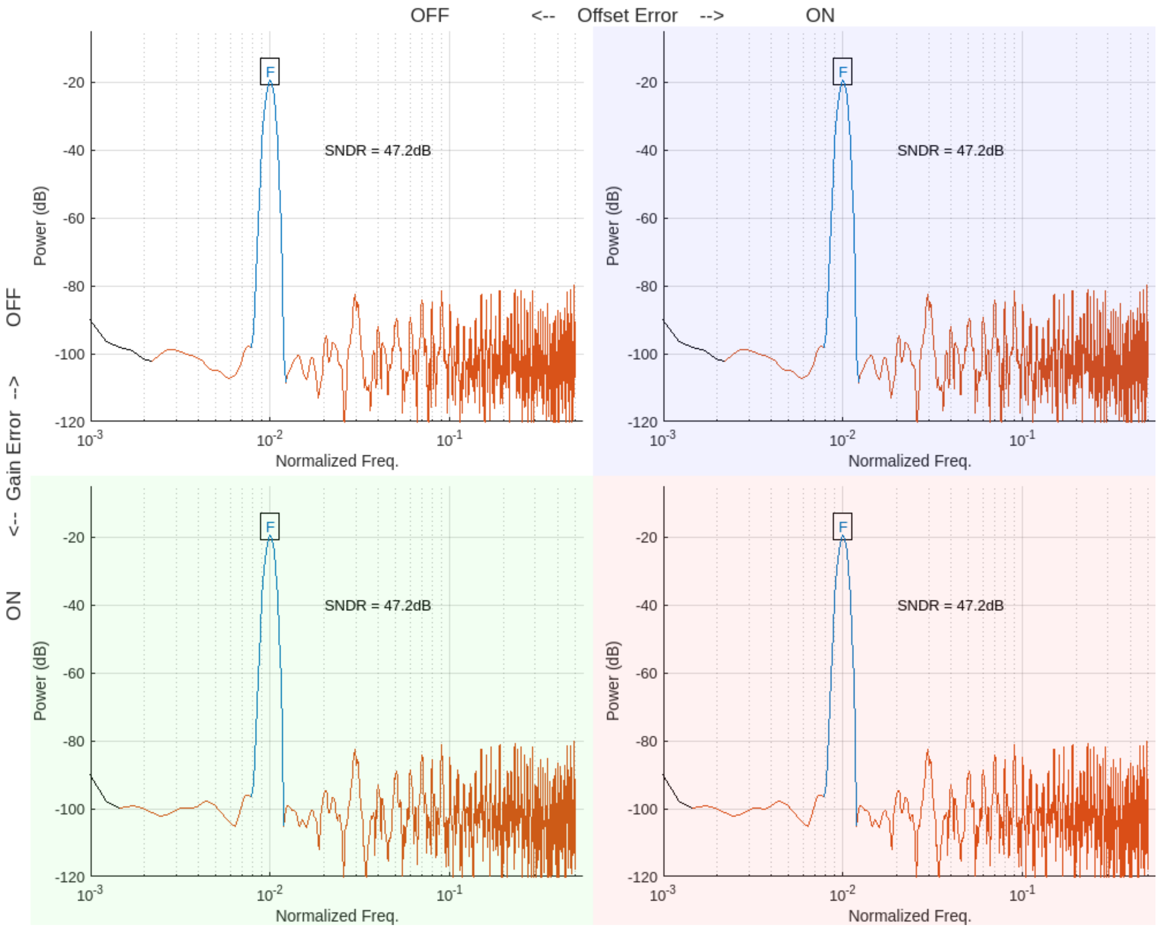

Figure 14. This figure is identical to Figure 7, except that the error compensation system is applied.

5 Conclusion

The ENOB of a time-interleaved ADC can be severely degraded due to path-dependent offset and gain mismatches. However, post-manufacturing calibration, either during power-up or during normal operation, can be used to sense and correct path offset and gain mismatches if the input signal characteristics are known; specifically, that the input signal is itself offset-free. The techniques shown also work if the input signal characteristics are unknown, but then the loops will converge towards agreement, and any upstream offset and or gain errors that are common will need to be compensated upstream. Using an offset- and gain-compensation engine, either via analog or digital feedback, significantly reduces the resulting ENOB degradation. The ability to design this compensation system independently of the ADC circuit is enabled by leveraging a behavioral model of the ADC derived from the architectural specification for the ADC itself. As the ADC design progresses and the gain and/or offset error measured during simulation is shown to be better or worse, the architectural model can be updated to match the ADC circuit, which, in turn, allows more accurate ADC behavioral models to be automatically generated and used for the design and validation of the static error compensation system. The decoupling of ADC and compensation system design allows us to left-shift not only the validation activities, but also the design activities, by leveraging up-to-date automatically-generated behavioral models.

References

[1] Y. Krupnik *et al.*, "112 Gb/s PAM4 ADC Based SERDES Receiver for Long-Reach Channels in 10nm Process," *2019 Symposium on VLSI Circuits*, Kyoto, Japan, 2019, pp. C266-C267, doi: 10.23919/VLSIC.2019.8778136.

[2] E.-H. Chen and C.-K. K. Yang, "ADC-Based Serial I/O Receivers," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 9, pp. 2248-2258, Sept. 2010, doi: 10.1109/TCSI.2010.2071431.

[3] P. Chen, "All-Digital Calibration Algorithms for the Correction of Static Non-Linearities in ADCs," M.A.Sc. Dissertation, University of Toronto, 2020.

[4] J. Luan et al. "A 56 GS/s 8 Bit Time-Interleaved ADC in 28 nm CMOS," in *Electronics*, vol. 11, no. 5, pp. 688, Feb. 2022. doi: https://doi.org/10.3390/ electronics11050688

[5] Whitcombe *et al.*, "A VTC/TDC-Assisted 4× Interleaved 3.8 GS/s 7b 6.0 mW SAR ADC With 13 GHz ERBW," in *IEEE Journal of Solid-State Circuits*, vol. 58, no. 4, pp. 972-982, April 2023, doi: 10.1109/JSSC.2022.3231783.

[6] A. Zandieh, *et al.*, "Design of a 55-nm SiGe BiCMOS 5-bit Time-Interleaved Flash ADC for 64-Gbd 16-QAM Fiberoptics Applications," in *IEEE Journal of Solid-State Circuits*, vol. 54, no. 9, pp. 2375-2387, Sept. 2019, doi: 10.1109/JSSC.2019.2917155.

[7] D. Halupka *et al.*, "Validation Shift-Left: Enabling Early SerDes Mixed-Signal Validation," *DesignCon*, 2022.

[8] "Measuring Offset and Gain Errors in ADC." MathWorks. https://www.mathworks.com/help/msblks/ug/offset-error-and-gain-error.html (accessed Oct. 20, 2023).

[9] W. Kester, Ed. "Testing Data Converters," in *The Data Conversion Handbook*, 2005 ed. Newnes, 2005, pp. 5.2-5.4, 5.28-5.31. [Online]. Available: https://www.analog.com/en/education/education-library/data-conversion-handbook.html

[10] D. Johns and K. Martin, *Analog Integrated Circuit Design*. Toronto, CA: Wiley, 1997.

[11] D. Stepanovic, "Calibration Techniques for Time-Interleaved SAR A/D Converters," Ph.D. Dissertation, University of California, Berkeley, 2012.

[12] "Verify SerDes Toolbox CTLE in Architectural, Behavioal, and Circuit Domains." MathWorks. https://www.mathworks.com/help/serdes/ug/verify-serdes-ctle-in-architectural-behavioral-circuit-domains.html (accessed Oct. 20 2023).

[13] "Verify Multi-Path ADC in Architectural, Behavioral, and Circuit Domains." MathWorks. https://www.mathworks.com/help/serdes/ug/verify-multi-path-adc-in-architectural-behavioral-and-circuit-domains.html (accessed Oct. 20 2023).