Chapter 9

Digitally assisted converters

The never-ending quest for converters with higher sampling rates, wider bandwidths, lower cost, and higher integration has progressively pushed the process technology of high speed ADCs to finer lithographies. Shorter length MOS devices have higher transconductance, smaller resistance, and lower parasitics. However, they suffer from smaller dynamic range, lower intrinsic gain, and lower output impedance. These limitations make it difficult to achieve good linearity and signal-to-noise ratio with reasonable power consumption, while pushing the sampling rates up. Digital assistance (i.e. calibration) has emerged as one of the primary tools to improve performance and reduce power consumption, especially in fine lithography processes where digital processing is more efficient. The goal has been to relax the requirements on the analog side, and correct for the resulting errors digitally.

Innovation in the field of digitally assisted converters has a long and rich history [1]. In the industry, it has evolved from factory calibration, which is used during production testing, into complex signal processing algorithms that operate continuously in the background to fix analog imperfections while adapting to a changing environment. Nevertheless, analog designers should not rejoice (or panic, depending on their perspective) yet. Employing digital assistance requires knowledge of signal processing and mixed-signal design to fully take advantage of the process technology's analog strength while circumventing its weaknesses. Successful implementation of a digitally assisted ADC involves understanding the analog imperfections and optimizing the analog circuits to be compatible with digital assistance. It has proven to be at least as challenging as traditional ADC design. So, in spite of the digital assistance, there will still be ample problems that need the creativity and resourcefulness of good analog designers.

Two of the architectures that have seen significant research in digital assistance are the pipelined and the time-interleaved ADCs. In pipelined ADCs, the MDAC is the keystone of the ADC's performance. It also determines the power consumption of the ADC because the MDAC is usually its major power consumer. Relaxing the design requirements of the amplifier has been an important area of research in digitally assisted ADCs to enable higher sampling rates, higher performance, and lower power consumption.

In time-interleaved converters, digital assistance is employed to correct the interleaving spurs caused by inter-channel mismatches. These include offset, gain,

and timing mismatches. Estimating and correcting these mismatch errors in the background is a challenging design problem.

In general, the difficulty in the background calibration of ADCs is usually not in correcting the error, but in estimating it accurately and seamlessly without disrupting normal operation. The correction can be done in the analog or digital domains, with the digital correction being easier and more efficient. One exception is the correction of timing mismatches in interleaved ADCs, where the digital correction can suffer from timing resolution and bandwidth limitations that do not exist in the analog correction.

In this chapter, we cover some of the calibration techniques used in pipelined and time-interleaved ADCs. It is important to note that many of the calibration techniques that are discussed in the context of pipelined ADC are applicable to other multi-step ADC architectures as well. Moreover, they can also be applied to MASH sigma-delta ADCs, because of their similarity to the pipeline architecture as noted in Chapter 3.

9.1 Calibration of pipelined ADC non-linearity

The pipelined ADC architecture is discussed in Chapter 7, and shown in Figure 9.1. A typical switched capacitor MDAC implementation is depicted in Figure 9.2 in a single-ended form for simplicity. The performance of the pipelined ADC's quantization is determined to a large extent by the accuracy of its MDACs, with their residue amplifier (RA) being the main design bottleneck.

As discussed in Chapter 7, the ideal residue of a stage (say the *i*th stage) can be represented as

$$V_{oi} = G_i(V_{in_i} - V_{dac_i}) = G_i\left(V_{in_i} - \frac{D_i \times V_{Ref}}{2^{k_i-1}}\right) \tag{9.1}$$

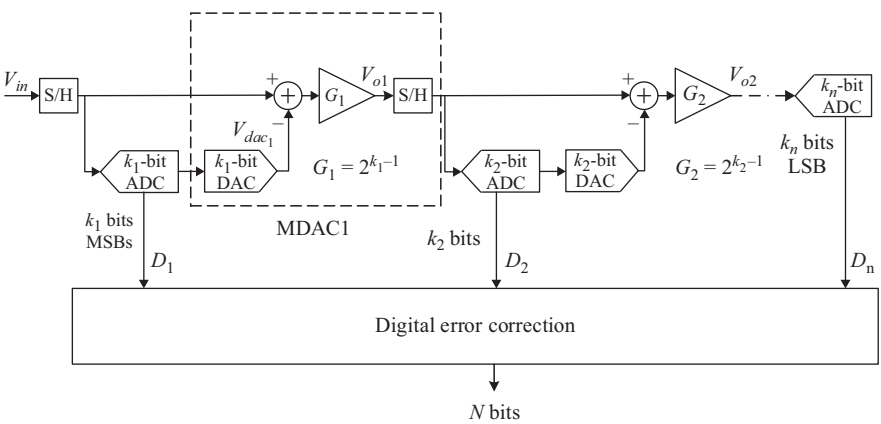


Figure 9.1 A basic pipelined ADC with redundancy

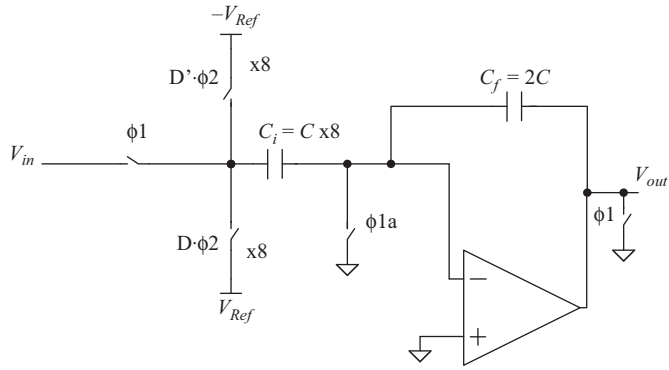


Figure 9.2 A typical switched capacitor MDAC shown in single-ended form for simplicity

where V_{in_i} is the input of the ith stage, G_i is the gain of the ith stage, and D_i is the digital code of the ith stage which is given by

$$D_i = 0, \pm 1, \pm 2, \ldots, \pm 2^{k_i - 1}, \tag{9.2}$$

V_{Ref} is the reference voltage, and k_i is the number of bits in the ith stage. In the presence of inter-stage gain error (IGE), the output can be represented as

$$V_{oi} = G_i^{'}(V_{in_i} - V_{dac_i}) = \frac{G_i^{'}}{G_i} V_{oi}\big|_{\text{ideal}} \tag{9.3}$$

where $G_i^{'}$ is the actual gain of the ith stage. Therefore, we can retrieve the ideal output value by applying the inverse of the gain to the output value, such that

$$V_{oi}\big|_{\text{ideal}} = \frac{G_i}{G_i^{'}} V_{oi} \tag{9.4}$$

Therefore, in the digital domain this is represented in terms of the digital residue $D(V_{oi})$ as follows

$$D(V_{oi})|_{\text{Cal}} = \frac{G_i}{G_i^{'}} D(V_{oi}) \tag{9.5}$$

where $D(V_{oi})|_{\text{Cal}}$ is the calibrated residue of the ith stage in the digital domain.

In the presence of DAC errors, (9.1) becomes

$$V_{oi} = G_i(V_{in_i} - V_{dac_i}^{'}) = G_i\left(V_{in_i} - \alpha_{dac_i}(D_i)\frac{D_i \times V_{Ref}}{2^{k_i - 1}}\right) \tag{9.6}$$

where $\alpha_{dac_i}(D_i)$ is the code-dependent DAC error term. The correction in the digital domain can be done using code-dependent addition of correction terms as follows

$$D(V_{oi})|_{\text{Cal}} = D(V_{oi}) + G_i\left(\alpha_{dac_i}(D_i)\frac{D_i \times V_{Ref}}{2^{k_i - 1}} - \frac{D_i \times V_{Ref}}{2^{k_i - 1}}\right) \tag{9.7}$$

Therefore, using the concepts described in (9.5) and (9.7), the quantization errors can be corrected in every stage of the pipeline. There are numerous techniques that apply these concepts to detect and correct those quantization errors. In the rest of this section, several calibration techniques of the MDAC and its RA are discussed.

9.1.1 Factory and foreground calibration

The earliest and most commonly used form of calibration is the factory calibration. In ADCs with an algorithmic nature, such as pipelined, SAR, and cyclic ADCs, some quantizer's non-linearities can be easily corrected in the digital back end. For example, in pipelined ADCs, as discussed in Chapter 7, the bits from the various stages are combined with the proper weighting as shown in Figure 9.3. The inter-stage gain applied in the digital domain must match the inverse of the inter-stage gain on the analog side with the required accuracy. If the analog gain is a power of 2, the corresponding digital gain can be implemented by simple bit shifts as shown in Figure 9.4 for a 14-bit converter with 3-bit stages.

An IGE in the analog domain results in a saw-tooth pattern in the integral non-linearity (INL) as shown in Figures 9.5 and 9.6. The INL pattern in Figure 9.5 shows an inter-stage gain that is larger than the ideal value, while Figure 9.6 depicts an INL for an inter-stage gain that is smaller than the ideal value. Either way, the IGE can be corrected by applying the gain value that matches the inverse of the analog inter-stage gain to the stage's residue in the digital back end. This was shown in (9.5) and is depicted conceptually in Figures 9.7 and 9.8. In Figure 9.7,

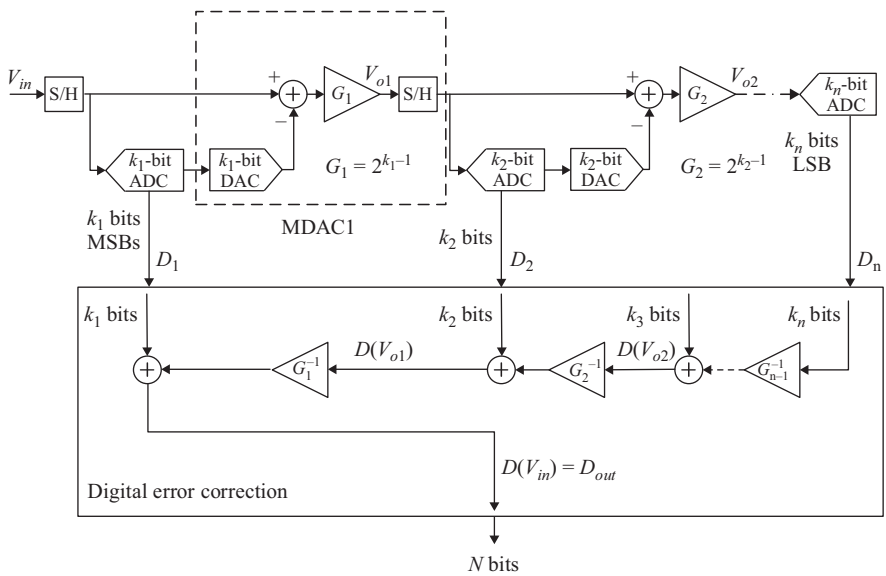


Figure 9.3 Combining the bits of the pipeline stages with redundancy and digital error correction. The gains on the digital side must match the gains on the analog side

Stage-1: $k_1 = 3$:	$d_{13}d_{12}d_{11a}$
Stage-2: $k_2 = 3$:	$d_{11b}d_{10}d_{9a}$
Stage-3: $k_3 = 3$:	$d_{9b}d_8d_{7a}$
Stage-4: $k_4 = 3$:	$d_{7b}d_6d_{5a}$
Stage-5: $k_5 = 3$:	$d_{5b}d_4d_{3a}$
Stage-6: $k_6 = 4$:	$d_{3b}d_2d_1d_0$
Output: 14-bits:	$d_{13}d_{12}d_{11}\ d_{10}\,d_9\ d_8d_7\ d_6d_5\ d_4\,d_3\ d_2d_1d_0$

Figure 9.4 Combining the bits of the pipeline stages with redundancy and digital error correction for a 14-bit converter. When the gain is a power of 2, multiplication and division in the digital domain is achieved by bit shifting

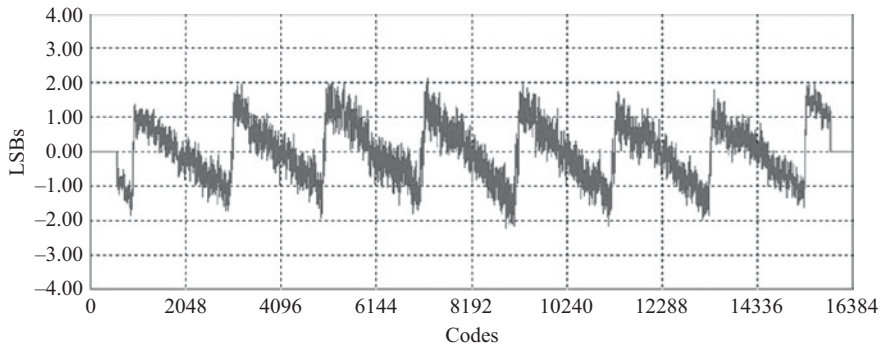


Figure 9.5 An INL showing an inter-stage gain error in stage-1, where the gain is larger than the ideal value

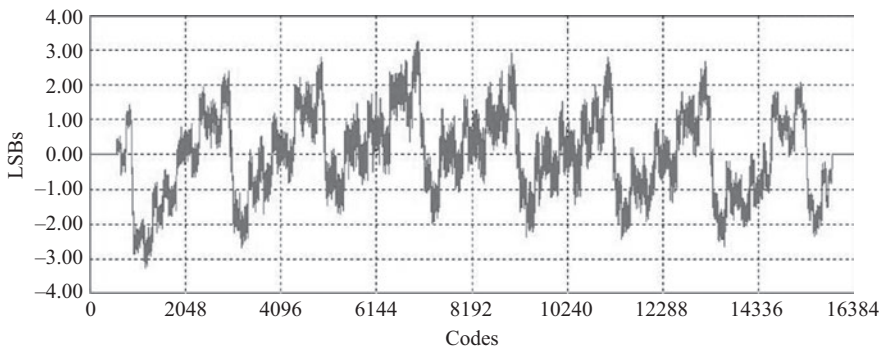


Figure 9.6 An INL showing an inter-stage gain error in stage-1, where the gain is smaller than the ideal value. We can also see inter-stage gain error in stage-2, which appears as saw-tooth pattern within each stage-1 sub-range

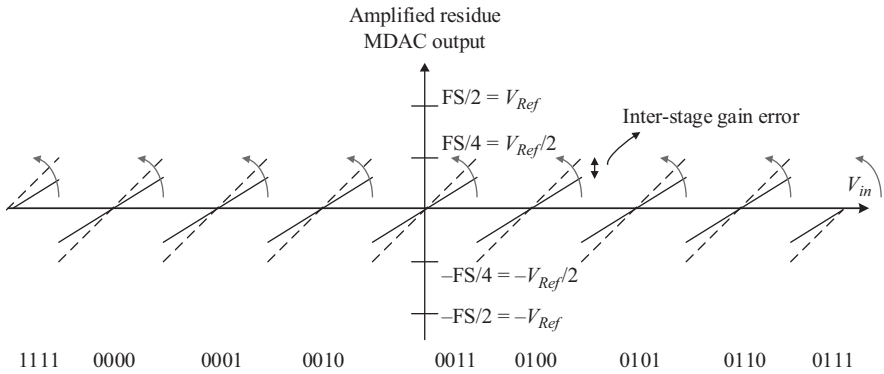


Figure 9.7 A stage-1 residue with a gain error. The arrows show the impact of a gain error fix in the digital domain

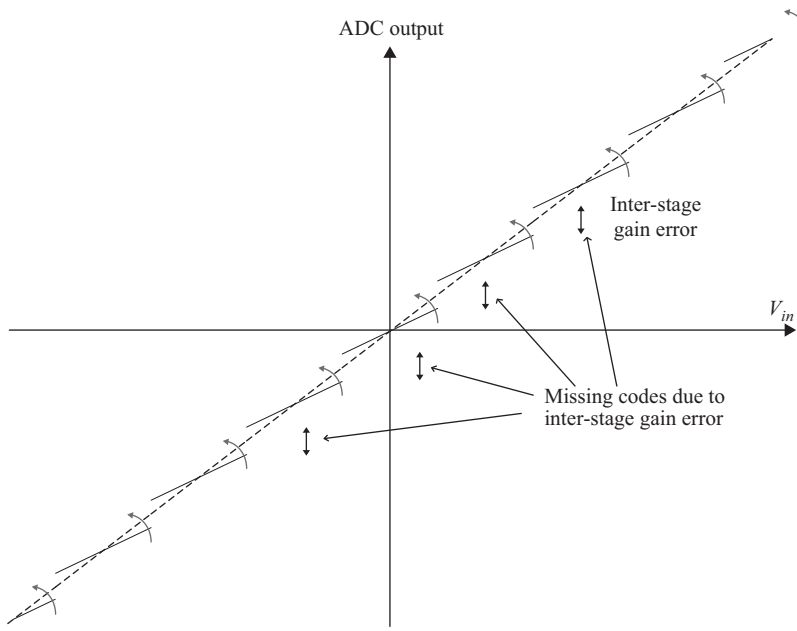


Figure 9.8 The ADC output with an inter-stage gain error. The curved arrows show the impact of a gain error fix in the digital domain

a residue with IGE is shown and the fix is done by multiplication with a gain correction factor that rotates the sub-ranges to match the correct slope. The resulting output characteristic is plotted in Figure 9.8.

Alternatively, since digital multipliers are expensive, the correction can be implemented using sub-range-dependent addition, where the sub-ranges are shifted to achieve alignment. This approach is shown in Figure 9.9 and fixes the IGEs, but

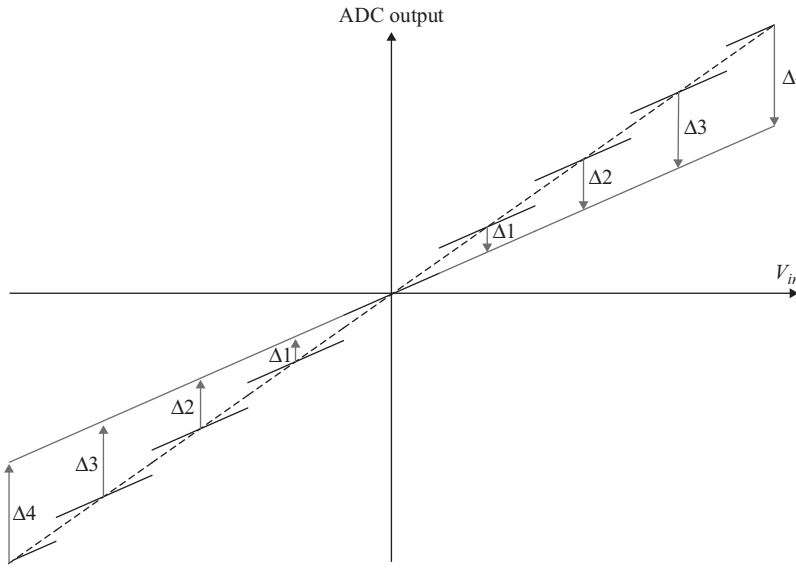


Figure 9.9 The ADC output with an inter-stage gain error. The arrows show the impact of a gain error fix in the digital domain using code-dependent additions instead of multiplications. The result is an overall gain error of the whole ADC shown as a difference in slope between the solid line and the dotted line

leads to a net overall gain error for the ADC that can be corrected using a single multiplier on the final digital word. Alternatively, the resulting ADC gain error can be corrected by adjusting the ADC reference on the analog side.

In Figures 9.5–9.7, we should note that the number of sub-ranges in the INL and residue plots is 9 sub-ranges (7 sub-ranges plus two half sub-ranges), not 8 as implied by using 3-bits in the first stage. This is due to the mid-tread implementation of the sub-ADC, and the extra comparator added to "fold" the end sub-ranges and limit of the output of the MDAC to stay within half the correction range, as mentioned in Chapter 7. The remaining unused half sub-ranges can be utilized to inject dither, as discussed later in this chapter.

In addition to IGEs, the MDAC can suffer from DAC errors due to capacitor mismatches, code-dependent settling errors, as discussed in Chapter 7. These errors cause shifts in the individual sub-ranges of the residue, as shown in Figure 9.10, and manifest themselves as shifts in some segments of the INL as shown in the example of Figure 9.11. They can be corrected by shifting the corresponding segments in the digital domain by the appropriate value. This was shown in (9.7).

Foreground calibration is often implemented in the factory during production test by observing the INL, or by forcing each comparator to switch between two different decisions to measure the corresponding "jump" in the output residue.

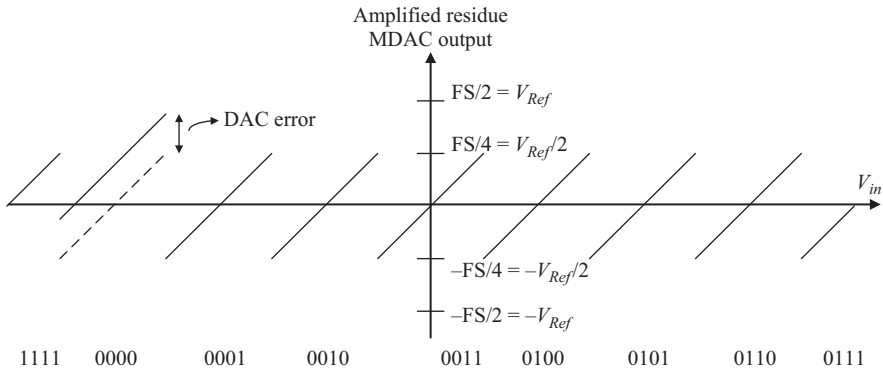


Figure 9.10 A residue showing a DAC error in stage-1. It can be fixed using code-dependent addition in the digital domain

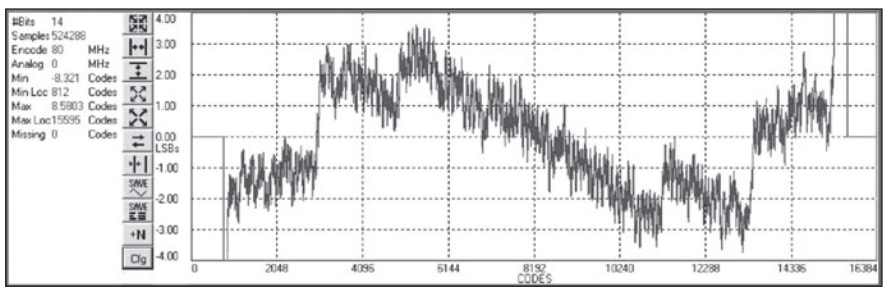


Figure 9.11 An INL showing DAC errors in stage-1

Factory calibration holds as long as the errors are constant and do not depend on temperature, supply, aging, etc. Capacitor mismatches tend to fall under this category and therefore are good candidates to fix using factory calibration, with digital coefficients that are fused permanently using non-volatile memory (NVM).

Foreground calibration can also be initiated by the user of the ADC in the form of "self-calibration." An example is described in Reference 1, where an internal calibration signal is applied, the DAC and IGE errors are measured, and the correction coefficients are applied without external intervention. These approaches require interrupting the normal operation of the ADC, which may be acceptable in some applications. They are effective in fixing constant errors, but not errors that vary with temperature, supply, aging, or sampling rate. These limitations created the need for adaptive calibration techniques that run continuously in the background to correct the errors while adapting to changes without the user's intervention or disrupting the ADC's operation. Examples of these background calibration algorithms are described in the following sections.

9.1.2 Correlation-based calibration

Correlation-based calibration is an effective technique for the background correction of IGEs in pipelined ADCs that are due to insufficient open loop gain in the amplifier, linear settling errors, and capacitance mismatches [2–5]. It can be also extended to correct for memory errors, kick-back errors, and the amplifier's nonlinearity. Moreover, it can be applied to other multi-step architectures, such as SAR ADCs, to correct for their DAC errors.

Instead of disrupting the operation of the ADC, this approach injects a pseudo-random (PN or pseudo-noise) calibration signal that is uncorrelated with the input signal. This PN signal (sometimes called dither signal) is typically added to the input signal in either the MDAC *or* the flash of the stage to be calibrated, but not both. Since it passes through the same path as the DAC signal, it encounters the same non-idealities and hence can detect the IGE. Using a statistical correlator, or the Least Mean Square (LMS) algorithm, the PN signal is "correlated out" in the digital back end, and the IGE is estimated in the process. The LMS algorithm for estimating the gain error is given as follows

$$G_e[n+1] = G_e[n] + \mu \times V_d[n] \times (V_R[n] - V_d[n] \times G_e[n]) \tag{9.8}$$

where $G_e[n]$ is the nth sample estimate of the inter-stage gain, μ is the algorithm's step size, V_d is the ideal PN (dither) digital signal, and V_R is the digital residue (output) of the stage being calibrated. The step size (μ) controls the accuracy and convergence time of the algorithm. A small μ leads to higher accuracy and longer convergence time.

Figure 9.12 depicts a pipelined ADC with correlation-based calibration of the IGE in the first two stages, where two PN signals (dither signals) are injected

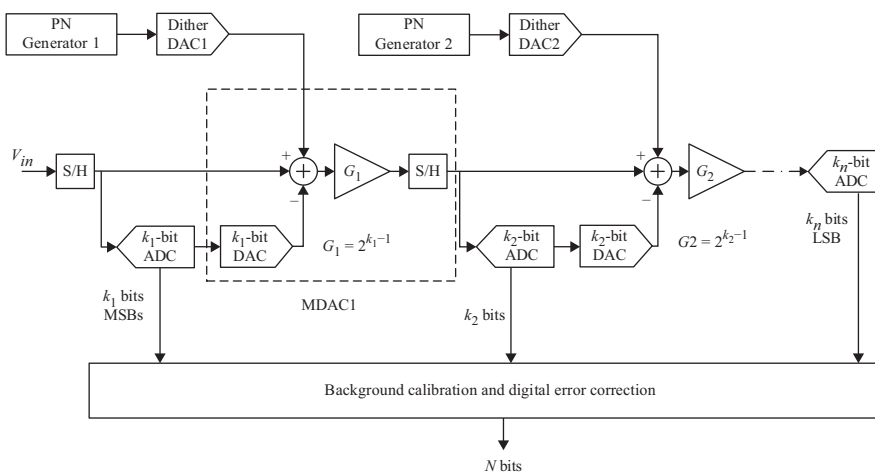


Figure 9.12 A block diagram showing the structure of a pipelined ADC with correlation-based calibration of the inter-stage gain error in the first two stages

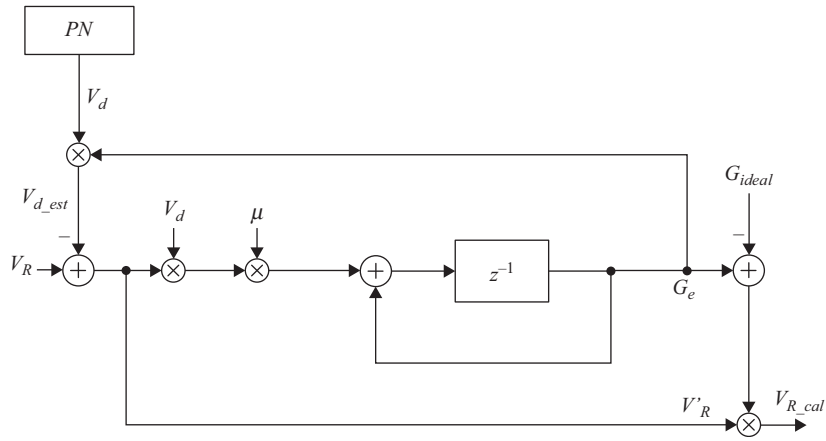


Figure 9.13 A block diagram showing the calibration and dither subtraction performed in the digital block using the LMS algorithm

the MDACs. These two dither signals must be uncorrelated with the input signal and with each other. If more stages need to be calibrated, additional uncorrelated PN signals will need to be injected in every stage that needs calibration. Although the convergence can proceed simultaneously for all the stages, the correction of a back-end stage must be applied before its bits propagate to a front-end stage.

Figure 9.13 shows the gain estimation and dither subtraction using the LMS algorithm described in (9.8). The digital residue signal V_R is obtained from the back-end pipeline with all the needed correction applied to make it as accurate a representation of the residue as possible. The dither is multiplied by the estimate of the gain and is subtracted from the residue signal. The result is multiplied by the ideal dither and by μ, then passes through an accumulator to give an estimate for the inter-stage gain G_e. The dither estimate is subtracted from the residue, which is corrected by the estimate of the gain error to give the calibrated residue V_{R_cal}. This LMS operation constitutes a feedback loop whose bandwidth is controlled by the step size μ. A large μ gives a large bandwidth, fast convergence, and low accuracy, and vice versa. Unlike feed-forward correlation approaches [2, 3], the LMS algorithm gives a smooth convergence of the gain estimate toward the final value, without the update "jumps" associated with the window-based feed-forward approaches [4, 5].

The injection of the calibration dither signal in the MDAC can be accomplished using the dither capacitances (C_{di}), as shown in Figure 9.14. The number of the dither capacitances (N_d) depends on the number of dither levels. During $\phi 1$, the dither capacitances are discharged to remove the previous charges and eliminate any memory. During $\phi 2$, they are connected to V_{Ref} or $-V_{Ref}$ depending on the

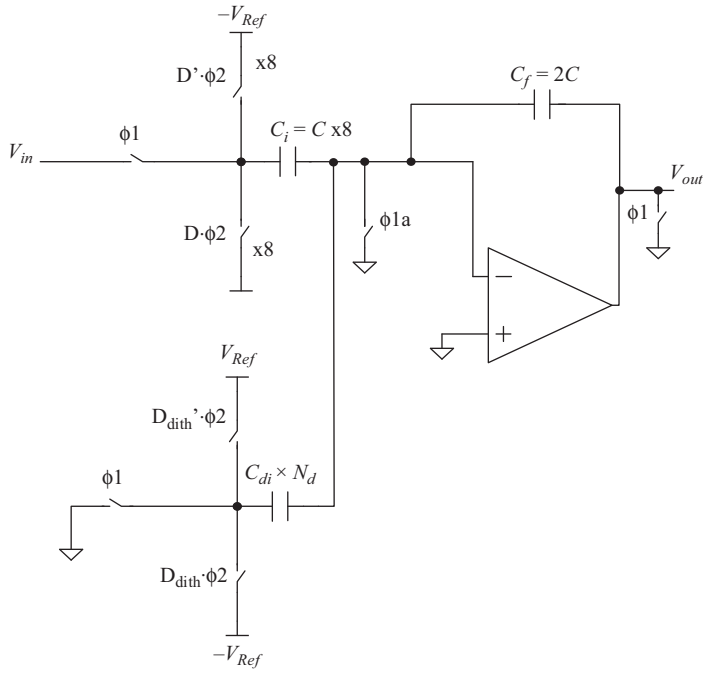


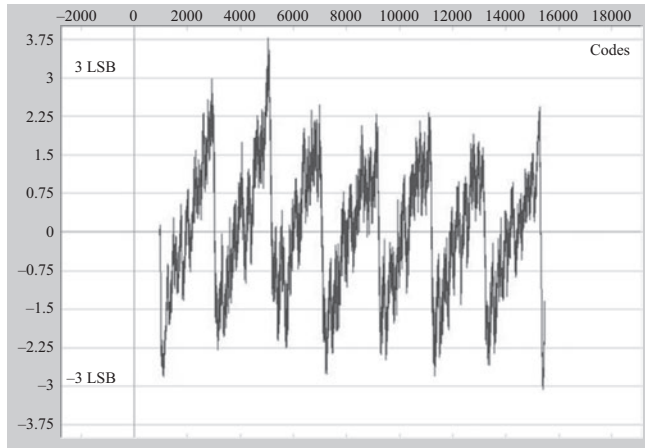
Figure 9.14 A simplified circuit diagram showing the dither injection in the MDAC. © 2014 IEEE. Reprinted, with permission, from Reference 5

randomly generated PN (dither) code. This adds the dither to the main signal as follows:

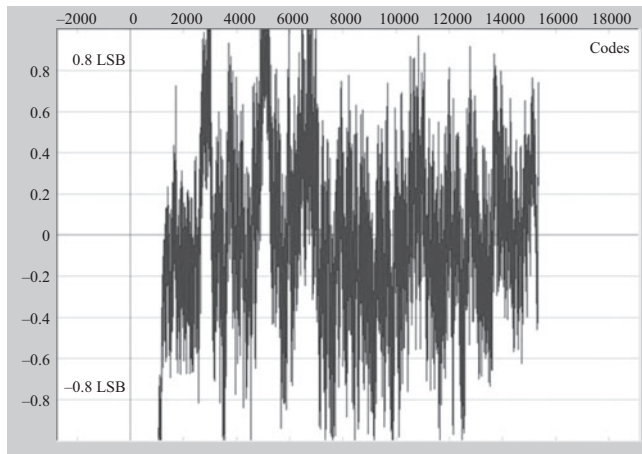
$$V_{out} = \frac{V_{in}C_t/C_f - \sum_{i=1}^{8} \mathrm{D}_i V_{Ref} C_i/C_f + \sum_{i=1}^{N_d} \mathrm{D}_{\mathrm{dith}} V_{Ref} C_{di}/C_f}{1 + \left(C_t + C_f + C_{dt} + C_p\right)/\left(C_f A\right)} \tag{9.9}$$

where $\mathrm{D}_{\mathrm{dith}}$ is the PN (dither) code (± 1), D_i is the DAC thermometer code (± 1), C_{di} is the individual dither capacitance, C_{dt} is the total dither capacitance, C_i is the individual sampling capacitance, C_p is the parasitic capacitance on the summing node, N_d is the number of dither capacitances, and C_t is the total sampling capacitance.

It is important to note that mismatches between the dither capacitances and the main DAC capacitance can result in an error in the gain estimate. The algorithm estimates the gain "seen" by the dither, and has no way of knowing how that relates to the gain that the main DAC signal sees. This mismatch needs to be measured and corrected. Since the capacitor mismatches are relatively constant, it can be corrected once and fused using factory calibration.



(a)



(b)

Figure 9.15 An INL of a 14-bit 1GS/s ADC with inter-stage gain error [5]. (a) Before calibration. (b) After calibration

An example INL before and after correlation-based calibration is shown in Figure 9.15 [5]. Before calibration, a clear IGE pattern is visible with INL jumps of about 5 LSBs. After calibration, the INL is fixed to about 1 LSB independent of supply, temperature, and sampling rate [5]. For 14-bit accuracy, a convergence time of a few seconds was needed at 1 GS/s, which translates into millions of samples. Generally, the number of samples required for convergence is proportional to the signal power that needs to be correlated out, and is inversely proportional to the dither power and the square of the allowed convergence error. That is

$$N \propto \frac{V_{\text{signal}}^2}{\varepsilon^2 V_{\text{dither}}^2} \tag{9.10}$$

where N is the number of samples needed for convergence, ε is the convergence error, V_{dither} is the amplitude of the dither signal, and V_{signal} is the amplitude of the signal.

9.1.2.1 Calibration accuracy

The calibration's accuracy and robustness are measured by evaluating the performance with changing conditions such as temperature, supply, aging, sampling rate, and so on. However, an often over-looked measure of the calibration's accuracy is its insensitivity to the input signal's frequency and amplitude. Ideally, a robust background calibration algorithm should be independent of the input signal. This is evaluated by sweeping the input signal's amplitude and frequency to confirm that the calibration is not impacted, and does not require re-convergence [5, 11].

In spite of the effectiveness of the correlation-based calibration algorithms, these techniques rely on injecting a PN calibration signal in the stage's MDAC, but not in its flash. Therefore, this PN signal consumes a portion of the correction range, which reduces the MDAC amplifier's dynamic range and indirectly increases the power consumption of the ADC. It also limits the budget available for comparator offsets and the bandwidth mismatch between the MDAC and flash of the first stage in a SHA-less architecture. To minimize this penalty, it is important to reduce the amplitude of the calibration dither signal [5, 15], which in turn degrades the accuracy and robustness of the calibration. It requires the back-end stages to have higher accuracy than what is required for the input signal. To calibrate the following stage (i.e. stage-2) to this additional accuracy requires even higher accuracy from stage-3, and so on. This represents a vicious cycle that limits the accuracy of the calibration [5].

This dilemma is shown conceptually in Figure 9.16. If the dither amplitude is reduced such that: *V_dither/V_signal* is equal to 1/8, then,

$$\delta/\Delta = \Delta/\Delta' = V\_dither/V\_signal = 1/8 \tag{9.11}$$

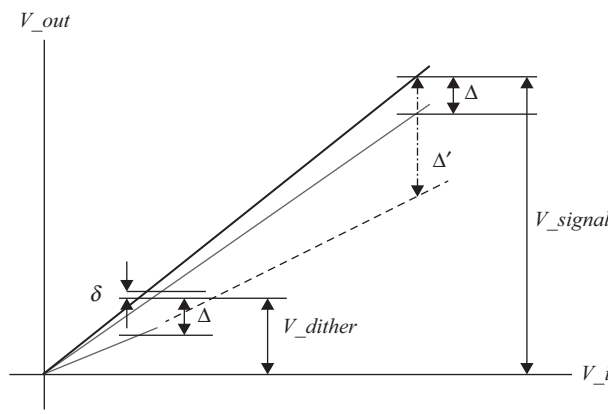


Figure 9.16 A conceptual illustration of the additional accuracy requirement if V_dither is less than V_signal

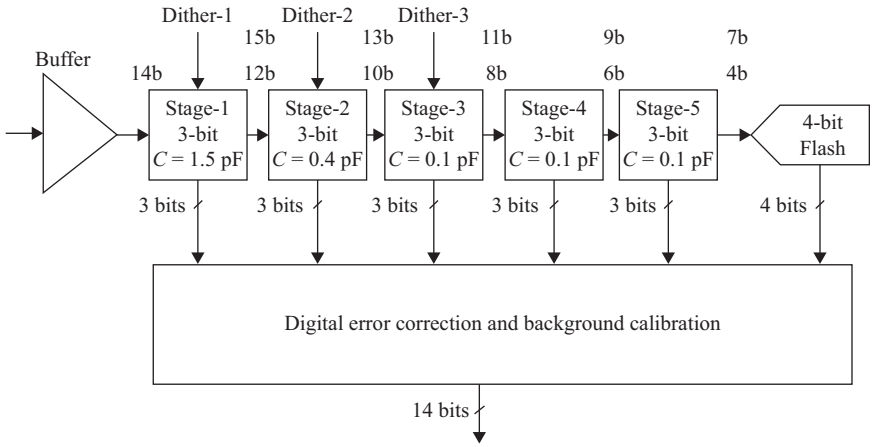


Figure 9.17 The accuracy of each stage, and the required accuracy for the dither processing (top) if V_dither is less than V_signal by a factor of 1/8. The accuracy is increased by a factor of 8 compared to the accuracy in processing the main signal, for example, from 12b to 15b as shown. © 2014 IEEE. Reprinted, with permission, from Reference 5

where Δ is the acceptable error of a full-scale signal. If the dither signal is processed with that same accuracy in spite of its small amplitude, it will result in a larger error for the full-scale signal of Δ'. Therefore, the required accuracy in processing the dither must be tightened to δ, which is scaled down relative to Δ by the ratio of the dither amplitude to the signal amplitude. For example, if the dither is 1/8th the signal, as given by (9.11), the processing of the dither needs to be more accurate by about 3 bits. This is represented on the stages of the pipeline in Figure 9.17.

It is interesting to spend a few minutes on Figure 9.17. It tells us that in order to calibrate the first stage residue to be 12-bit accurate using a dither signal that is 1/8th the full-scale, we need stage-2 to process that dither with 15-bit accuracy. If we had a stage-2 that is 15-bit accurate, we would not need to calibrate stage-1 to be 12-bit accurate in the first place! This is a clear paradox that needs to be addressed. If this additional accuracy in the back-end stages is not achieved, the estimation of the IGE may be input-dependent. An example of this problem is shown in Figure 9.18, where the IGE changes with the input amplitude by about 9 LSBs. With adequate accuracy in the back end, the change is improved to less than 3 LSBs.

Although this discussion is in the context of correlation-based calibration, the principle applies to all calibration techniques. Whenever we process a calibration signal in the digital domain, we need to ensure that the accuracy is adequate for our purpose. Assuming an "ideal" pipeline back end in analyzing calibration algorithms is a common pitfall that can end up limiting the accuracy and even the viability of the calibration algorithm.

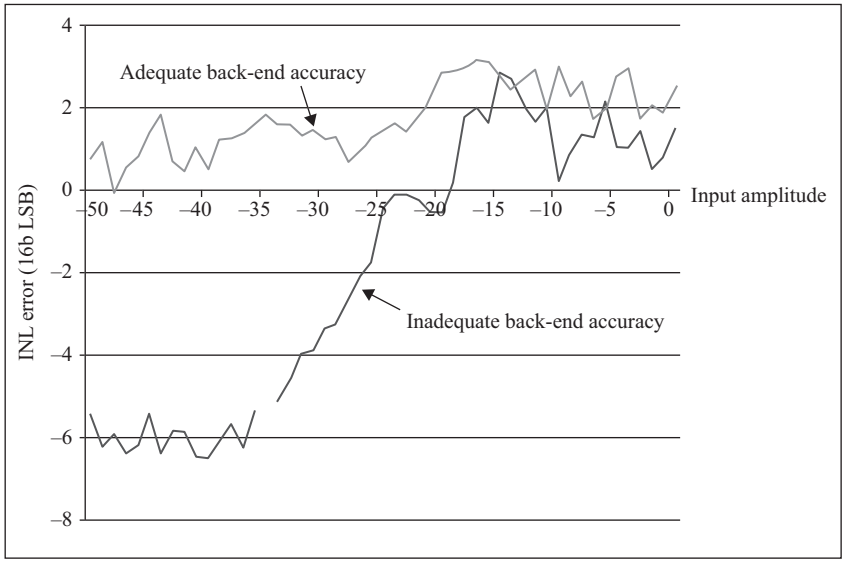


Figure 9.18 A comparison of the INL versus input amplitude for two cases: with and without adequate accuracy in the back end

One way to overcome this limitation is to employ multi-level dithering to enhance the linearity of the following stages. This multi-level dithering can be embedded in the calibration signal itself to perform both the calibration and the dithering [5]. The amplitude and number of dither levels need to be adequate to achieve the additional accuracy required. For example, if the PN calibration signal is 1/8th the input signal, at least 8 levels of dither are required to calibrate the back end. If it is 1/16th the amplitude of the input signal, we would need 16 levels of dither, and so on.

However, a binary number of dither levels leads them to fold on top of each other as they propagate down the pipeline if the inter-stage gains are powers of 2, as shown in Figure 9.19. In order to preserve the number of dither levels as they propagate down the pipeline, an odd number of equally spaced levels is preferred [5]. This arrangement, shown in Figure 9.19, preserves the number of levels for every stage down the pipeline. That is, if we use N equally spaced dither levels, where N is odd, the number of dither levels will be N for every stage down the pipeline.

An example of this implementation is discussed in Reference 5, where 9 levels of dither were used to improve the calibration effectiveness. It is important to note that the dither calibration signal is added to the MDAC only. That makes it effective in dithering the back-end stages, but not necessarily the first few stages of the pipeline. In fact, since it is injected in the MDAC only, it cannot dither its own stage. Moreover, if its amplitude is small, the gain of its stage will not be adequate

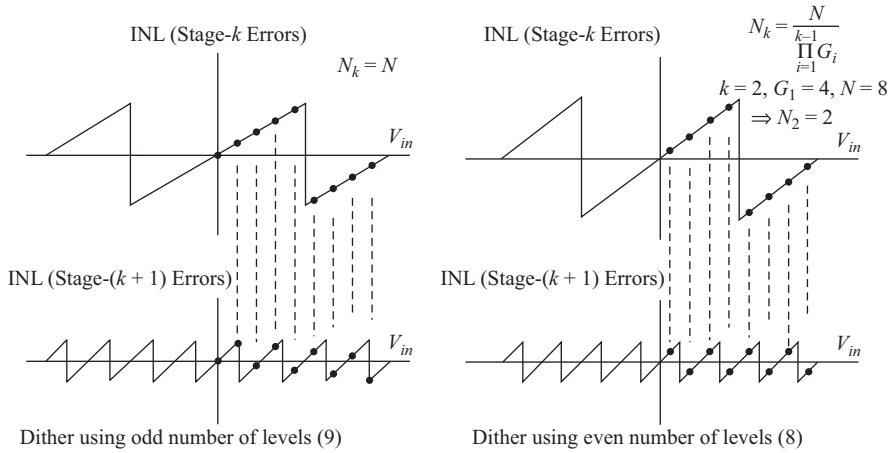


Figure 9.19 Comparison of using a binary number of dither levels versus an odd number of levels. With binary number of levels and stage gain that is also binary, the dither levels fold on each other and their number quickly dwindles as they propagate down the pipeline. With odd number of levels that are properly spaced, the dither levels stay the same and uniformly distributed as they propagate down the pipeline.

to amplify it to cover a full sub-range of the next stage. So it may not be effective in dithering the following stage either.

Therefore, it is desirable to add an additional "large" dither signal, to both the MDAC and the flash [39], to dither any residual non-linearity in the first few stages, which cannot be effectively dithered with the small calibration dither signal and to improve the calibration accuracy. It is injected in both the MDAC and the flash to effectively dither the first stage and to avoid using any portion of the correction range. In order to prevent it from consuming any portion of the ADC's dynamic range, an extra comparator is used in the flash as shown in Figure 9.20 and discussed earlier. This additional comparator frees up two halves of a sub-range at the two ends of the ADC's dynamic range, which can be occupied by the large dither without compromising the ADC's dynamic range or its linearity. Since the dither's amplitude is about half a sub-range, it will be effective in dithering the saw-tooth pattern due to the IGE. This is discussed again in more detail later in the "Dither" section of this chapter.

Correlation-based approaches represent an effective approach to calibration. They have relatively low analog overhead, and their digital processing is quite simple. They have reasonable accuracy and robustness. Their main drawback is their long convergence time. They require millions of samples to achieve 14/16-bit accuracy [3, 5]. This can result in long test times and slow response to changes. In addition, the heavy reliance on PN signals may lead to noise and jitter degradation

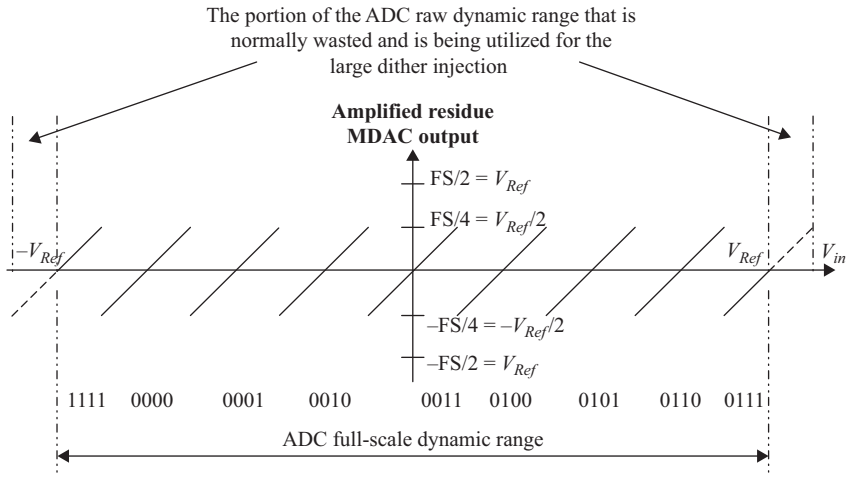


Figure 9.20 A stage-1 residue showing the portion of the dynamic range that is usually not utilized when using an additional comparator in the flash and can be consumed by the large dither signal

due to inadvertent coupling, which can degrade the ADC performance, especially at high input frequencies.

9.1.3 Summing node calibration

The summing-node method is a deterministic calibration approach that relies on measuring the error due to the amplifier's finite open loop gain directly. Examples of this approach are described in References 6 and 7. In a typical switched capacitor MDAC circuit, shown in Figure 9.21, the summing node voltage is equal to $-V_{out}/A$. Analyzing the circuit gives the following expression for the output:

$$V_{out} = \frac{V_{in}C_t/C_f - \sum_{i=1}^{8} \mathrm{D}_i V_{Ref} C_i/C_f}{1 + \left(C_t + C_f + C_p\right)/\left(C_f A\right)} \tag{9.12}$$

where D_i is the DAC code, C_i is the individual sampling/DAC capacitance, C_p is the parasitic capacitance on the summing node, and C_t is the total sampling capacitance. Assuming no capacitor mismatch or DAC errors, the output can be expressed in terms of its ideal value as

$$V_{out} = \frac{V_{out}|_{\text{ideal}}}{1 + \left(C_t + C_f + C_p\right)/\left(C_f A\right)} \tag{9.13}$$

If we substitute $K = 1/\beta$ in (9.13), where β is the feedback factor, we get

$$V_{out} = \frac{V_{out}|_{\text{ideal}}}{1 + K/A} = V_{out}|_{\text{ideal}} - V_{out}K/A \tag{9.14}$$

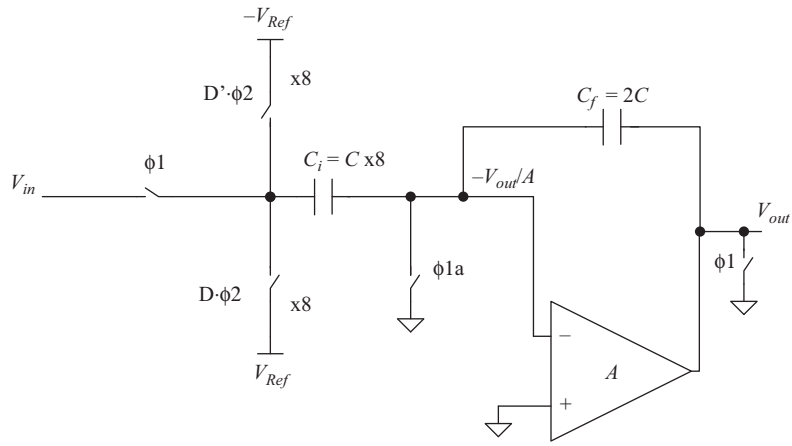


Figure 9.21 A simplified schematic of a switched capacitor MDAC shown as a single-ended circuit for simplicity

which can be rearranged to give

$$V_{out}|_{\text{ideal}} = V_{out} + V_{out}K/A \tag{9.15}$$

It is clear from (9.15) that the error due to the amplifier's finite open loop gain is equal to $V_{out}K/A$, which is proportional to the summing node voltage V_{out}/A. The error can be corrected by sampling the summing node voltage, multiplying it by the constant K, and adding it back to the signal directly in a following stage, or in the digital domain after digitizing using a "shadow pipeline," on a sample-by-sample basis [6]. This is shown conceptually in Figure 9.22.

Alternatively, the sampled summing node voltage can be amplified by gain G_s and digitized using a cheap and slow ADC. The LMS algorithm can be used to estimate the gain error for digital correction [7], as shown in Figure 9.23, which is given by

$$\alpha_{i+1} = \alpha_i - \mu \times D(V_{out1i}) \times [\alpha_i \times D(V_{out1i}) - D(V_{out1i}/A)] \tag{9.16}$$

where α_{i+1} is the estimate of the inverse of the open loop gain ($1/A$), α_i is the previous estimate, μ is the step size of the LMS algorithm, $D(V_{out1})$ is the digital representation of the residue of the stage being calibrated, which is stage-1 in this example, and $D(V_{out1}/A)$ is the digital representation of the summing node voltage. The gain correction is then applied digitally as follows:

$$D(V_{out1\_cal}) = D(V_{out1}) + D(V_{out1}) \times K \times \alpha \tag{9.17}$$

where $D(V_{out1\_cal})$ is the calibrated digital residue, and K is a constant equal to the inverse of the feedback factor.

Alternatively, the error can be processed in the digital domain and the correction done in the analog domain. The steepest descent approach is used to minimize the error, which is proportional to the sampled summing node voltage [7].

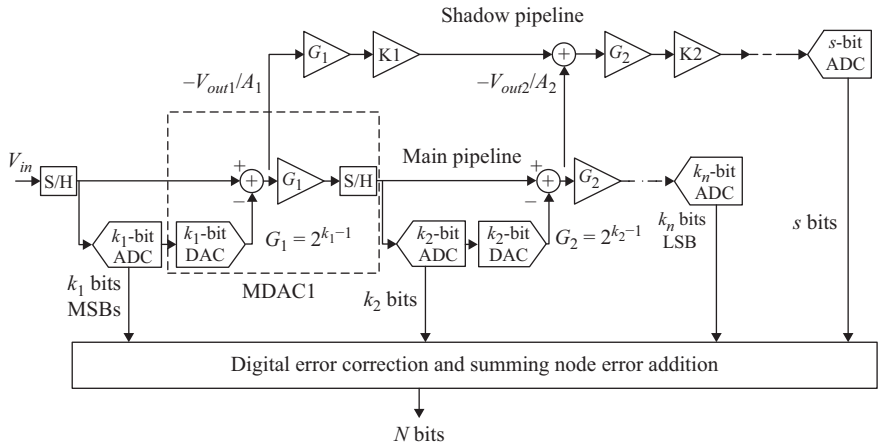


Figure 9.22 A block diagram illustrating the calibration of the amplifier gain error. The summing node voltage is sampled on the stages that are to be calibrated, processed by the shadow pipeline, and the result is added to the digital output with the right polarity on a sample-by-sample basis [6]

An example of the analog adjustment is using programmable positive feedback in the amplifier to enhance the gain. This is shown in Figure 9.24 and is given by

$$\nu_{i+1} = \nu_i - \mu \times D(V_{out1i}) \times [D(V_{out1i}/A)] \tag{9.18}$$

where ν_{i+1} is the new estimate of the analog control parameter, and ν_i is its previous estimate. Convergence occurs when the error, which is the summing node voltage $D(V_{out1}/A)$ is minimized with the desired accuracy. It is important to note that, unlike (9.16), the exact value of the error in (9.18) is irrelevant, as the algorithm attempts to minimize it by feeding back the control parameter ν to the analog domain, regardless of its absolute value.

Equation (9.18) describes an error minimization problem, instead of a system identification one. It converges much faster and requires significantly lower accuracy. However, the correction in the analog domain is possible only if there is a programmable way to control the gain of the amplifier. The gain might be dependent on the sampling rate, temperature, or supply. So the goal of the calibration is to optimize the analog gain such that it is always in its "sweet spot."

An example of an amplifier with programmable open loop gain is shown in Figures 9.25 and 9.26 [7], where positive feedback is used to enhance its gain by providing a negative transconductance that cancels the amplifier's output conductance. This increases the output impedance and the open loop gain, but exact cancellation may require adjustment to some bias voltage or device size that is programmable. On the other hand, if achieving the required gain is not possible in the analog domain, the analog correction approach will not be viable, and digital correction will be necessary.

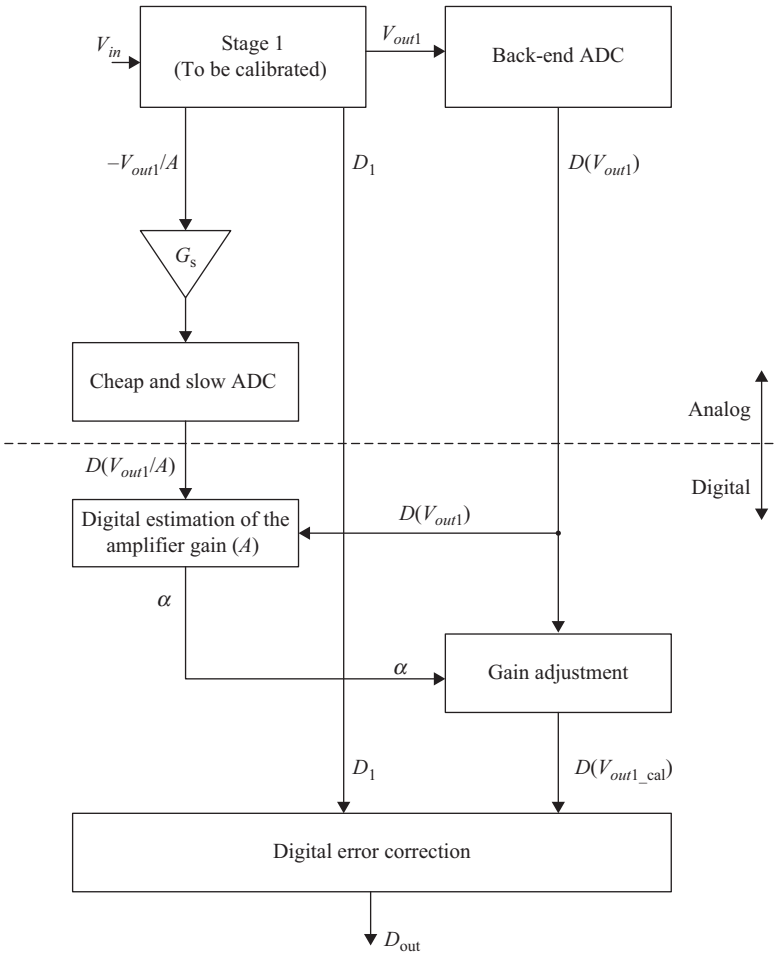


Figure 9.23 A block diagram showing the summing node algorithm with digital correction of the inter-stage gain error. © 2010 IEEE. Reprinted, with permission, from Reference 7

Figure 9.27(a) shows an implementation of the summing node sampling and digitization using a slow and cheap ADC. The timing diagram is shown in Figure 9.27(b). The summing node voltage is sampled on a small capacitor C_{e1} at a much slower rate determined by ϕ2e. Buffering can also be used to reduce the impact of the slow sampling on the summing node of the MDAC. A dummy network can be connected to the summing node in the "off" phases using the clock ϕ2eb to reduce the spurs due to the slow clock of the summing node sampler and quantizer. Once sampled, the summing node voltage is amplified and digitized at a much slower rate for the digital calibration processing.

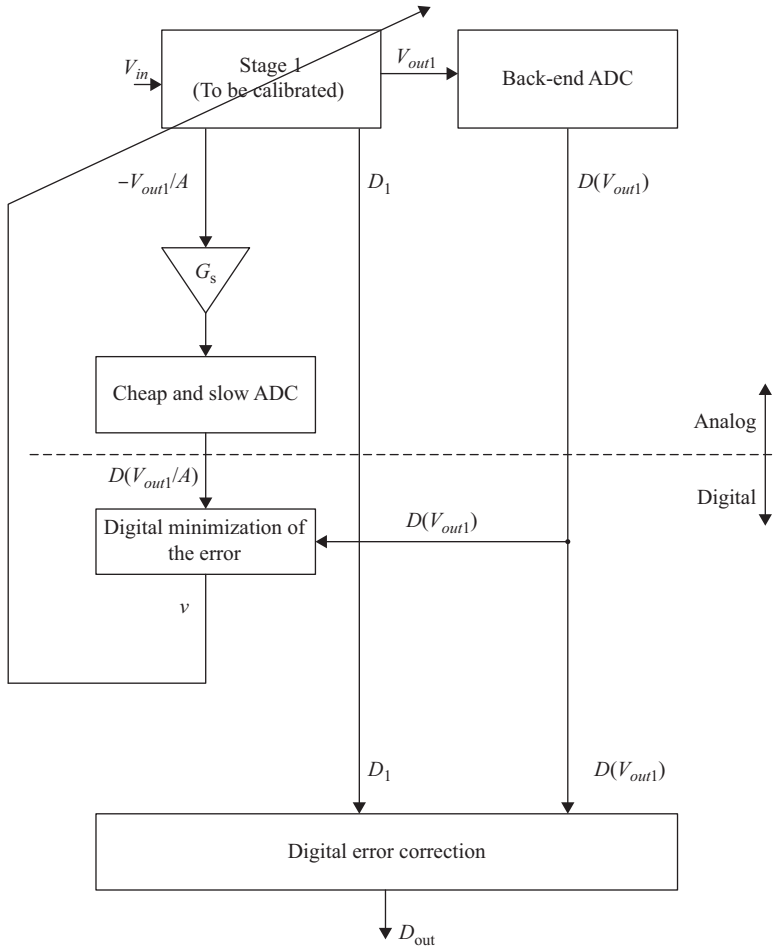


Figure 9.24 A block diagram showing the summing node algorithm with analog correction of the inter-stage gain error. © 2010 IEEE. Reprinted, with permission, from Reference 7

Unlike correlation-based and statistical techniques, the summing node technique measures the error directly in a deterministic fashion. The convergence of this algorithm is much faster and typically requires only a few thousand samples, as opposed to millions of samples for the correlation-based techniques. The convergence speed of the summing node algorithm depends on:

– The required calibration accuracy
– The noise in the calibration path (i.e. the slow ADC)
– The amplitude of the output residue

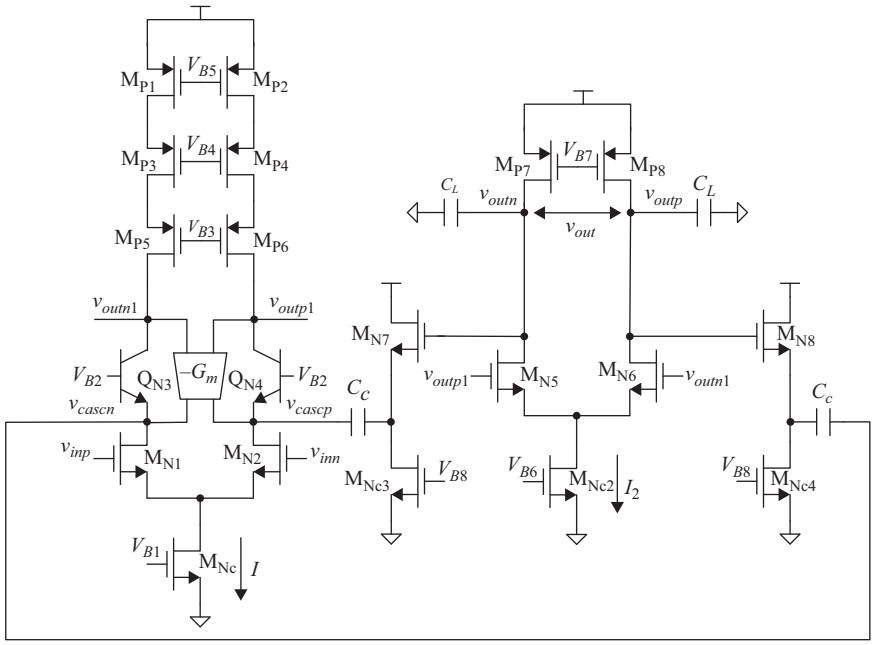


Figure 9.25 An MDAC amplifier employing positive feedback ($-G_m$ block) to enhance the gain. © 2014 IEEE. Reprinted, with permission, from Reference 7

The number of samples N needed for convergence are typically given by

$$N \approx K\left(\frac{V_{Noise\_calpath}}{V_{o1RMS}/A_{needed}}\right)^2 \approx K \times 10^{[(A_{needed\_dB} - SNR_{calpath} - V_{o\_dBFS})/10]} \tag{9.19}$$

where K is a proportionality constant, V_{o1RMS} is the RMS value of the residue signal, V_{o_dBFS} is the output residue amplitude relative to the full scale in dBFS, A_{needed} is the amplifier's open loop gain needed to achieve the desired accuracy, and $V_{Noise\_calpath}$ is the RMS noise voltage of the summing node processing path. An example is shown in Figure 9.28, where about 30,000 samples are needed for convergence with 16-bit accuracy. The noise in the calibration path in this case was about 65 dB and the required gain accuracy was about 110 dB [7]. This was used for a 250 MS/s ADC, where the slow path was operating at 12.5 MS/s. The convergence time was about 2.4 ms, which is orders of magnitude faster than the correlation-based algorithms. Equation (9.19) also shows that the absence of a signal can be problematic and would require the calibration to be frozen.

Since the algorithm measures the error directly, any error in measurement will have a limited second-order effect on the estimation's accuracy. This relaxes the requirement on the accuracy of the error processing. This is in contrast with the correlation-based and statistical approaches, where the calibration processing needs

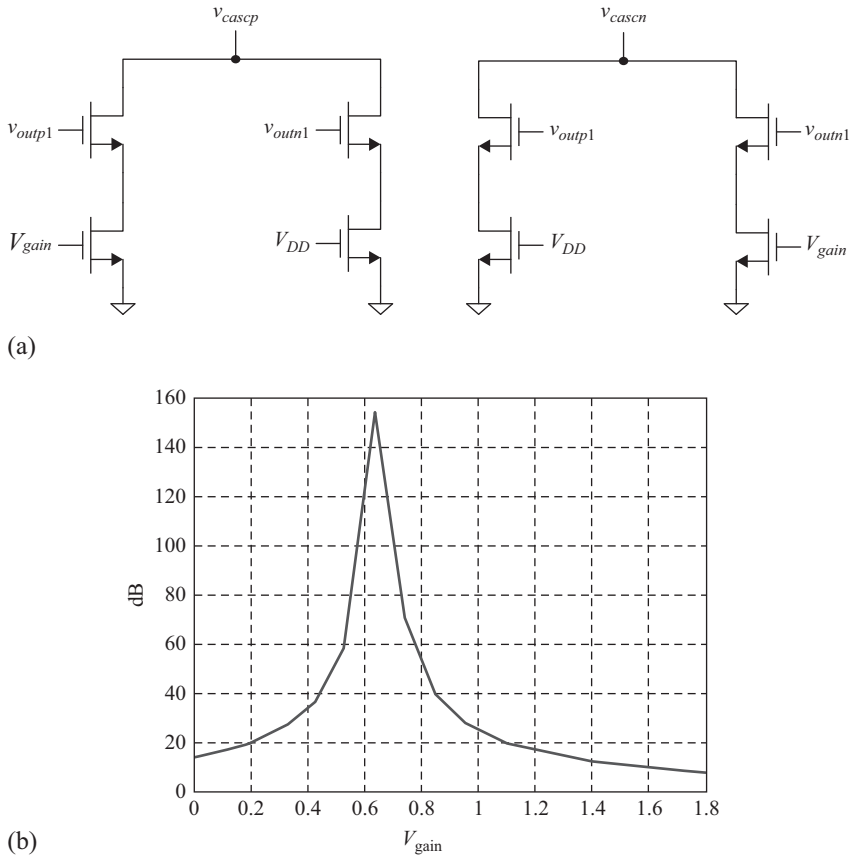


Figure 9.26 (a) An example of a $-G_m$ block in Figure 9.25. (b) The open loop gain of the amplifier as a function of the control voltage V_{gain}.
© 2014 IEEE. Reprinted, with permission, from Reference 7

to be as accurate as the overall required accuracy, and sometimes even more as described in the previous section. This relaxation in the required accuracy is demonstrated using the summing node gain correction formula, given by

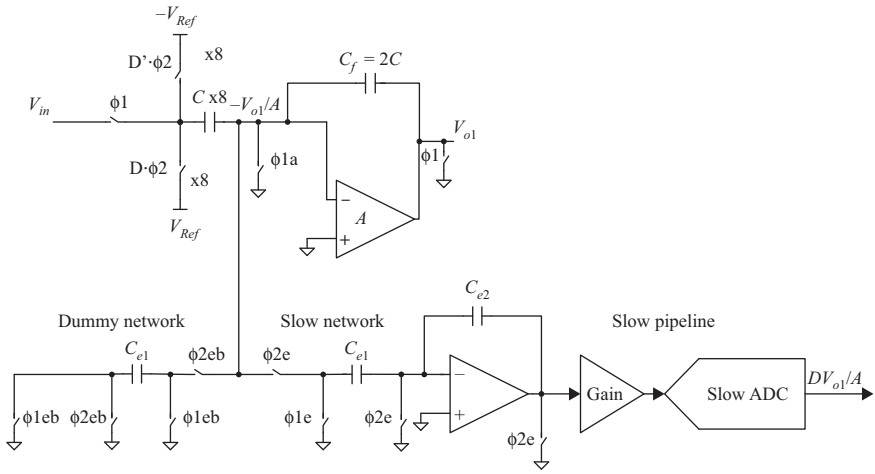
$$D(V_{out1\_cal}) = D(V_{out1}) + D(V_{out1}) \times K \times \alpha \tag{9.20}$$

For simplicity, we can represent (9.20) as

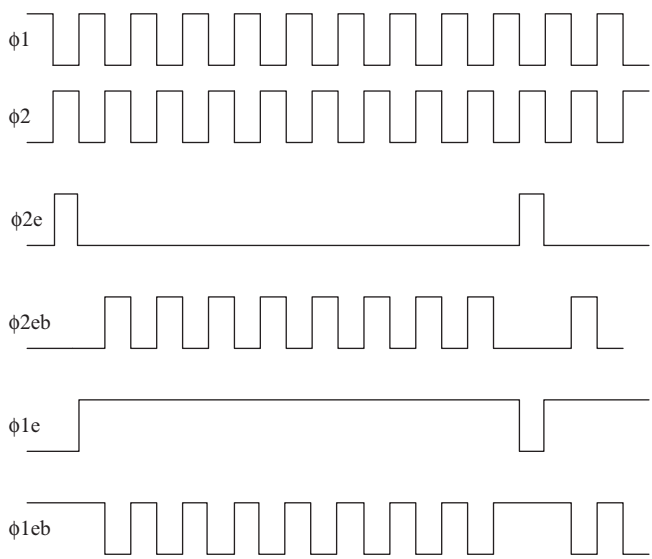
$$V_{out1\_cal} = V_{out1} + V_{out1}K/A$$

Therefore,

$$\left|\frac{\partial V_{out1\_cal}}{V_{out1\_cal}}\right| = \frac{K}{A} \times \left|\frac{\partial A}{A}\right| \tag{9.21}$$



(a)



(b)

Figure 9.27 (a) A simplified schematic showing the sampling of the summing node voltage and its processing using a slow and cheap network. The dummy network is used to match the slow sampling network in the other phases. (b) Timing diagram showing the sampling of the summing node voltage during ϕ2e and its processing using the slow pipeline during ϕ1e. A dummy network is switched in to sample the summing node during the ϕ2eb phases. © 2010 IEEE. Reprinted, with permission, from Reference 7

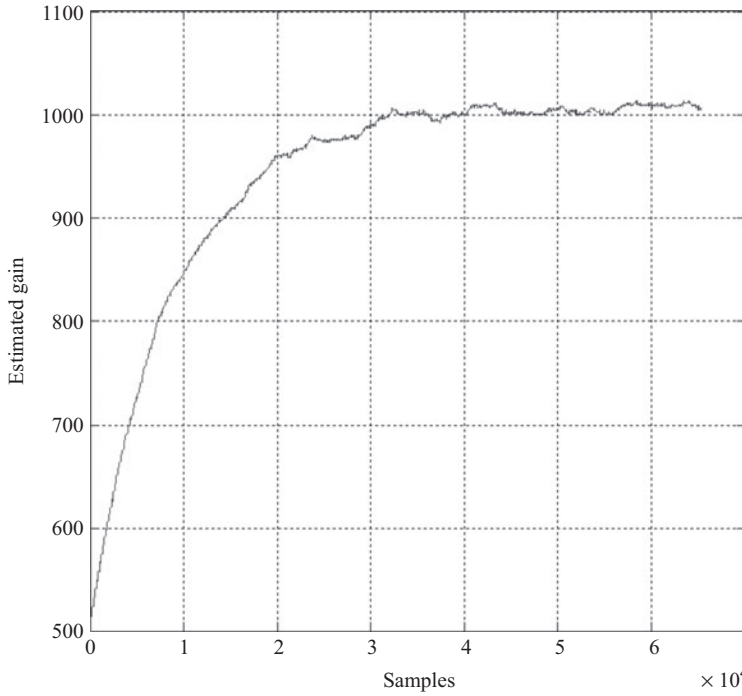


Figure 9.28 An example showing the convergence of the gain to the correct value

that is,

$$\text{The allowed calibration gain error} = \text{required stage accuracy} \times A/K \quad (9.22)$$

where A is the uncalibrated open loop gain and $1/K$ is the feedback factor. For example, if the required stage accuracy is 16 bits, the analog open loop gain A is 80 dB, and the feedback factor is 1/4, then the allowed calibration gain error is 4%, which is significantly less than the required 16-bit accuracy.

On the negative side, the summing node techniques require more analog changes than correlation-based techniques. Digitization of the error requires implementing a slow and cheap ADC. Adding the error to the following stages requires developing analog blocks that perform the error processing and addition. However, as the focus on the amplifier non-idealities increases, with an increasing need to correct the amplifier's non-linearity, deterministic approaches (such as the summing node technique) may become more attractive.

9.1.4 Reference ADC calibration

In this approach, a slow-but-accurate reference ADC is used for calibration, as shown in Figure 9.29. The difference between the output of the reference ADC and

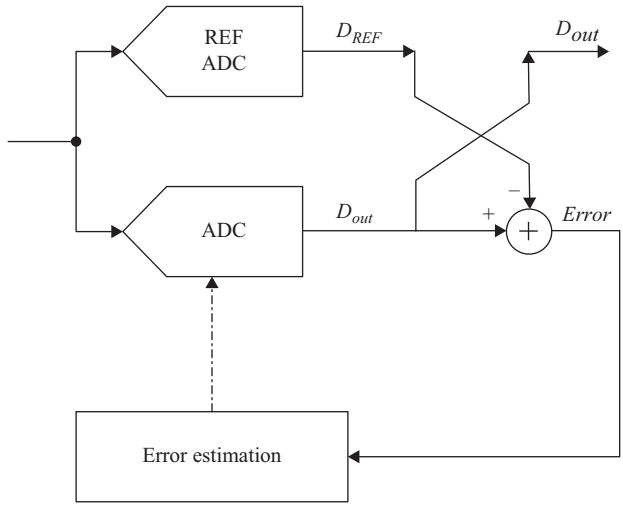


Figure 9.29 A block diagram showing the reference ADC method. The correction can be done in the digital or analog domains

the actual ADC is the error signal, and the LMS algorithm can be used to minimize the error [18, 19], such that

$$G_e[n+1] = G_e[n] - \mu \times V_R[n] \times (V_R[n] \times G_e[n] - V_{REF}[n]) \tag{9.23}$$

where V_R and V_{REF} are the digital representations of the residue voltages of the main and the reference ADCs, respectively, and G_e is the estimate of the inter-stage gain. This can also be represented as follows:

$$G_e[n+1] = G_e[n] - \mu \times D_{REF}[n] \times (D_{out}[n] - D_{REF}[n]) \tag{9.24}$$

where D_{out} and D_{REF} are the outputs of the main and reference ADCs, respectively.

This technique falls under the umbrella of the deterministic approaches. Similar to the summing node calibration techniques, this method has fast convergence time and suffers from significant analog design overhead to develop the reference ADC. Unlike the summing node method, the slow reference ADC needs to be accurate. Its noise can be high, but its linearity must match the required accuracy of the calibrated ADC.

The reference ADC method suffers from limitations due to possible mismatches between the main path and the reference path. Some of these mismatches, such as clock skew and kick-back, can limit the calibration accuracy. In addition, the reference ADC can degrade the performance of the main ADC due to its slow clock, and its loading effect on the input.

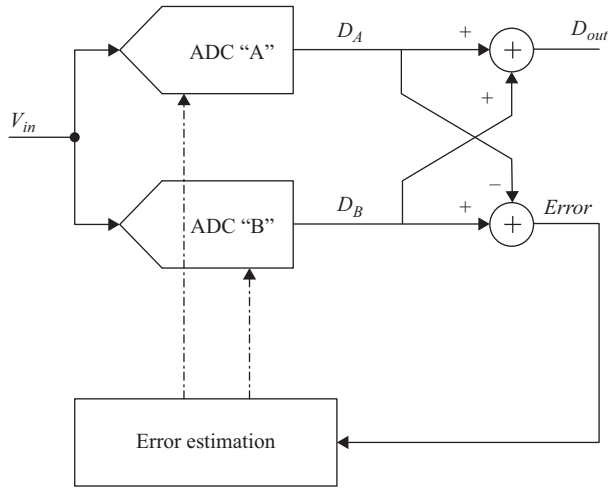


Figure 9.30 A block diagram showing the Split ADC method. The correction can be done in the digital or analog domains. © 2009 IEEE. Reprinted, with permission, from Reference 30

9.1.5 Split-ADC calibration

This technique is similar to the reference ADC method. However, instead of using a slow-but-accurate ADC, the main ADC is divided into two identical halves. The sum (or average) of the two outputs is the overall ADC output. The difference between the two outputs represents the error signal that needs to be minimized by the LMS algorithm [20]. This is shown conceptually in Figure 9.30. In order to ensure that the difference is nulled only when the two ADCs are correctly calibrated, and not just when they are identically wrong, the two halves operate in different residue modes that are randomly selected by forcing the comparators of the sub-ADCs to make different decisions for the same input. The different decision paths taken by the two halves make it unlikely for the error to be minimized when both are equally wrong.

Like the reference ADC approach, this technique is deterministic, has fast convergence time, and requires significant changes on the analog side. In addition, it is sensitive to the offset, gain, and timing mismatches between the two halves, which can lead to calibration errors.

9.1.6 Settling error calibration

An important goal of the digital assistance in high speed pipelined ADCs is to correct the dynamic settling errors, and hence enable higher sampling rates, better accuracy, and lower power consumption compared to what is possible using analog circuit techniques alone. The MDAC amplifier's settling is typically divided into large-signal settling and small-signal settling [21]. The large-signal settling is

limited by the slew rate, the DAC reference's large-signal settling, and to some extent the common-mode settling of the amplifier. It is highly non-linear, and results in non-linear distortion in addition to the gain error. The non-linear charge injection can limit the settling linearity as well. On the other hand, small-signal settling is linear and results in IGEs. Therefore, it can be calibrated using the IGE calibration techniques described in the previous sections.

Without loss of generality, if we assume a single pole system, the small-signal settling will be given by

$$v_o = v_{initial} e^{-t_s/\tau} + v_{final}(1 - e^{-t_s/\tau}) \tag{9.25}$$

where t_s is the setting time and τ is the settling time constant, which is given by

$$\tau = \frac{1}{2\pi \times BW_{cl}} = \frac{1}{\beta \omega_u} \tag{9.26}$$

and BW_{cl} is the closed loop bandwidth of the amplifier, β is the feedback factor, and ω_u is the unity gain angular frequency. If the amplifier is reset and starts from zero, the output after a settling time t_s will be given by

$$v_o = v_{final}(1 - e^{-t_s/\tau}) = \left(\frac{V_{in} C_t / C_f - \sum_{i=1}^{8} \mathrm{D}_i V_{Refi} C_i / C_f}{1 + 1/\beta A} \right)(1 - e^{-t_s/\tau}) \tag{9.27}$$

If t_s and τ are constant, the output will have a *linear* settling gain error $\delta G/G$ that is given by

$$\delta G/G = e^{-t_s/\tau} \tag{9.28}$$

In the presence of more than one pole, the small-signal settling behavior may be over-damped, under-damped, or critically damped. However, the small-signal settling error will still be linear and cause a gain error, as long as it is within the range of the small-signal settling. In practice, since the large-signal settling time is variable, the initial voltage term $v_{initial} e^{-t_s/\tau}$ in (9.25) cannot be ignored and will degrade the settling error linearity. Moreover, the large-signal settling is highly non-linear, which complicates the calibration further.

Figure 9.31 shows the large- and small-signal settling for two cases A and B. It is clear that in Output A, the large-signal settling is about 50 ps longer than that for Output B. In the former case, the large-signal settling was limited by longer DAC and reference settling time in addition to the slew rate. Up to points 5 and 6, the settling is highly non-linear. The non-linearity decreases substantially until points 2 and 4, where the settling becomes almost entirely linear within the required accuracy. The IGE calibration would be very effective in correcting for settling errors between points 1 and 2, or 3 and 4. Its effectiveness decreases progressively as we move further to the left beyond points 2 and 4 due to the increased non-linearity in the settling errors.

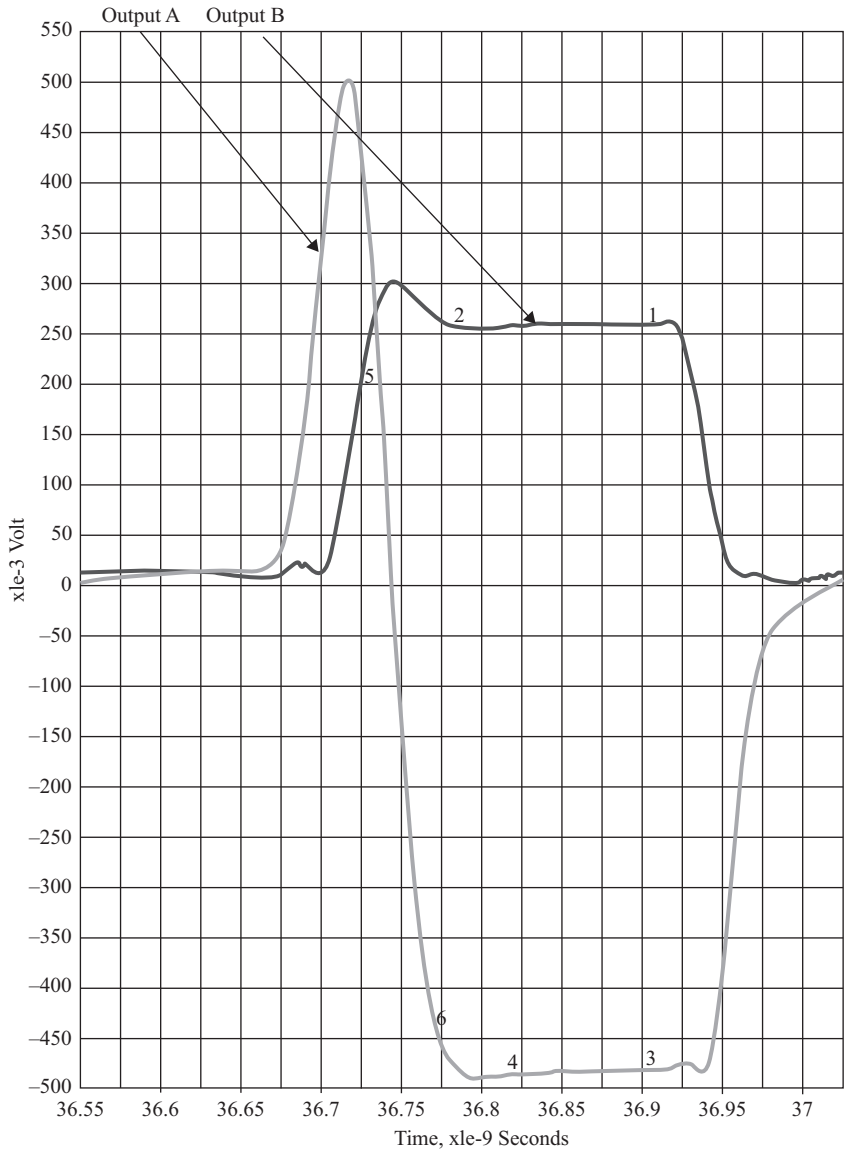


Figure 9.31 *A simulation plot of the MDAC output showing the large- and small-signal settling in two cases. A: The curve shows the output going in the wrong direction (due to the DAC delay), then recovering. B: The curve shows a case where the output goes directly to the right direction and hence has a relatively short period of large-signal settling. The large-signal settling time is different by about 40 ps between the two cases*

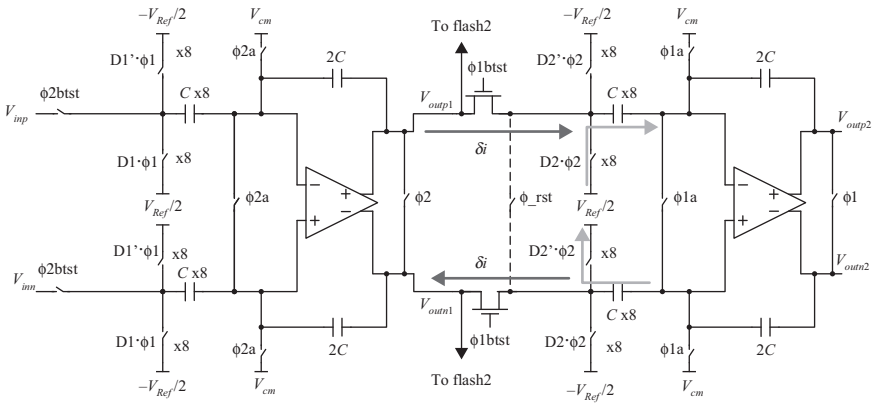


Figure 9.32 A simplified schematic showing the non-linear charge injection from stage-2 onto stage-1. During ϕ2, the capacitances of stage-2 are connected to $V_{Ref}/2$ or $-V_{Ref}/2$ based on the DAC code. During ϕ1, the capacitances are connected to the output of stage-1. Non-linear charge injection (δi) results from the DAC charge on the capacitance and the charge in the input switch

Another factor that limits the settling linearity is the memory and kick-back errors. This is illustrated in Figure 9.32. When ϕ2 is high, the stage-2 DAC is connected to the reference according to its DAC code. When ϕ1 is high, the capacitors are connected to the output of the stage-1 amplifier. If the stage-2 capacitors are not reset before being connected to the amplifier output, there will be non-linear charge injection (kick-back) on the stage-1 amplifier because of their initial charge that depends on the previous DAC code. Since that charge is a quantized version of the previous output and hence is highly non-linear, the charge injection will be a non-linear memory error. This indicates that the term $v_{initial}e^{-t_s/\tau}$ in (9.25) is not zero and is highly non-linear. Normally, if the amplifier has adequate time for settling, that non-linear term will dissipate exponentially together with the amplifier's output settling. However, if the settling time is too short, that error will not settle, and may not be corrected by the IGE calibration because of its non-linear nature and its dependence on the previous sample. This indicates that the effective correction of settling errors may sometimes require the correction of the memory and kick-back errors as well [5, 39].

9.1.7 Memory calibration

Memory errors can be caused by dielectric relaxation/absorption in capacitors, incomplete resetting of amplifiers, and incomplete settling from previous charges. It can also be caused by charge injection from the capacitances or switches of the following stages that are not fully settled. The correction of this class of errors can be accomplished using Finite Impulse Response (FIR) filters in the digital

domain [12]. The FIR filters will correct the present sample using memory coefficients that apply to the previous samples. Detecting the magnitude of the memory errors can be done using correlation-based approaches by injecting a dither signal in the MDAC, as is done for the IGE calibration, and utilizing the LMS algorithm [13] to detect its memory content, such that

$$Ge_{n+1,k} = Ge_{n,k} - \mu \times V_d[n-k] \times (V_d[n-k] \times Ge_{n,k} - V_R[n]) \tag{9.29}$$

where $V_d[n-k]$ is the kth past dither value and $Ge_{n+1,k}$ is the estimate of the gain coefficient from the kth past sample on the present nth sample. If $k=0$, then $Ge_{n+1,0}$ is the coefficient for the inter-stage gain and linear settling error. If $k>0$, then $Ge_{n+1,k}$ is the kth memory coefficient.

The correction and equalization are performed digitally using FIR filters whose M memory taps are ($Ge_{n+1,k}$), as follows:

$$V_{R\_cal}[n] = \sum_{k=0}^{M} V_R[n-k] \times Ge_{n+1,k} \tag{9.30}$$

where V_{R_cal} is the calibrated output residue, and V_R is the uncalibrated residue. We can see from (9.29) and (9.30) that the memory calibration is an extension of the correlation-based IGE calibration, by incorporating the previous samples in addition to the current sample.

9.1.8 Kick-back calibration

A new category of calibrations that has been recently proposed is the kick-back calibration [5]. This calibration corrects for errors due to the non-linear charge injection (kick-back) from the ADC's switched capacitor sampling network onto the ADC's input driver. If the MDAC's input sampling network uses the same capacitors for input sampling and the DAC operation, which is desirable for faster settling and lower noise, a non-linear charge injection occurs due to the DAC charge stored on the capacitors in the previous gain phase. A reset switch controlled by a brief reset pulse is usually used as shown in Figures 9.33 and 9.34. However, the reset pulse consumes a portion of the sampling (acquisition) time, which reduces the time available for the non-linear kick-back coming from other sources to settle. This degrades the distortion as discussed in Chapter 4. In addition, generating such a short pulse increases the power consumption.

This non-linear kick-back can be quite severe in buffer-less ADCs. When an input buffer is used to drive the sampling network, it provides a low output impedance that is capable of driving the sampling capacitance at the desired sampling rate. However, buffers have finite isolation, which can be in the order of only 6–10 dB for source followers. Some of the kick-back propagates to the ADC driver's network whose bandwidth is limited by the driver's impedance and the ADC's input capacitance. This limits the settling accuracy and the resulting performance of the ADC.

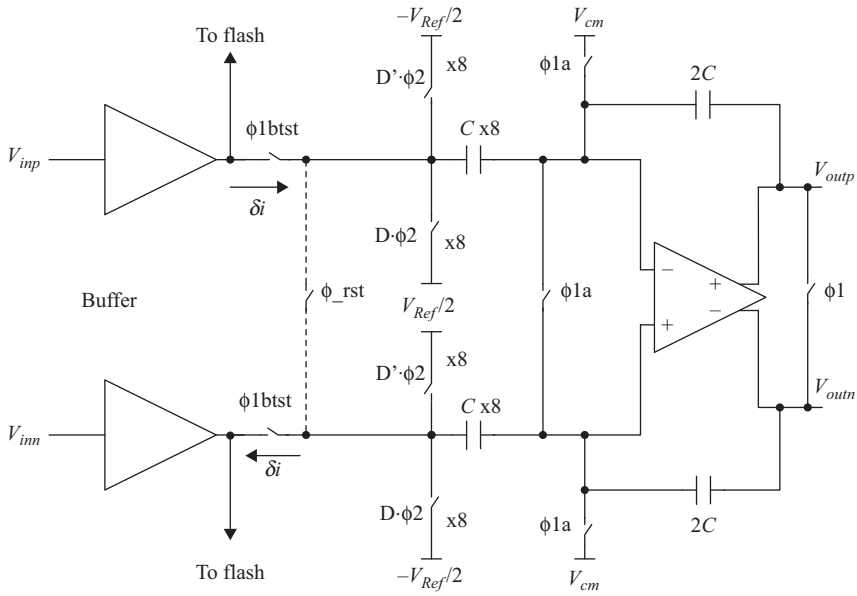


Figure 9.33 A simplified schematics of the first stage MDAC driven by input buffers. The reset switch is used to briefly discharge the sampling capacitances after the gain phase and before the next sampling phase. © 2014 IEEE. Reprinted, with permission, from Reference 5

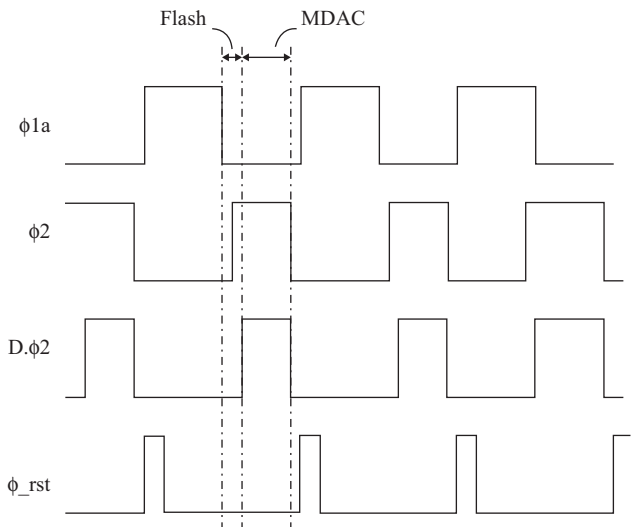


Figure 9.34 A simplified timing diagram showing the MDAC gain phase, the flash time, and input sampling phase. The ϕ_rst clock is used to briefly discharge the capacitances to discharge the previous DAC charge before the next sampling phase. This is done to reduce the non-linear charge injection (kick-back) on the input driver. © 2014 IEEE. Reprinted, with permission, from Reference 5

Since this non-linear kick-back is proportional to the previous DAC values, the previous codes of the first flash can be used to correct for the resulting distortion as follows [5]:

$$V_{out\_kbcal}[n] = V_o[n] + \sum_{i=1}^{M_{kb}} D_1[n-i] \times Gkb_{n+1,i} \tag{9.31}$$

where D_1 is the digital code of the first stage flash and V_o is the ADC output code. The M_{kb} kick-back coefficients (Gkb_i) are obtained using the LMS algorithm by injecting an uncorrelated PN calibration signal that "kicks" the input during the sample phase. This is shown in Figure 9.35.

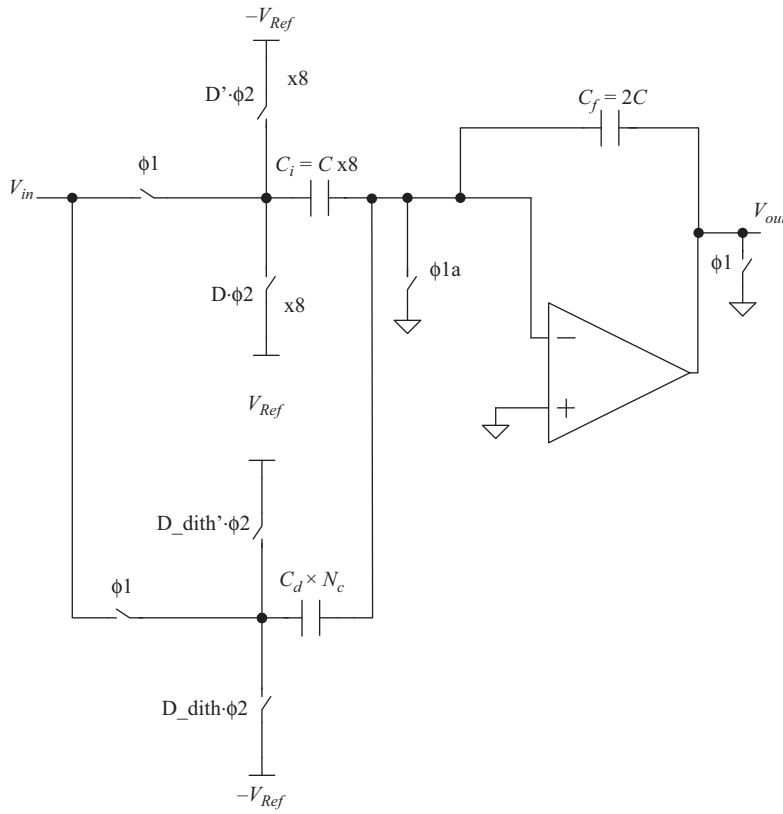


Figure 9.35 A simplified schematic showing the dither injection for the kick-back calibration. During the gain phase, the dither is applied to the dither capacitance. During the sampling phase, the dither capacitance is connected to the input in order to "kick" the input in a manner similar to the sampling capacitances. © 2014 IEEE. Reprinted, with permission, from Reference 5

During the gain/hold phase, the PN signal is stored on additional "kick-back" dither capacitances in a manner similar to the PN signal injection for the inter-stage gain and memory error calibration. In the sampling phase, the kick-back capacitors are connected to the input in parallel with the MDAC sampling capacitors. These capacitances "kick" the input with the stored PN signal charge, which then gets sampled at the end of the sampling phase and digitized with the input signal. The LMS algorithm is used to remove that kick-back PN signal and to estimate the kick-back coefficients using the recursive formula [5]

$$Gkb_{n+1,k} = Gkb_{n,k} - \mu \times Vd[n-k] \times (Vd[n-k] \times Gkb_{n,k} - V_{in}[n]) \quad (9.32)$$

where $Vd[n-k]$ is the kth past dither value and $Gkb_{n+1,k}$ is the estimate of the kick-back coefficient from the kth past sample on the present nth sample.

This calibration is similar to the memory calibration described in the previous section, and is an extension of the correlation-based calibration techniques. Unlike the other calibration methods, this technique involves injecting dither onto the ADC input driver. Since the impedance and bandwidth of the network driving the ADC will affect how it reacts to the injected dither, care must be taken in implementing this technique in order to ensure that adequate number of memory taps are employed.

The kick-back coefficients depend on the process, supply, temperature, sampling rate and the ADC driving network. The kick-back calibration algorithm operates in the background in order to adaptively correct for the changing conditions. An example is shown in Figures 9.36 and 9.37, where the INL improved from 5 LSBs down to less than 1 LSB. Unlike IGEs that manifest themselves as a saw-tooth pattern, kick-back memory errors are usually irregular and depend heavily on the frequency of the input signal.

This technique can be extended to the following pipeline stages to correct for the non-linear charge injection of the DAC charge of one stage onto the previous stage, as discussed earlier and shown in Figure 9.32. In this case, the dither is injected in a way similar to Figure 9.35 and the kick-back error can be corrected as follows

$$V_{R\_cal,i}[n] = V_{R,i}[n] + \sum_{k=0}^{M} D_{i+1}[n-k] \times Gkb_{n+1,k,i} \quad (9.33)$$

where $V_{R\_cal,i}$ is the calibrated output residue of the ith stage, $V_{R,i}$ is the uncalibrated residue of the ith stage, D_{i+1} is the flash code of the following stage, and $Gkb_{n+1,k,i}$ is the kick-back calibration coefficient of the kth previous sample on the present sample obtained using the LMS algorithm as shown in (9.32) when applied to the ith stage.

9.1.9 Residue amplifier non-linearity

In addition to the gain error, the RA may suffer from non-linearity that can limit the performance of the pipelined ADC, even with IGE correction. There are multiple possible causes of the non-linearity, which include the amplifier's slewing,

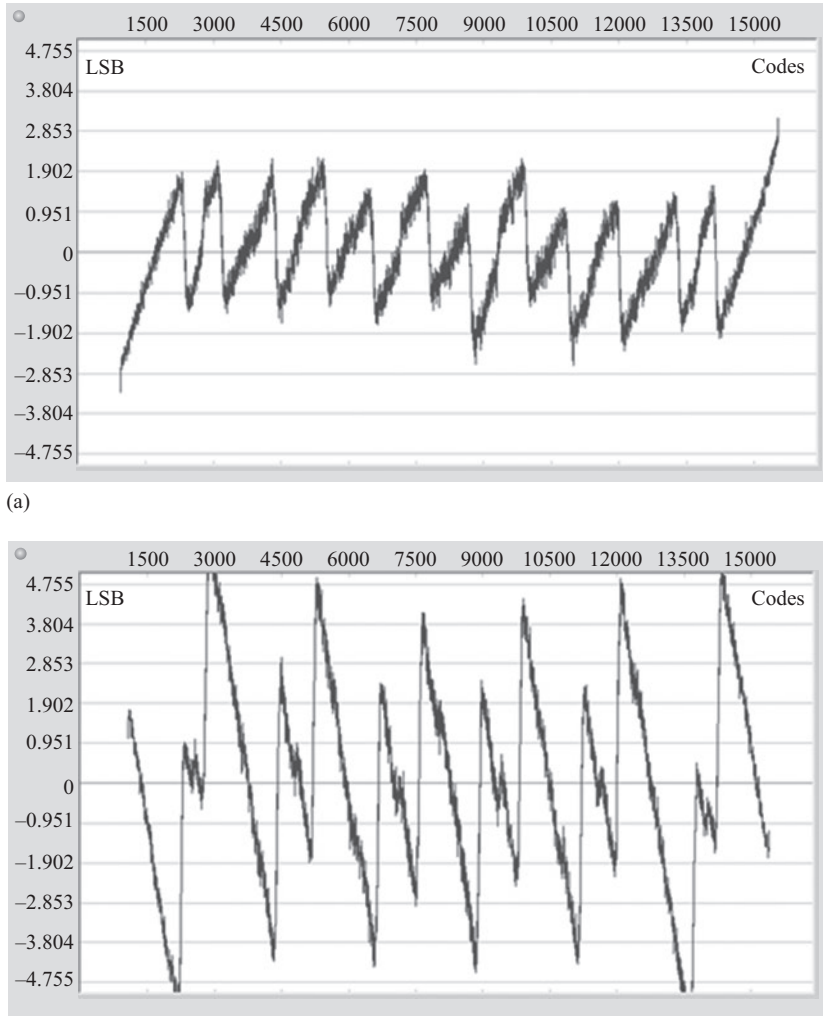
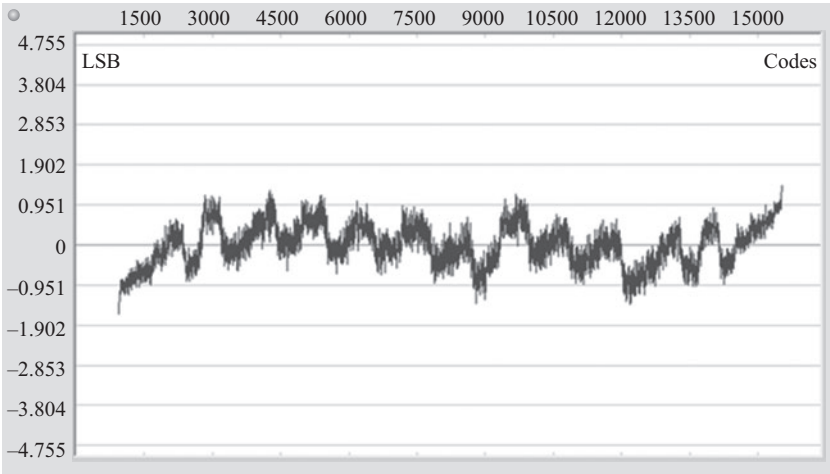
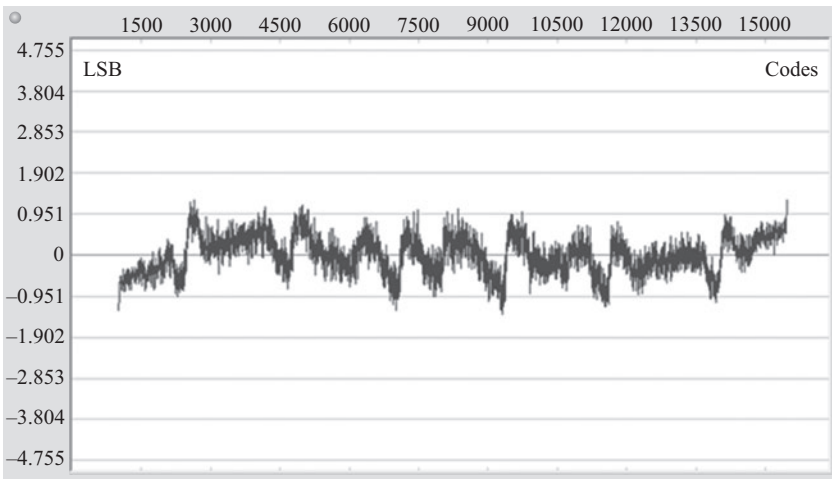


Figure 9.36 INL plots showing the effects of the non-linear kick-back for a 14-bit 1GS/s for an input frequency of (a) 700MHz, (b) 900MHz. © 2014 IEEE. Reprinted, with permission, from Reference 5

the DAC and reference large-signal settling, the amplifier's common-mode settling, and the non-linear charge injection. It is preferable for the RA to have a low gain that is uniform across its dynamic range, than to have high gain for small amplitudes with a highly compressive non-linear characteristic. The uniformly low gain is more easily correctable with the IGE calibration, and is an example of a calibration-friendly analog design. However, that is not always possible. In addition, we may need to push the speed or performance beyond the linear settling region of



(a)



(b)

Figure 9.37 INL plots with kick-back calibration for a 14-bit 1GS/s for an input frequency of (a) 700MHz, (b) 900MHz. © 2014 IEEE. Reprinted, with permission, from Reference 5

the amplifier, and hence have to calibrate the amplifier's non-linearity as well. In this section, we discuss several approaches to calibrate the amplifier's non-linearity, which include correlation-based, statistical, and deterministic techniques.

The non-linearity of the inter-stage amplifier can be represented by modifying (9.2) of the *i*th stage residue V_{oi} as follows

$$V_{oi} = V_{oi}|_{\text{ideal}} + \alpha_1 V_{oi} + \alpha_2 V_{oi}^2 + \alpha_3 V_{oi}^3 + \cdots = V_{oi}|_{\text{ideal}} + \sum_{k=1}^{m} \alpha_k V_{oi}^k \quad (9.34)$$

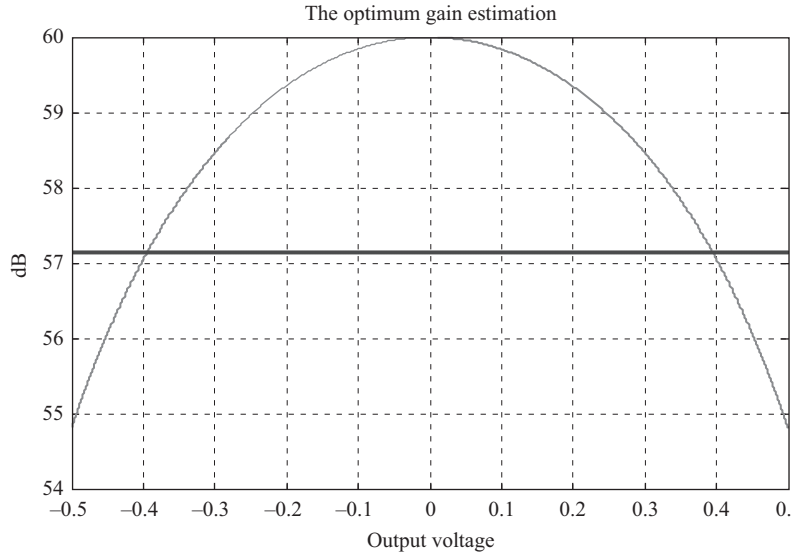


Figure 9.38 A plot of a non-linear amplifier gain with a fixed gain estimate also shown

where α_k is the kth order error term, and m is the highest order non-linearity considered. In the digital domain, these errors can be corrected by applying the opposite non-linearities to the digital residue $D(V_{oi})$ which can be approximated as

$$D(V_{oi})|_{\text{cal}} \approx D(V_{oi}) - \alpha_1 D(V_{oi}) - \alpha_2 D(V_{oi})^2 - \alpha_3 D(V_{oi})^3 - \cdots$$

$$\approx D(V_{oi}) - \sum_{k=1}^{m} \alpha_k D(V_{oi})^k \tag{9.35}$$

where $D(V_{oi})|_{\text{cal}}$ is the digital representation of the ith stage residue, and $D(V_{oi})|_{\text{cal}}$ is the ith stage calibrated digital residue.

In order to illustrate the possible need to calibrate the amplifier's non-linearity in some situations, an example is shown in Figure 9.38 for an amplifier with a compressive gain non-linearity. The gain of the amplifier changes from 60 dB for small signals down to 55 dB near the full-scale. Without any calibration, the SFDR of the resulting pipelined ADC is about 70 dB and the SNDR is limited to 64 dB, as shown in Figure 9.39. Calibrating for the gain error term gives a gain estimate of about 57 dB. The SFDR performance will improve to about 80 dB and the SNDR to about 75 dB, as shown in Figure 9.40. This indicates a substantial improvement, despite the residual non-linearity of the amplifier.

Using piecewise linear approximation of the non-linear gain with two segments, as shown in Figure 9.41, gives an SFDR performance of about 90 dB and SNDR of about 85 dB, as shown in Figure 9.42. Finally, using a third-order polynomial as shown in Figures 9.43 gives an SFDR and SNDR performance of better

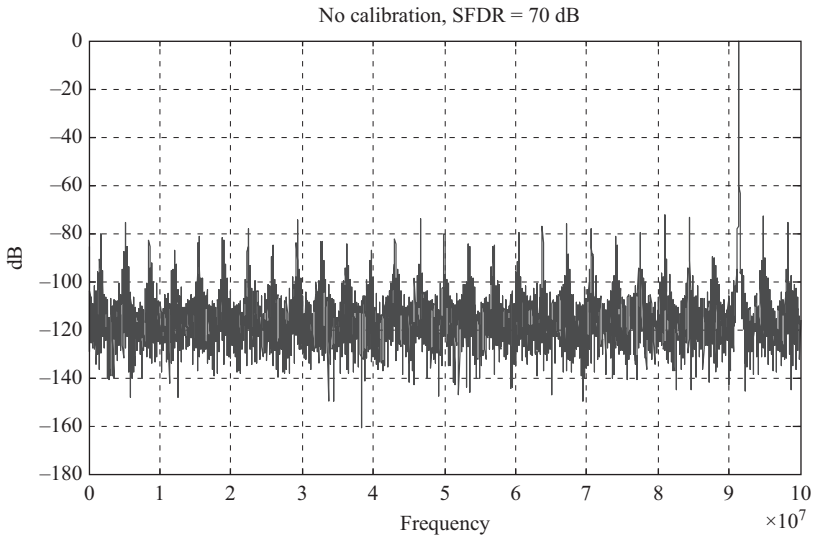


Figure 9.39 An FFT of the digital output using the MDAC amplifier shown in Figure 9.38 without any correction. SFDR = 70 dB and SNDR = 64 dB

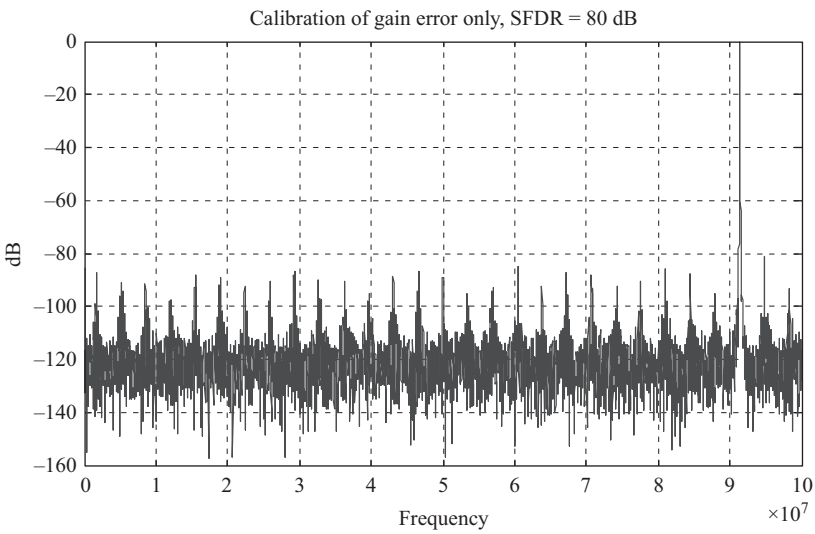


Figure 9.40 An FFT of the digital output using the MDAC amplifier shown in Figure 9.38 with fixed gain correction using the estimate shown in Figure 9.38. SFDR = 80 dB and SNDR = 75 dB

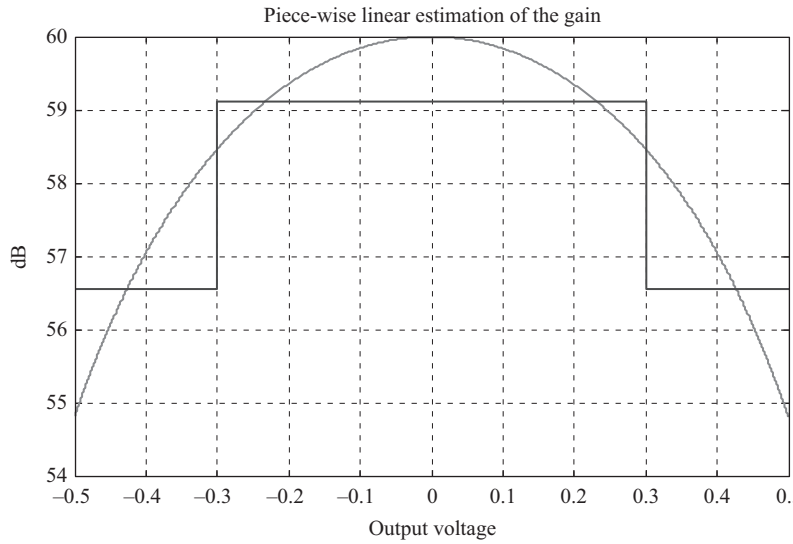


Figure 9.41 A plot of a non-linear amplifier gain with a two-segment piece-wise linear estimate of the gain also shown

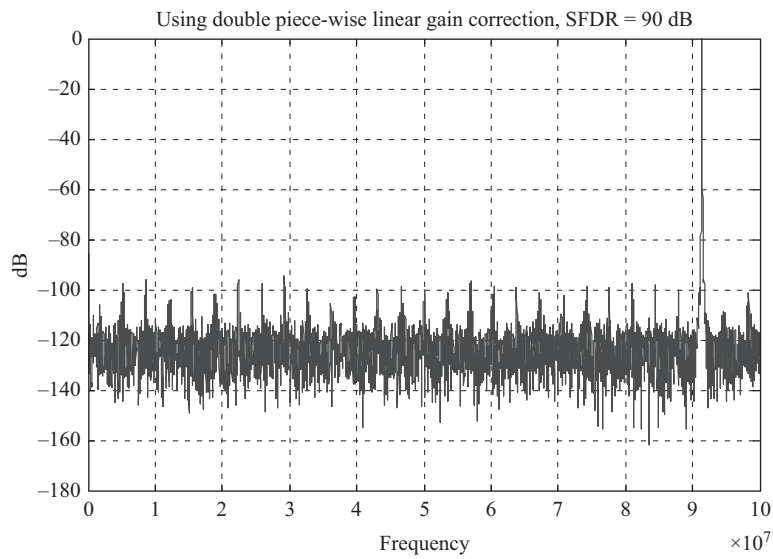


Figure 9.42 An FFT of the digital output using the MDAC amplifier shown in Figure 9.41 with the piece-wise linear gain correction using the LMS estimates shown in Figure 9.41. SFDR = 95 dB and SNDR = 85 dB

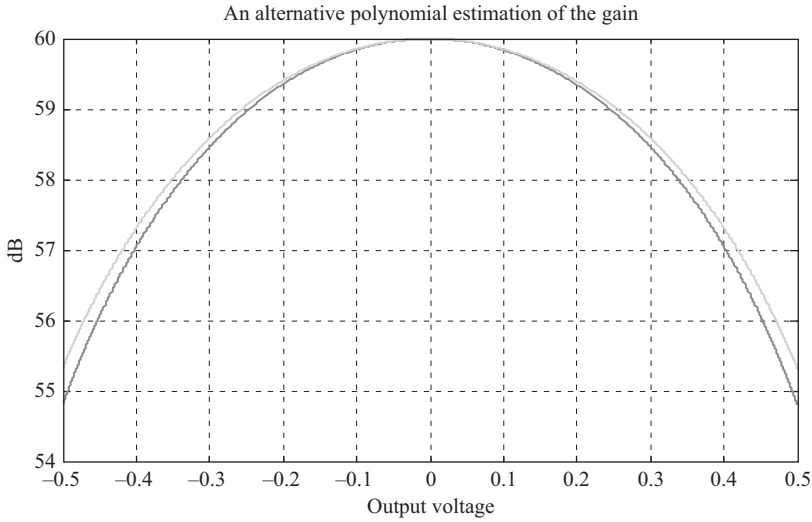


Figure 9.43 A plot of a non-linear amplifier gain with a polynomial estimate of the gain

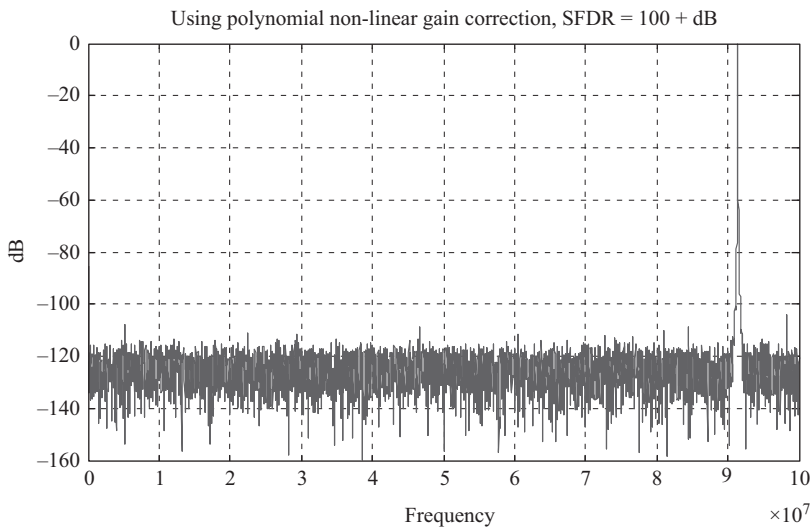


Figure 9.44 An FFT of the digital output using the MDAC amplifier shown in Figures 9.43 with the polynomial correction using the LMS estimate shown in Figures 9.43. SFDR = 110 dB and SNDR = 110 dB

than 110 dB as shown in Figure 9.44. That is, the SNDR will be practically limited by the thermal noise, which is modeled to be quite low in this case. In every case the digital correction represents the *inverse* of the analog error and non-linearity to correct their impact. This example illustrates the importance of correcting the

amplifier's non-linearity and the progressive improvement in performance with gradually approaching the amplifier's real characteristics. It also illustrates the effectiveness of the piece-wise linear correction even with a few number of segments. In the following sections, we discuss some techniques to correct for the amplifier non-linearity.

9.1.9.1 Correlation-based method

The correlation-based IGE calibration can be extended to correct for the amplifier's non-linearity. This can be achieved by using multiple uncorrelated PN signals in the stage to be calibrated [9]. In the digital back end, the cross-correlation between the PN signals is estimated and used as a measure of the non-linearity. For example, a third-order non-linearity can be estimated using the cross correlation of three uncorrelated dither signals; a fifth-order non-linearity requires five dither signals, and so on.

An implementation of this calibration approach is discussed in References 8–10, where m uncorrelated dither signals are injected into the MDAC. The residue can be represented by

$$V_R[n] = V_R|_{\text{ideal}}[n] + \sum_{k=1}^{m} \alpha_k V_R^k[n] \tag{9.36}$$

where α_1 is the gain error term, α_2 is the second-order distortion term, α_3 is the third-order distortion term, and so on. Ignoring the even order terms, which are minimized by using a differential implementation, the algorithm's goal is to estimate the odd α terms. This is done by estimating the odd γ terms that represent the cross-correlations between the dither signals, such that the first-order term is

$$\gamma_1 = -K_1 \times \mathrm{E}\left[V_{d1}[n]\left(V_R[n] + \sum_{k=1}^{m} V_{dk}[n]\right)\right] \tag{9.37}$$

where E[z] is the estimation (or average value) of z. V_{dk} is the kth dither signal, V_R is the residue signal, and K_1 is a constant. The third-order term is expressed as

$$\gamma_3 = -K_3 \times \mathrm{E}\left[V_{d1}[n]V_{d2}[n]V_{d3}[n]\left(V_R[n] + \sum_{k=1}^{m} V_{dk}[n]\right)\right] \tag{9.38}$$

The fifth-order term is

$$\gamma_5 = -K_5 \times \mathrm{E}\left[V_{d1}[n]V_{d2}[n]V_{d3}[n]V_{d4}[n]V_{d5}[n]\left(V_R[n] + \sum_{k=1}^{m} V_{dk}[n]\right)\right] \tag{9.39}$$

where

$$K_i = \frac{A^{-2i}}{i!} \tag{9.40}$$

and A is the amplitude of each calibration dither signal, E[z] is the estimation (or average value) of z, $V_R[n]$ is the residue signal, and $V_{dk}[n]$ is the kth dither signal.

The cross-correlation γ values are not equal to the α values because of the effect of the higher-order terms. It can be shown that [9, 10]

$$\gamma_3 = \alpha_3 + \alpha_5\left[30A^2 + 10Y[n]\right] \tag{9.41}$$

and

$$\gamma_1 = \alpha_1 + \alpha_3\left(13A^2 + 3Y[n]\right) + \alpha_5\left\{241A^4 + 130A^2(Y[n] + 5Z[n])\right\} \tag{9.42}$$

where,

$$Y[n] = \mathrm{E}\left[\left(V_R[n] + \sum_{k=1}^{m} V_{dk}[n]\right)^2\right] \tag{9.43}$$

and

$$Z[n] = \mathrm{E}\left[\left(V_R[n] + \sum_{k=1}^{m} V_{dk}[n]\right)^4\right] \tag{9.44}$$

Therefore, after estimating the γ values, the α values will need to be estimated as shown in Figure 9.45 [9].

In spite of the demonstrated effectiveness of this algorithm, it suffers from long convergence time. The number of samples needed depends on the required

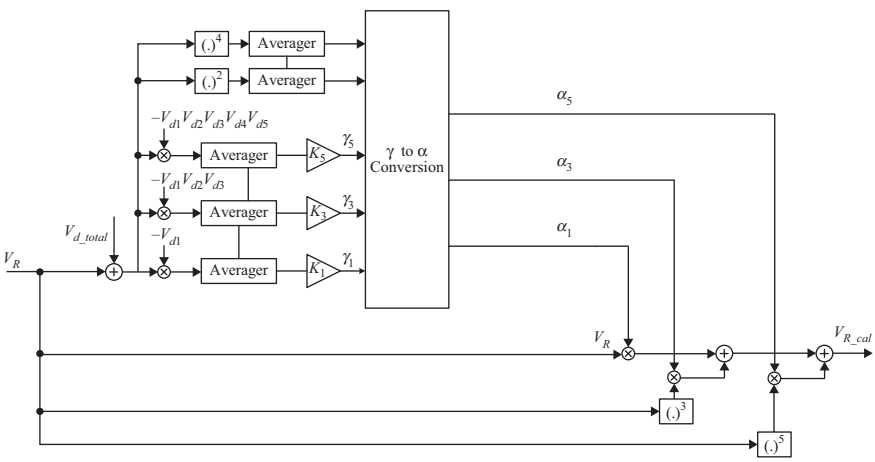


Figure 9.45 A block diagram showing the correlation-based calibration of the MDAC amplifier's IGE and non-linearity. © 2006 IEEE. Reprinted, with permission, from Reference 9

accuracy ε, the amplitude of the signal, the amplitude of the dither, and the number of dither signals, which is related to the order of the non-linearity that needs to be corrected. This is approximated as follows [9]

$$N \propto \frac{V_{signal}^{2k}}{(k!)^2 \varepsilon^2 V_{dither}^{2k}} \tag{9.45}$$

where ε is the allowed convergence error, V_{dither} is the amplitude of each dither signal, V_{signal} is the signal amplitude, and k is the order of the distortion to be corrected. The number of samples required for third-order non-linear correction can be in the billions, which would result in minutes of convergence time for sampling rates below 100 MS/s [9, 10].

9.1.9.2 Statistics-based approach

In statistical approaches, histograms are used to deduce the IGE and the amplifier's non-linearity by statistically estimating the distances between two residue signals or between two sub-ranges of the stage under calibration. This can be done in the factory or foreground using the INL or residue waveforms for a large number of samples in the presence of the input, or by forcing the comparator to switch between two different decisions to measure the corresponding "jump" in the output residue. Extending this approach to correct for the amplifier's non-linearity is straightforward.

Alternatively, non-linear calibration can be employed in the background as proposed in Reference 23. In that work, a PN sequence is used to switch the flash thresholds between two modes and hence create enough variation in the output to estimate the IGE and non-linearity. This is shown in Figure 9.46, where an additional bit is added in the flash ADC to shift the thresholds between the two modes. If the amplifier were linear, the distances h_1 and h_2 would be equal. However, the amplifier's non-linearity causes them to be different, with h_2 typically being less than h_1 in the presence of compression. By measuring the distances h_1 and h_2 over a large number of input samples, estimates of the IGE and non-linearity can be obtained.

The distances h_1 and h_2 are measured by estimating the cumulative histograms of the codes that correspond to the locations of those parameters. Once determined, they can be used to estimate the parameters p_1 and p_2, which are used to perform the gain and non-linear correction, as shown in Figure 9.47.

The estimation is performed using the LMS algorithm. One loop (with step size μ_1) uses an estimate of the parameter h_2 to estimate the gain correction parameter p_1, which is applied to the residue as a gain correction term. The term h_{ideal} is used as a correction factor to account for constant gain errors that need to be zeroed out. The other loop (with step size μ_2) uses estimates of both h_1 and h_2 to estimate the parameter p_2, which is used with look-up tables to correct for the amplifier's non-linearity.

This method requires an input signal with acceptable distributions that can be restrictive to its application. It also requires some analog design overhead, which is

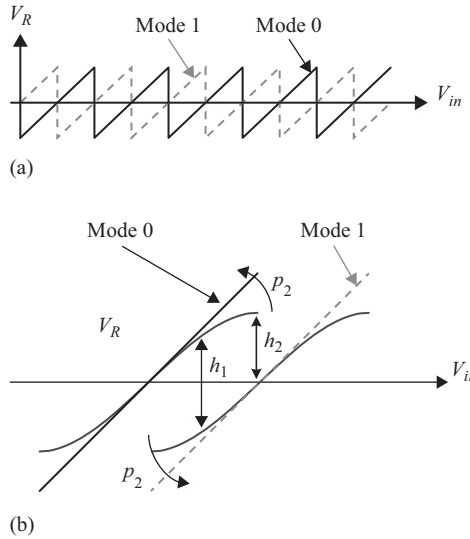


Figure 9.46 A plot of the stage's residue in the two modes of operation. (a) Two modes of operation for a linear residue. (b) Two modes in the presence of non-linearity [22–24]. © 2003 IEEE. Reprinted, with permission, from Reference 23

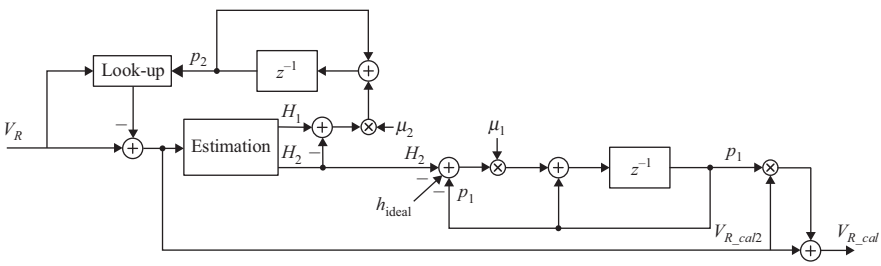


Figure 9.47 A block diagram showing the statistical calibration of the MDAC amplifier's IGE and non-linearity [23, 24]. © 2003 IEEE. Reprinted, with permission, from Reference 23

not substantial. In addition, it requires digital processing and storage that can be significant.

9.1.9.3 Using summing node sampling

A third approach to calibrate the MDAC amplifier's non-linearity is to use a deterministic approach, such as: the summing node algorithm [7]. The summing node voltage is sampled and used to estimate the gain for different ranges of signal amplitudes, using separate buckets for the LMS algorithm. Then, piece-wise linear or polynomial fit can be employed to approximate and correct for the non-linearity.

Unlike correlation-based approaches, this technique has very fast convergence, which is in the order of a few thousand samples, and does not cause dither correlation issues when dividing the samples into buckets.

9.1.10 Coupling calibration

Correlation-based approaches can be extended to correct for coupling errors. This is achieved by injecting an uncorrelated PN signal at the location of the offending signal, and detecting it at the destination. Using the LMS algorithm, the coupling coefficient can be estimated. This is effective in reducing cross-talk, digital coupling, and inter-channel coupling in multi-channel ADCs [25].

9.2 Dither

In spite of the effectiveness of the calibration algorithms, the linearization of the ADCs may still be limited in its accuracy after calibration. *Subtractive dither* can be used to linearize any residual non-linearity by randomizing the non-linear effects and hence spreading them in the noise floor. Subsequently, it is subtracted in the digital back end. Unlike calibration, dithering does not "fix" the errors or improve the SINAD of the ADC. In fact, it sometimes degrades the SINAD because of the imperfect subtraction of the dither signal. However, it can substantially improve the linearity, and hence the SFDR of the ADC. In addition, it can also reduce the variation of the ADC gain with the input amplitude, and improve the accuracy of the background calibration by linearizing the back end, as described earlier.

The dither can be injected on the input signal to propagate down the pipeline and be eventually subtracted in the digital back end. The LMS algorithm can be used to calibrate the dither's subtraction in the background to track changing temperature, supply, and sampling rate, as shown in Figure 9.48.

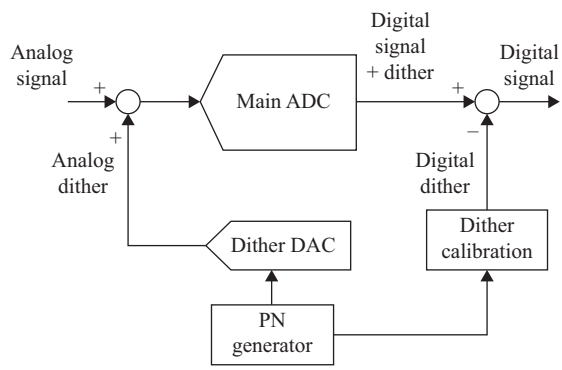


Figure 9.48 A block diagram showing the dither injection and calibration. The dither is used to linearize the ADC

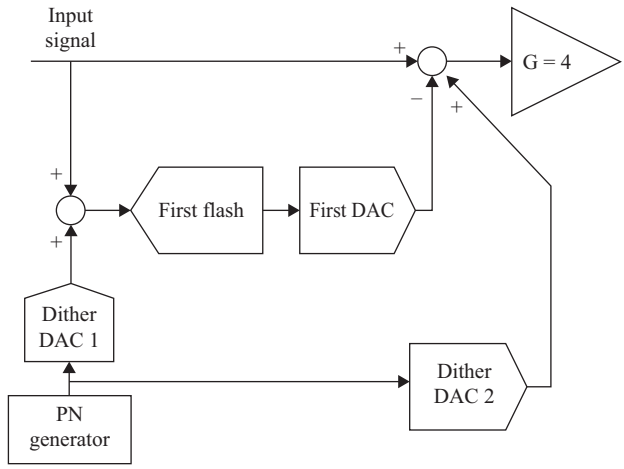


Figure 9.49 A block diagram showing the dither injection in the pipeline stage. The dither is injected in both the MDAC and the flash. It does not use any portion of the correction range

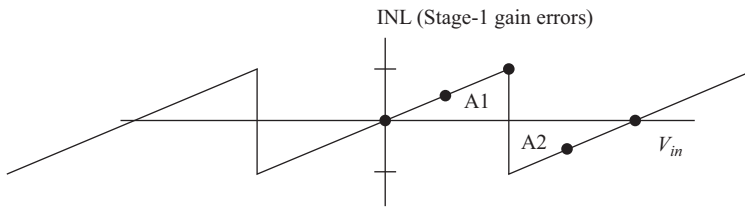


Figure 9.50 A plot of the residue showing the inter-stage gain error and the dither's effectiveness. If A1 is equal to A2, a dither with a magnitude equal to the sub-range size (or multiples of it) is capable of dithering the saw-tooth pattern that results from the inter-stage gain error

In pipelined ADCs, the dither can be injected in both the MDAC and the flash of the first stage, as shown in Figure 9.49. This is equivalent to injecting it on the input of the ADC, which is necessary to effectively dither the IGE errors of the first stage.

It is important to note that a dither signal that is injected on the input of the ADC (i.e. in both the MDAC and flash of the ADC) will "see" the IGE as an INL saw-tooth pattern, as shown in Figure 9.50. Therefore, it will effectively dither that error if the area A1 is equal to A2 *and* if the dither is large enough to span the whole sub-range. However, this dither signal cannot be used as a calibration signal to measure the IGE because it sees the gain of the whole ADC, which is marked by the dotted lines emanating from the origin in Figure 9.51, and not the inter-stage gain.

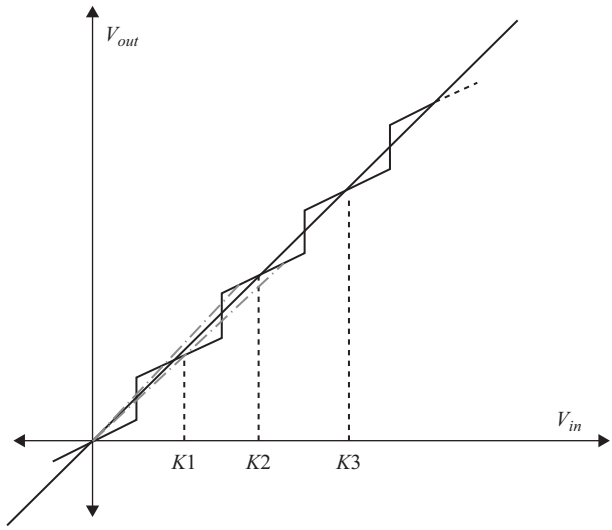


Figure 9.51 The output of an ADC versus its input in the presence of inter-stage gain errors. The different sub-ranges are denoted by K1, K2, K3, etc. The inter-stage gain error is given by the slope of the segments, the overall gain is given by the slope of the dotted lines from the origin to the operating point

On the other hand, a dither signal injected in the MDAC *only* (but not in the flash) can be used as a calibration signal to measure the IGE, but cannot effectively dither the IGE of the stage in which it is injected. This dither experiences the inter-stage gain given by the slope of the various sub-ranges in Figure 9.51, but does not "cross" the sub-ranges, and hence cannot dither them. However, it can dither the errors of the back-end stages, as long as it is amplified to be large enough to cover a full sub-range of that back-end stage.

Another observation is that a dither injected in the MDAC only will consume a portion of the correction range, while a dither injected in both the MDAC and flash will not consume (at least ideally) any portion of the correction range. It may consume, however, a portion of the ADC's dynamic range because it is added to the input signal. This can be avoided by using an extra comparator in the first-stage flash, such that the last two sub-ranges on both sides are only half-used when a full-scale input signal is applied, as was shown previously in Figure 9.20 and again in Figure 9.52. The dither can occupy the two remaining half sub-ranges beyond the full-scale range, without compromising the dynamic range or the linearity of the ADC, as long as its peak-to-peak amplitude is equal to or less than a sub-range. Fortunately, this amplitude is usually adequate to effectively dither the IGE of the first stage.

In order to estimate the improvement in SFDR due to dithering, we need to review the SFDR due to a saw-tooth pattern, which is discussed in Chapters 1 and 2.

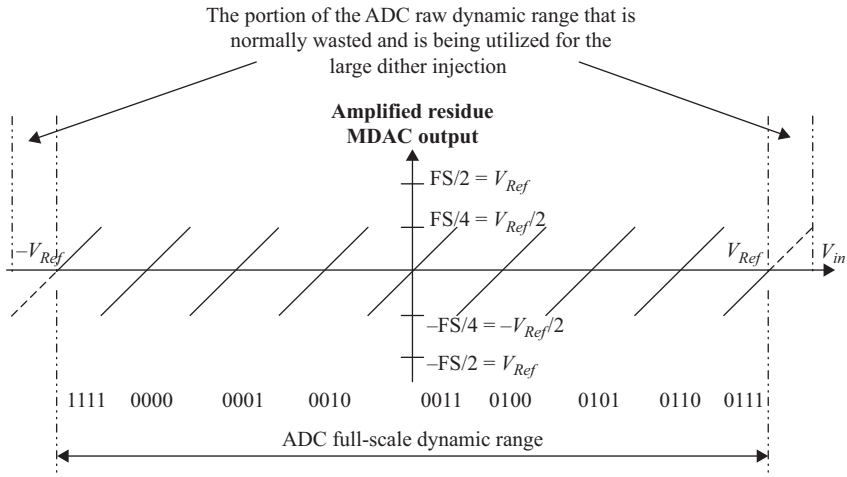


Figure 9.52 A stage-1 residue showing the portion of the dynamic range that is usually not utilized when using an additional comparator in the flash and can be consumed by the large dither signal

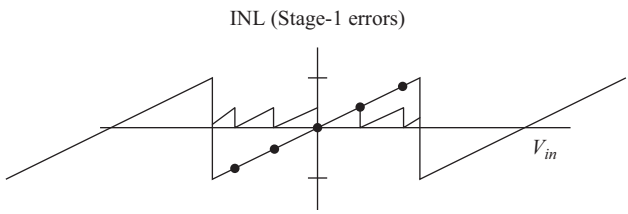


Figure 9.53 A plot of the residue (quantization error) showing the effectiveness of the dither in reducing the spurs due to the quantization noise by increasing the number of segments and reducing the peak magnitude. The total quantization energy is not improved

The saw-tooth error pattern looks like the standard quantization error, whose SFDR for n bits is given by

$$\text{SFDR} = 9n - c \tag{9.46}$$

where c is a constant. Every additional bit, reduces the error energy by 6 dB, and doubles the number of spurs, which results in an additional 3 dB reduction, and hence a total of 9 dB improvement. Since n-bit dithering can be thought of as quantization with an additional n bits, which happen to be random, the SFDR improvement due to dither is expected to follow the same quantization trend. Therefore, dithering a saw-tooth error pattern by n bits of dither (N levels, where $n = \log_2 N$) results in an improvement of about $9n$ dB in SFDR ($30 \log N$) at the same input amplitude. This is shown conceptually in Figure 9.53. In addition, the dither bits act as additional quantization bits in the stage, which effectively increase

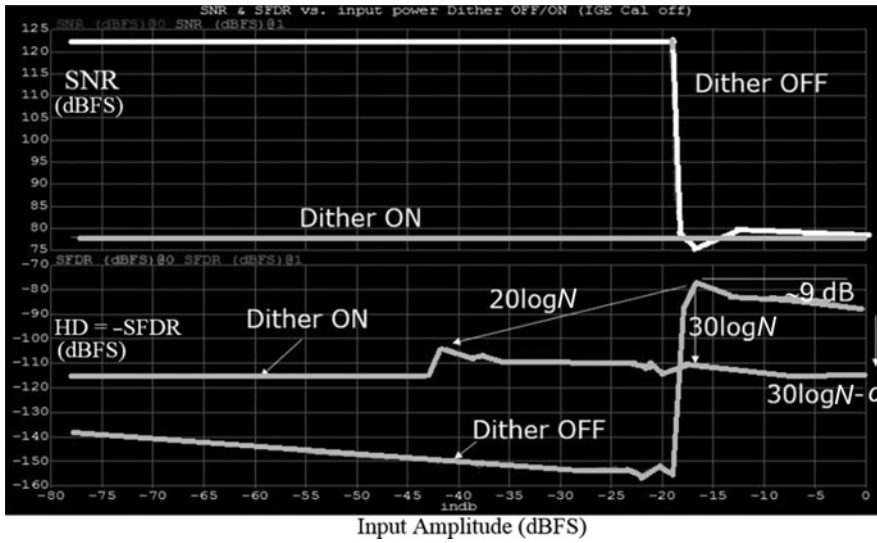


Figure 9.54 *A simulation plot showing the effect of dithering with 4 bits ($N = 16$) on SNR and SFDR for a 3-bit/stage pipeline with IGE in the first stage*

the number of sub-ranges and reduce the maximum error amplitude. Therefore, the worst case SFDR, which typically occurs at the boundary of a single sub-range, will occur at a different point that is lower in amplitude by $20 \log N$ and improve by $6n$ ($20 \log N$) in magnitude.

For example, if the pipeline stage is 3 bits, the worst case SFDR normally occurs at -18 dBFS, which is $-6k_1$ ($k_1 = 3$). If this stage is dithered by 16 levels of dither, the SFDR at the same amplitude (i.e. at -18 dBFS) will improve by $30 \log 16$ (i.e. 36 dB better). In addition, the SFDR notch will move by $6n$ ($n = 4$ bits of dithering) from -18 dBFS to -42 dBFS, and its level will improve by $20 \log 16$, which is 24 dB. This is illustrated in Figure 9.54, which shows the SNR and SFDR simulation results with and without dither, in the presence of an IGE in stage-1.

In Figure 9.54, we can also see that the dither does not improve the SNR. In fact, without dither, the SNR improves substantially for input signals that are smaller in amplitude than a single sub-range of stage-1. In the presence of dither, the dithering of the IGE degrades the SNR for very small signals.

So to summarize, if we use N levels of dither, we expect:

- The improvement in SFDR at the same amplitude to be: $30 \log N$
- The improvement in SFDR at the worst case amplitude to be: $20 \log N$
- The amplitude of the worst case SFDR is reduced by: $20 \log N$.

Measurement results are shown in Figures 9.55 and 9.56 for the SFDR and SNR, respectively. This is a 16-bit 250 MS/s with 3-bits in the first stage and 4 bits/stage in the following stages. The results show the impact of using calibration and

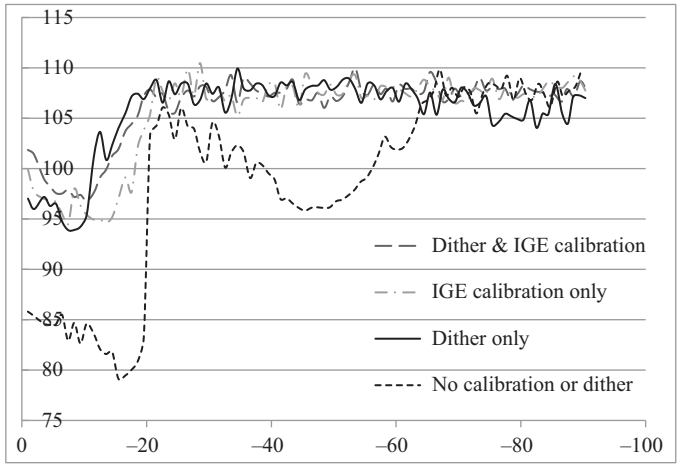


Figure 9.55 A plot of SFDR (in dBFS) versus input amplitude (in dBFS) showing the effect on SFDR due to dither, calibration, and both

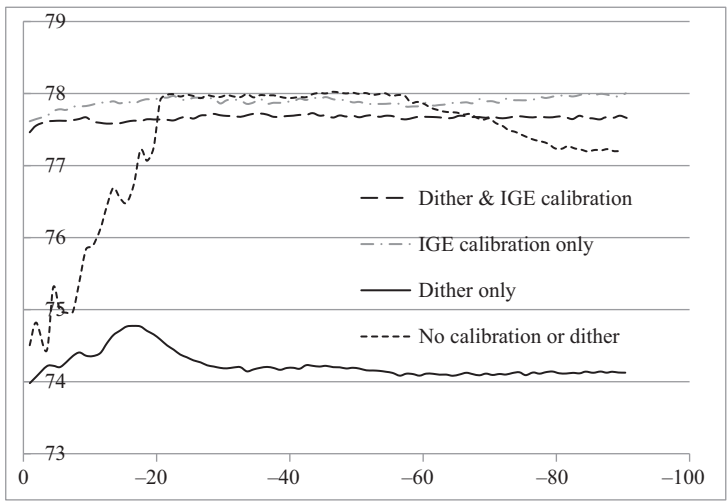


Figure 9.56 A plot of SNR (in dBFS) versus input amplitude (in dBFS) showing the effect on SNR due to dither, calibration, and both

dither. In the baseline curve of Figure 9.55, without calibration or dither, we can see degraded performance due to the inter-stage gain errors. In addition, the notches are clear at the stage-1 sub-range boundary at -18 dBFS ($-6k_1$, $k_1 = 3$) and the stage-2 sub-range boundary at -42 dBFS ($-6k_1 - 6k_2$, $k_1 = 3$, $k_2 = 4$). The SFDR improves substantially with dither or calibration. The best performance is clearly observed with both calibration and dither combined.

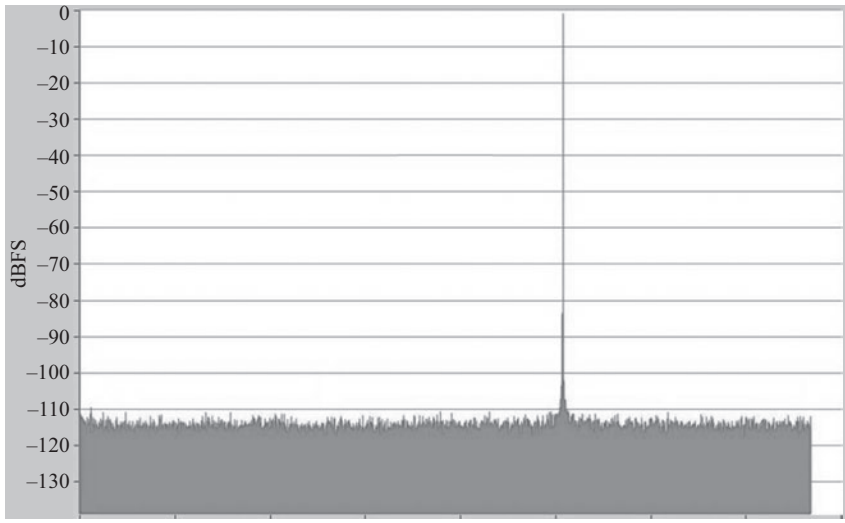


Figure 9.57 An FFT plot of an ADC showing the effectiveness of the dither in delivering an outstandingly clean spectrum with better than 110 dB of SFDR

Similarly, for the SNR curve in Figure 9.56, we see the baseline SNR, and the improvement with calibration. On the other hand, dither alone does not improve the SNR. In fact, it degrades the SNR as expected, and the degradation can be substantial for small signal amplitudes. This is because the calibration fixes the errors thereby improving the SNR, while the dither spreads the errors in the noise floor and hence does not improve the SNR but can actually degrade it in some cases. Once the linearization takes place, the dither is subtracted in the digital back end. Any imperfection in the dither subtraction will result in an SNR degradation.

An example of an output spectrum of an ADC with dither is shown in Figure 9.57. This is an astoundingly excellent linearity which represents the current state of the art and the best published linearity achieved for a high speed ADC in the literature [7].

9.3 Flash sub-ADC calibration

In addition to the MDAC, the calibration of the flash sub-ADC in a pipelined ADC has gained attention. The flash offsets lead to the consumption of a portion of the correction range, which impacts the MDAC's amplifier linearity and power consumption. Calibrating these offsets relaxes the requirements on the MDAC's dynamic range and helps reduce the power consumption. In addition, in SHA-less architectures, the bandwidth and timing mismatches between the MDAC and the flash create errors that also consume a portion of the correction range, and degrade with the input frequency. These mismatches impose an upper limit on the input frequencies that can be sampled by the ADC.

Although these offsets and mismatches can be factory calibrated, it is desirable to develop background calibration techniques that track the variation due to temperature, supply, sampling rate, and aging. Examples of these flash ADC calibrations can be found in References 37–39.

In Reference 39, a background calibration algorithm is described that uses the residue signal to correct the comparator offsets in the background, while averaging out errors due to bandwidth and timing mismatches. This separation of the static offsets from the errors due to phase mismatches is key to achieving accurate offset correction, and to prevent over- or under-correction. The technique is also independent of the input signal characteristics and distribution. It uses the first-stage flash code and the residue of the first stage to generate an error function given by

$$\varepsilon = \mathrm{E}\{[\mathrm{maxL}(V_R)|_{\mathrm{code=x}}] + [\mathrm{minL}(V_R)|_{\mathrm{code=(x+1)}}]\} \tag{9.47}$$

Alternatively, the digital output can be used instead of the residue as follows

$$\varepsilon = \mathrm{E}\{[\mathrm{maxL}(V_{out})|_{\mathrm{code=x}}], [\mathrm{minL}(V_{out})|_{\mathrm{code=(x+1)}}]\} - \mathrm{Ideal\_threshold} \tag{9.48}$$

where ε is the error function used to estimate the offset, V_R is the first stage's digital residue, V_{out} is the ADC digital output, E{z} is the expectation (or average) of all elements of z and E{z,y} is the expectation of all elements of z and y combined. The functions maxL() and minL() are leaky peak and trough detector functions, respectively. A leaky peak detector can be implemented as follows

$$V_p[n+1] = V_p[n] + \alpha(V_R[n] - V_p[n]) \tag{9.49}$$

where V_p is the peak value, V_R is the residue, and the factor α determines the time constant, such that:

$$\alpha = \begin{cases} \text{Large, if } V_R[n] - V_p[n] \text{ is positive} => \text{attack time constant is small (fast)} \\ \text{Small, if } V_R[n] - V_p[n] \text{ is negative} => \text{decay time constant is large (slow)} \end{cases}$$

On the other hand, the leaky trough detector is implemented as follows

$$V_{tr}[n+1] = V_{tr}[n] + \gamma(V_R[n] - V_{tr}[n]) \tag{9.50}$$

where V_{tr} is the trough value, V_R is the residue, and the factor γ determines the time constant, such that:

$$\gamma = \begin{cases} \text{Large, if } V_R[n] - V_p[n] \text{ is negative} => \text{attack time constant is small (fast)} \\ \text{Small, if } V_R[n] - V_p[n] \text{ is positive} => \text{decay time constant is large (slow)} \end{cases}$$

These values are accumulated until enough samples are collected to ensure adequate pairing between the two neighboring sub-ranges, and to average out the effects of the phase mismatch and hence eliminate its contribution. This is shown in Figure 9.58 for a 3-bit MDAC that has 8 sub-ranges, or more accurately, 7 sub-ranges plus 2 half sub-ranges. One of the comparators has an offset, which appears as a shift in the threshold value between codes 0110 and 0111. Since this is

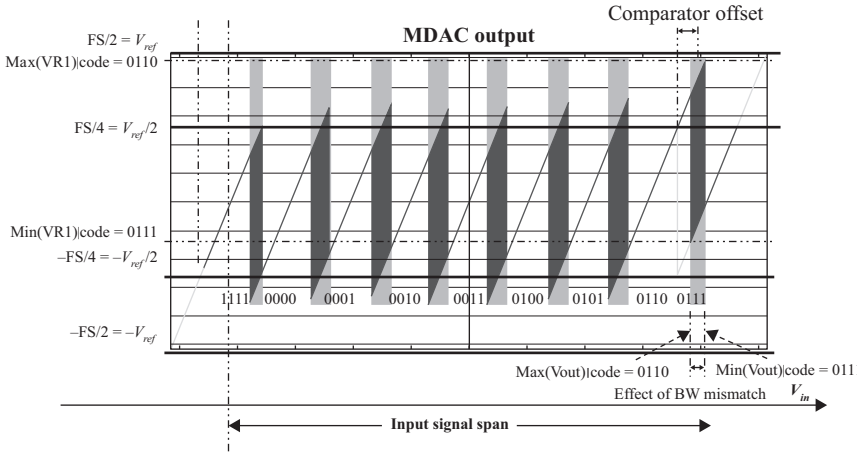


Figure 9.58 A simplified representation of the first-stage residue measured in the lab. The effect of the comparator offset is shown as a shift in one of the sub-ranges. The timing/BW mismatch between the flash and MDAC appears as the shaded areas at the boundaries between the sub-ranges. An example of the parameters used by the algorithm is superimposed on the plot [39]

a SHA-less ADC, the timing and bandwidth mismatches between the flash and the MDAC create errors that consume a portion of the correction range. These mismatch errors depend on the instantaneous amplitude and slope of the input signal. Therefore, they change sign and magnitude with the input signal and appear as shaded regions at the sub-range boundaries in Figure 9.58.

The offset control can then be given by: $V_{th} = V_{th0} + \mu\,\varepsilon$ (9.51)

where μ is a weighting factor that controls the time constant of the algorithm, V_{th} is the comparator's threshold voltage, and ε is the error defined in (9.47) or (9.48). This threshold voltage is fed back to the comparator to control its offset.

It is interesting to note that this flash ADC calibration is facilitated by the pipeline architecture. The MDAC's residue contains the needed information about the comparators' behavior. This can be processed in the digital back end to extract the offset and/or the bandwidth mismatch, and use that to feedback an offset correction parameter to the comparators in the flash, without disrupting their normal operation.

9.4 Calibration of mismatches in interleaved ADCs

In interleaved ADCs, background calibration can be used to estimate and correct inter-channel mismatches. These include offset, gain, and timing mismatches. Offset and gain mismatches are relatively straightforward to estimate and fix in the digital domain. Timing mismatch, on the other hand, can be quite challenging and

represents an active area of research. In the following sections, some mismatch calibration techniques are discussed. For simplicity, they will be shown in the context of two-way interleaving. However, the concepts can be applicable to larger numbers of channels.

9.4.1 Offset mismatch calibration

The offset of the ADC can be estimated using a low-pass filter or integrator to filter out the signal. By measuring the offsets of the individual channels, the required DC corrections can be added to their outputs to eliminate the mismatches in the digital domain. This algorithm can be applied in the factory, in the foreground or in the background. This is shown in Figure 9.59 for a two-way interleaved ADC, where the correction is done by removing the offsets from both channels. This technique, however, does not differentiate between the offset and a DC input signal.

Chopping can be employed to preserve the DC input from being removed [26] such that the only energy at DC would be from the ADC offsets. This is shown in Figure 9.60. Since chopping is applied on the analog input, it must be applied again on the digital output to undo its effect on the input signal. This chopping will convert a DC input into a square wave or a random signal depending on the code used for chopping. Therefore, the DC that remains is the DC that is added after the chopping, which is the DC offset.

9.4.2 Gain mismatch calibration

The gain of the ADC can be estimated from its digital output using multiple approaches. One method is to estimate the average power of the output signal for each channel. A mismatch in the average power between the channels is considered

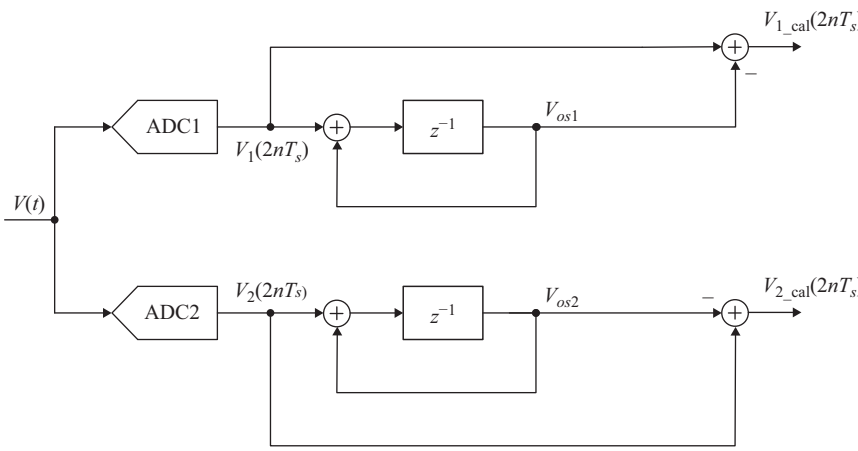


Figure 9.59 A block diagram of an offset mismatch calibration in a two-way interleaved ADC

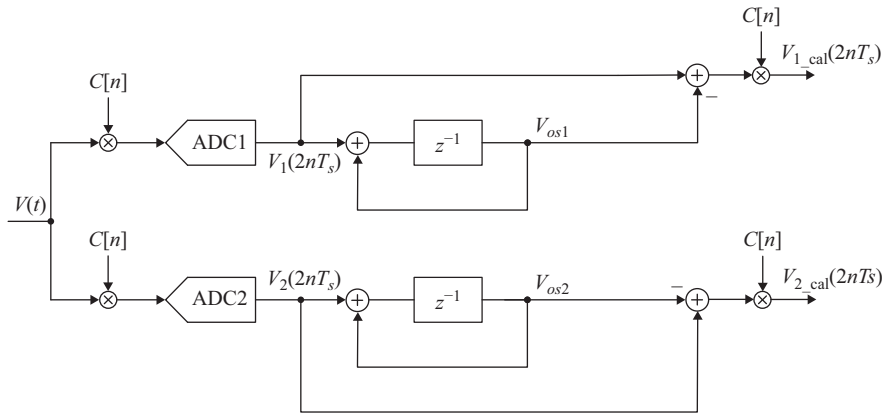


Figure 9.60 A block diagram of an offset mismatch calibration in a two-way interleaved ADC that employs chopping. © 1998, 2002 IEEE. Reprinted, with permission, from References 26 and 27

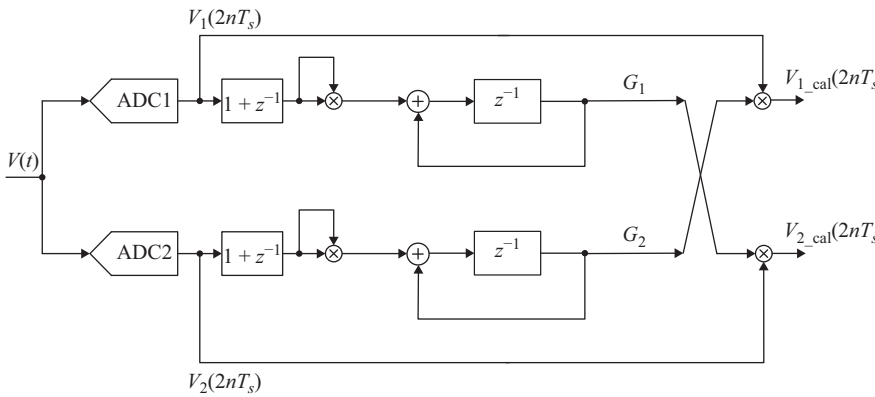


Figure 9.61 A conceptual representation of gain mismatch calibration in a two-way interleaved ADC

a gain mismatch and is corrected using a digital multiplier. This is shown in Figure 9.61. The average power estimation can be done using a squarer or an absolute value estimate of the digital output.

If the input signal's frequency is $f_s/2M$, where M is the number of channels and f_s is the interleaved ADC sampling rate, this will be the Nyquist frequency of each channel. The digital power estimate of this Nyquist signal changes with the sampling instant (phase) even in the absence of mismatches, and hence can confuse the algorithm and lead to misleading results. Therefore, it is important to filter out the energy at the Nyquist frequency before the power estimation. This is performed in Figure 9.61 using the $(1 + z^{-1})$ filter, which has a zero at $z = -1$. This corresponds

to a zero at the Nyquist frequency of the individual channels, which acts as a notch filter to remove or attenuate this problematic signal. Once the gain of each channel is estimated, it can be used to normalize the other channels as shown conceptually in Figure 9.61.

Another method for calibrating the gain mismatch involves multiplying the interleaved digital output by $(-1)^n$ [26]. As discussed in Chapter 8, the gain mismatch creates an image of the fundamental signal (f_{in}) that is located at $f_s - f_{in}$ and is in phase with the fundamental. Therefore, this multiplication of the signal by its chopped version produces a DC component that is proportional to the gain mismatch. This can be shown by analyzing the effect of chopping, given by

$$V_{ch}[n] = (-1)^n V[n] \tag{9.52}$$

where $V_{ch}[n]$ is the chopped version of the interleaved signal, and $V[n]$ is the original interleaved signal. In the frequency domain, this gives

$$V_{ch}(e^{j\omega}) = V(e^{j(\omega-\pi)}) \tag{9.53}$$

which in the continuous time domain gives

$$V_{ch}(\omega) = V(\omega - \omega_N) \tag{9.54}$$

where ω_N is the Nyquist angular frequency. Therefore, chopping moves the signal to the location of its image and the correlation between the signal and its chopped version gives an output that is proportional to the gain mismatch ΔG_c. That is,

$$\Delta G_c[n+1] = \Delta G_c[n] + \mu \times V[n] \times V_{ch}[n] \tag{9.55}$$

As shown in Figure 9.62, the interleaved signal $V[n]$ in (9.55) is obtained by adding the outputs from the individual channels after up-sampling and delaying one of the channels by one sample. In addition, a notch filter $(1 + z^{-2})$ at $f_s/4$ is used to filter out the signal at $f_s/4$ that would cause misleading gain estimates as mentioned earlier. Unlike the filtering performed in Figure 9.61, this filtering is performed

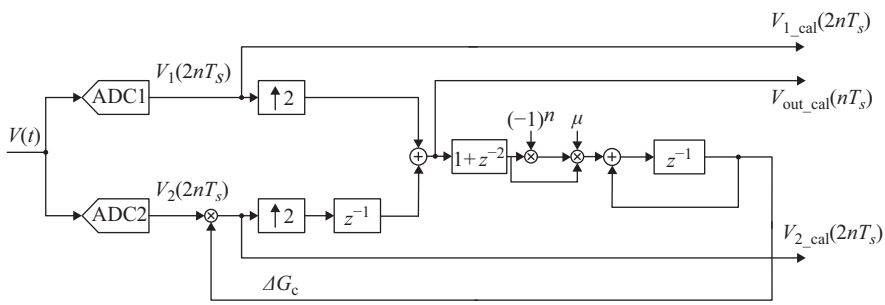


Figure 9.62 A block diagram of gain mismatch calibration in a two-way interleaved ADC [26]. © 2002 IEEE. Reprinted, with permission, from Reference 26

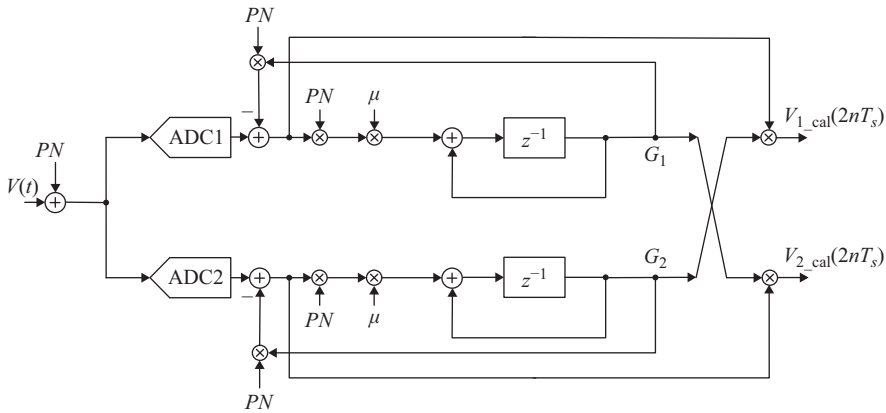


Figure 9.63 A block diagram of a correlation-based gain mismatch calibration in a two-way interleaved ADC using an injected dither signal (PN). © 1998 IEEE. Reprinted, with permission, from Reference 27

after up-sampling. Therefore, the filter needs to be in the form of $(1 + z^{-2})$, instead of the $(1 + z^{-1})$ that was used in Figure 9.61. Finally, the LMS algorithm is used to estimate the gain error and apply it to one of the channels.

These gain estimation techniques rely on the presence of an input signal for the calibration to converge. In the absence of an input signal, the calibration needs to be frozen to prevent wrong convergence. Alternatively, a signal can be injected to ensure activity and continuous calibration at all times. This calibration signal is usually in the form of a pseudo-random signal that is uncorrelated with the input signal. The LMS algorithm can be used to correlate out the signal and estimate the gain of each channel [27]. This is shown in Figure 9.63, and is given by

$$G_i[n+1] = G_i[n] - \mu \times PN[n] \times (PN[n] \times G_i[n] - V_i[n]) \tag{9.56}$$

where G_i is the gain estimate of the ith channel, $PN[n]$ is the injected PN signal, μ is the step size, and $V_i[n]$ is digital of the ith channel.

The gain mismatches can also be estimated using a reference ADC [28, 32, 35, 36], which can be used to correct for multiple kinds of mismatches. Using the LMS algorithm, the gain error of each channel can be eliminated using the formula

$$G_i[n+1] = G_i[n] - \mu \times V_i[n] \times (V_i[n] \times G_i[n] - V_{REF}[n]) \tag{9.57}$$

where G_i is the gain of the ith channel, V_i is the output if the ith channel, μ is the step size, and V_{REF} is the output of the reference ADC. Alternatively, the signs of the signals can be used instead to simplify the digital processing at the expense of convergence time. This is given by

$$G_i[n+1] = G_i[n] - \mu \times \text{sgn}(V_i[n]) \times \text{sgn}(V_i[n] \times G_i[n] - V_{REF}[n]) \tag{9.58}$$

The reference ADC can have high noise, but needs to have a gain that is independent of the input signal amplitude.

9.4.3 Timing mismatch calibration

The background calibration of the timing mismatches in interleaved ADCs is a challenging calibration problem. Since the effect of timing mismatches degrades as the input frequency increases, this problem becomes significantly more challenging for high input frequencies and RF sampling.

Good layout practices are usually employed to improve the timing mismatches. Proximity of the sampling networks of the different channels helps with the matching. In addition, using a common clock whenever possible helps reduce the mismatch errors further. However, with the best effort, the timing mismatch tend to be in the order of a few hundred femto-seconds or worse. This typically results in SINADs that are in the order of 50 dB for an input signal of about 2 GHz. It is important to note that a systematic or fixed timing mismatch can be corrected by applying a fixed adjustment to the sampling clock. However, mismatches that change with temperature, supply, sampling rate, or aging are more problematic as they require algorithms that operate in the background and adaptively track the timing changes. The timing correction can be done in the analog or digital domains.

9.4.3.1 Analog correction

In analog correction, the sampling clock passes through programmable delays that adjust the sampling instant of the different channels to eliminate the timing mismatches. This is shown conceptually in Figure 9.64. The delays can be controlled by switchable small load capacitances in the clock path. Alternatively, the threshold level of the sampling switch, or the level of its clock, can be adjusted in order to achieve finer timing adjustments [39].

9.4.3.2 Digital correction

Digital correction relies on interpolation to deduce the expected values of the samples at the correct time instants in-between the available samples. Digital filtering is typically used to perform the interpolation and correction. This is shown in Figure 9.65 and an example is shown in Figure 9.66 [26]. The discrete-time nature of the digital output creates fundamental limitations on the bandwidth and frequency of the signals whose timing mismatch can be corrected in the digital domain. Increasing the complexity of the interpolator helps, but limitations will

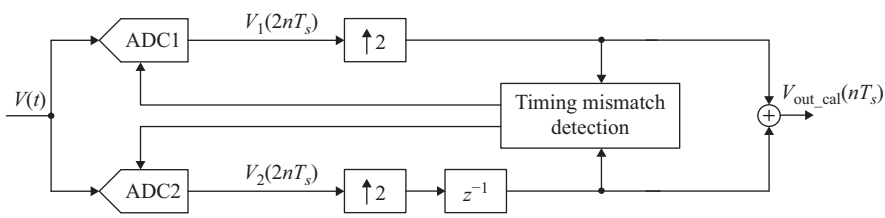


Figure 9.64 A block diagram of a timing mismatch calibration in a two-way interleaved ADC using digital detection and analog correction. © 2011 IEEE. Reprinted, with permission, from Reference 31

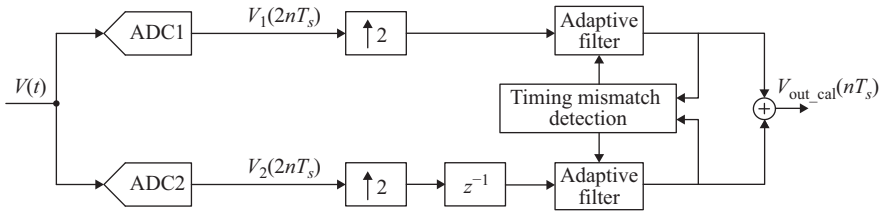


Figure 9.65 A block diagram of a timing mismatch calibration in a two-way interleaved ADC using digital detection and digital correction. © 2011 IEEE. Reprinted, with permission, from Reference 31

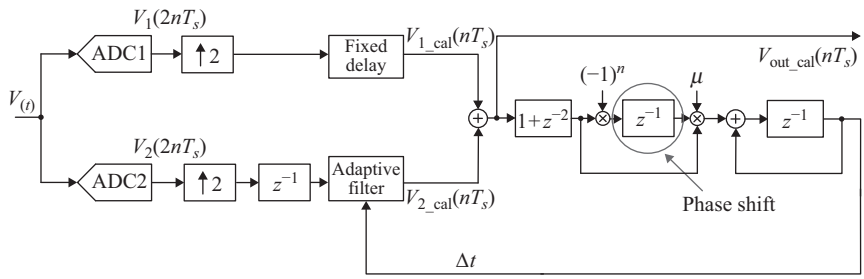


Figure 9.66 A block diagram of timing mismatch calibration in a two-way interleaved ADC using the LMS algorithm and an adaptive filter for correction. © 2002 IEEE. Reprinted, with permission, from Reference 26

still exist. Therefore, analog correction of timing mismatches is often preferred to avoid the complexity and limitations of the digital timing correction [26].

Aside from the correction, the main challenge in the timing mismatch calibration is the background estimation of the mismatch without disrupting the normal operation of the ADC. There are several approaches, which are discussed in the following sections.

9.4.3.3 Estimation using chopping

An example is shown in Figure 9.66, where the correlation between the interleaved signal and a chopped version of it is used to estimate the timing mismatch. This is the same methodology used for the gain mismatch correction of Figure 9.62, where the LMS algorithm is employed to estimate the timing mismatch error. An important difference between calibration of the timing mismatch in Figure 9.66 and the gain mismatch in Figure 9.62 is the phase shift introduced to the chopped signal. This is marked by the circle in Figure 9.66. The timing mismatch causes an image at the same location as the image due to the gain mismatch. However, since the effect of the timing mismatch depends on the slope of the signal, the image is 90° out of phase with the signal. This means that the chopped signal will be

orthogonal to the image signal, and hence their product will average to zero. A 90° phase shift is needed for the chopped signal, which requires a Hilbert transform [26]. In this implementation, this is approximated by the delay (z^{-1}), which shifts the phase by 90° near $f_s/4$. However, this $f_s/4$ frequency is the location of the zero created by the notch filter ($1 + z^{-2}$). Moreover, the phase shift created by the delay (z^{-1}) will approach zero at low frequencies and 180° near $f_s/2$. However, near $f_s/4$, some energy will remain and be phase shifted adequately by 90° to give an estimate of the timing mismatch. More elaborate filters can be used to achieve more accurate phase shifting and better detection.

9.4.3.4 Estimation using reference ADC

A reference ADC that samples the input at the same instants as the individual channels can be used to detect the timing mismatch. The LMS algorithm uses the timing error and the signal slope to estimate the timing mismatch. An additional resistance ΔR is added in the input network of the reference ADC. As shown in Figure 9.67 [32], the outputs of the two channels are subtracted to give the error *e*. The slope *D* is estimated using one of the channels and the reference channel that has the additional resistance ΔR, which is given by

$$D = V_2 - V_R \tag{9.59}$$

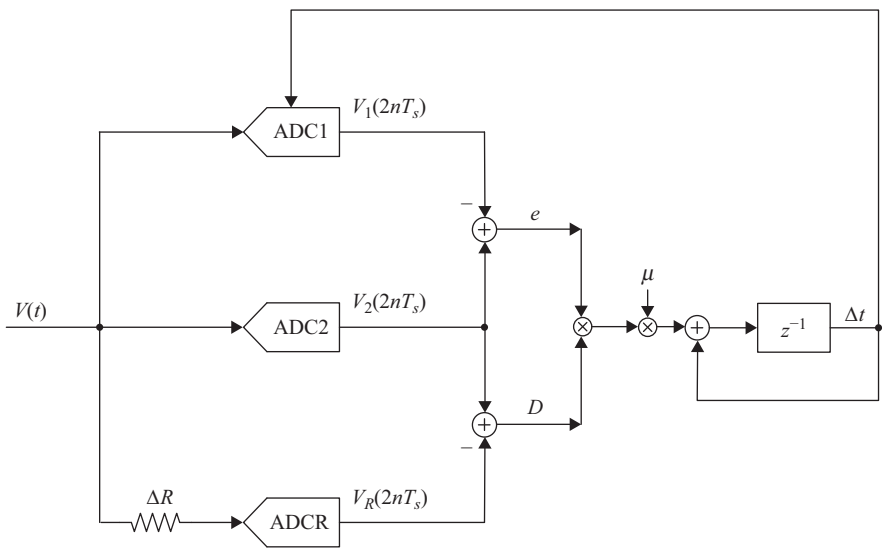


Figure 9.67 A block diagram of a timing mismatch calibration in a two-way interleaved ADC using the LMS algorithm and reference ADC. The correction is done in the analog domain by controlling the timing of one of the ADCs [32]. © 2013 IEEE. Reprinted, with permission, from Reference 32

The error e is given by

$$e = V_1 - V_2 \tag{9.60}$$

The LMS algorithm is implemented by

$$\Delta t_{i+1} = \Delta t_i + \mu \times e \times D \tag{9.61}$$

where Δt is the estimate of the timing mismatch, and μ is the algorithm's step size. The timing mismatch estimate is fed back to one of the ADCs, in order to fix its timing. This feedback loop will converge to a solution where the timing mismatch is minimized.

9.4.3.5 Estimation using cross-correlation

The output of each ADC is cross-correlated with an additional ADC (ADCC) as shown in Figure 9.68 [31]. The feedback loops converge to a point that maximizes the cross-correlations. The reference ADC can be low resolution (even one bit) and can be slow, as long as its sampling instant coincides regularly with each of the ADC channels. If the reference is clocked at the full speed f_s, it will be able to coincide with all the channels, which are running at f_s/M each. However, designing such a fast ADC can be quite challenging. Alternatively, the correlation ADC can

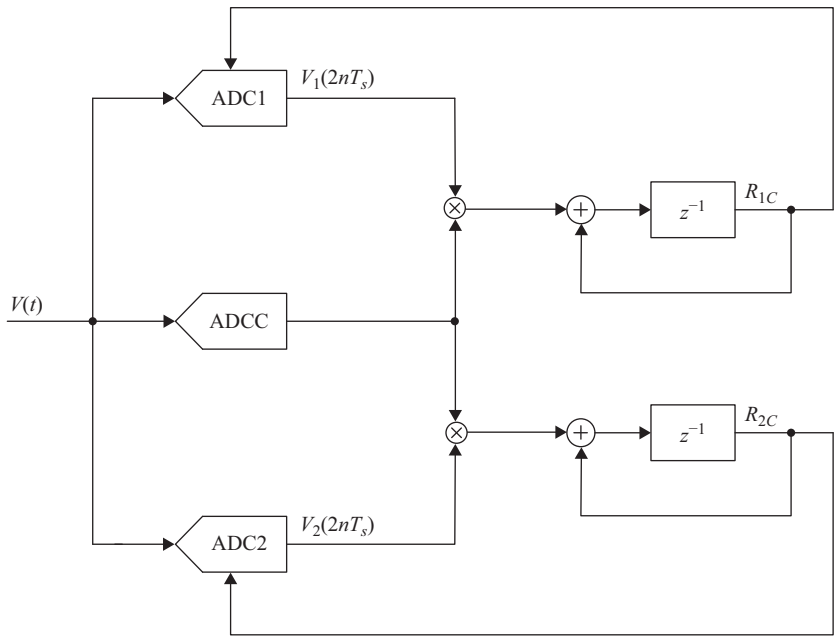


Figure 9.68 A block diagram of a timing mismatch calibration in a two-way interleaved ADC using cross-correlation. The correction is done in the analog domain by controlling the timing of one of the ADCs. © 2011 IEEE. Reprinted, with permission, from Reference 31

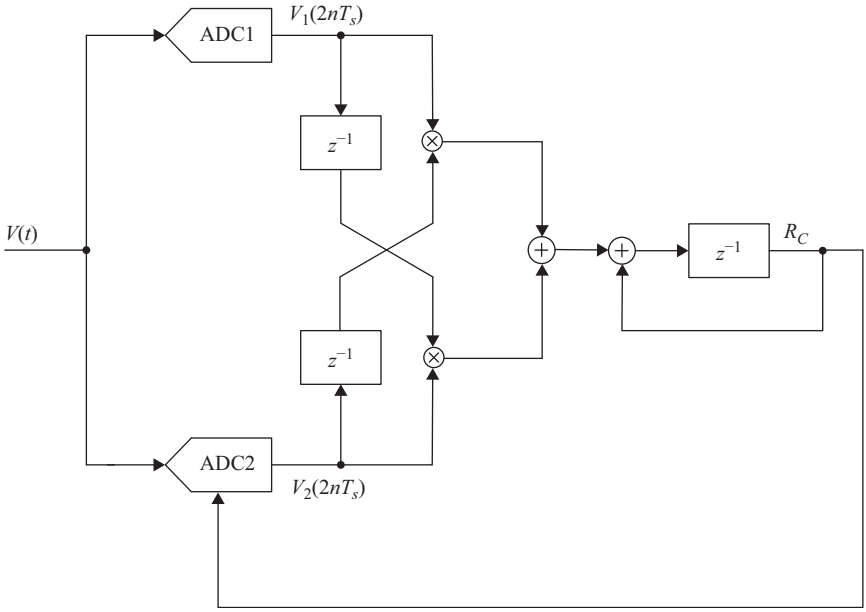


Figure 9.69 A block diagram of a timing mismatch calibration in a two-way interleaved ADC using cross-correlation. The correction is done in the analog domain by controlling the timing of one of the ADCs. © 2013 IEEE. Reprinted, with permission, from Reference 40

be clocked at a lower rate f_s/K, such that the greatest common divisor between M and K is 1. For example, that condition is satisfied if M is equal to 8 and K is equal to 9. In this case, clocking the correlation ADC at $f_s/9$ would be adequate, as it ensures that its sampling time will cyclically coincide with each of the other channels in a periodic fashion.

An alternative cross-correlation approach that does not require an additional ADC is described is shown in Figure 9.69 [40]. In this approach, the outputs of the two channels are cross-correlated after a delay of one cycle. These delays are necessary to circumvent the fact that the signal and its image are orthogonal, and hence their cross-correlation would have averaged to zero even in the presence of timing mismatch. Therefore, by adding the delays, the loop converges only when the cross-correlation is zero and the mismatch is minimized.

9.4.3.6 Estimation using split ADC

In this approach each of the interleaved ADC cores is broken down into two halves. The differences (errors) between the two halves are used as estimates of the mismatch errors [29, 30]. Shuffling the halves will exercise all combinations, and the LMS algorithm can be used in a manner similar to the reference ADC method. Intuitively, it is clear that an extra half-channel will be needed to allow for all combinations.

As discussed earlier, the split ADC method has analog and digital overhead. However, it has the additional advantage of being compatible with channel randomization, which is discussed in Chapter 8 and later in this chapter.

9.4.4 Other mismatches

In addition to offset, gain, and timing mismatches, interleaved ADCs may suffer from other sources of error, such as bandwidth and non-linearity mismatches. The bandwidth mismatch causes phase mismatch similar to that of the timing mismatch. However, it also causes frequency-dependent gain mismatch that can be very problematic. Examples of techniques to address the bandwidth mismatch are described in References 34 and 35. In practice, designers may prefer to increase the analog bandwidth such that the effect of its mismatch is negligible.

The mismatch in the ADC non-linearity is another source of error that is often left uncorrected. It is usually not a dominant source of error if the harmonic distortion level is low. An example of the research done to calibrate this kind of mismatch using the reference ADC approach is described in Reference 36.

9.4.5 Randomization

In order to improve the spurs due to any residual mismatch errors after calibration, channel randomization can be employed as discussed in Chapter 8. The order of the channel usage is randomized to avoid the periodic pattern that creates the interleaving spurs. This spreads the mismatch spur energy into the noise floor, which improves the linearity and SFDR of the ADC. Unlike calibration, randomization does not fix the mismatches or improve the SNR or SINAD. It merely disperses the spur energy into the noise floor without fixing their cause.

In order to effectively randomize the channels, at least one additional channel is required. This can be intuitively understood if we note that the same channel cannot be used to handle two samples that are spaced less than M samples apart, where M is the number of interleaved channels. This is the speed limitation that necessitated interleaving in the first place. For example, if we have two channels, the randomization cannot be achieved without an additional third channel, because none of the two channels is fast enough to handle two successive sample. This represents a significant overhead.

One of the issues with randomization is that it spreads all of the mismatch spurs' energy in the noise floor, even those that might have existed in benign locations and hence were practically harmless. For example, in two-way interleaving, the spur due to the offset mismatch exists at the Nyquist frequency ($f_s/2$), which may not be a problem for most applications. However, randomization spreads this energy in the noise floor, which makes it imperative to minimize the offset mismatch before randomization. In addition, since the channel shuffling is not completely random, it may create "humps" and "coloring" in the noise floor as discussed in Chapter 8. Another limitation of randomization is the memory effects that can be exacerbated when changing the channel ordering as is done in randomization, thereby causing a degradation in the SINAD and NSD. These memory effects may have to be calibrated before randomization as shown in References 12, 25, and 34.

9.5 Conclusion

In this chapter, we discussed some of the advanced calibration techniques used in pipelined and time-interleaved ADC. These techniques represent the state-of-the-art in digitally assisted ADCs. They enable higher sampling rates, higher performance, lower power consumption, and better integration in fine lithography CMOS processes. It is important to note that the field of digitally assisted converters is an active area of research with developments and breakthroughs occurring at a fast pace. This chapter is meant to present a snapshot of the state of the art with some of the most effective techniques. The common theme is that there are a lot of limitations and active problems to solve. The field of digitally assisted converters is nowhere where it needs to be. So, analog designers should not worry about being obsolete yet.

Problems

1. Using a behavioral modeling language, build a model of a 2-stage pipelined ADC with 3 bits in every stage and a 3-bit back-end flash. The full-scale is 2 V. With 1-bit redundancy, what is the resolution of the whole ADC? What are the inter-stage gain values? Plot an FFT of the digital output using a unity amplitude sine wave with 100 MHz frequency.
   (a) If the inter-stage gain has an error of 1%, plot the FFT and INL of the digital output.
   (b) Apply an IGE correction in the digital domain, and plot the FFT and INL of the digital output.
   (c) Apply an IGE correction using code-dependent addition, and plot the FFT and INL of the digital output.
   (d) Comment on the difference between the plots of parts (b) and (c)?
2. Add a random signal (dither) to the input of the pipeline in previous problem with amplitude equal to -24 dBFS, and subtract it from the digital back end. Plot the FFTs and INLs of the previous problem with the dither added.
3. Model a 4-bit MDAC using SPICE or a behavioral modeling language.
   (a) Investigate the amplifier settling of (9.25) and (9.27), and correcting it using a gain term.
   (b) Force the initial voltage to be non-linear using the formula $v_{initial} = 0.01v_{in}^3 + 0.01v_{in}^5$ or any other non-linear formula. Investigate the effect of the initial condition on the output when the settling time is adequate and inadequate.
4. Given the similarities and differences between the SAR and pipelined ADCs, how do they compare in terms of digital assistance? Discuss how the feedback nature of a SAR ADC compare with the feed-forward pipelined ADC? What kind of errors can be calibrated digitally in a SAR ADC?
5. Discuss if and how the correlation-based calibration algorithm can be applied to a SAR ADC.

6. Simulate the LMS algorithm and apply it to a 1-bit dither signal combined with a sinusoidal signal, in order to accurately subtract the dither from a sinusoidal signal. Investigate the impact of the dither amplitude, the sine wave amplitude, and the step size μ on the results.
7. Simulate the MDAC of Figure 9.21 using a SPICE simulator. Using the summing node voltage to correct the inter-stage gain error of the output.
8. Using a behavioral modeling language, model the IGE calibration method using a reference ADC and the LMS algorithm. Apply it to the pipeline of Problem 1. Could you apply the same method to a SAR ADC with the same resolution?
9. Using a behavioral modeling language, model the IGE calibration method using the split-ADC method and the LMS algorithm. Apply it to the pipeline of Problem 1. Could you apply the same method to a SAR ADC with the same resolution?
10. For the pipeline model of Problem 1, apply the following non-linearity to the inter-stage amplifier: $V_{out} = 0.99\,V_{in} - 0.01V_{in}^3$. How does the non-linearity impact the output FFT? Apply gain correction, piece-wise linear correction and polynomial correction, and plot the output FFT in every case.
11. Using a behavioral modeling language, model the effects of offset and gain mismatch in a 2-way interleaved ADC. Model the algorithms described in Figures 9.60 and 9.61 for the offset and gain mismatch calibrations, respectively. Can we extend this approach to 4-way interleaving? If yes, draw the block diagram.
12. Using a behavioral modeling language, model the algorithms described in Figures 9.62 and 9.66 for the gain and timing mismatch calibrations, respectively. Can we extend this approach to 4-way interleaving? If yes, draw the block diagram.
13. Using a behavioral modeling language, model the algorithms described in Figures 9.67 and 9.68 for timing mismatch calibration. Can we extend this approach to 4-way interleaving? If yes, draw the block diagram.
14. What is the magnitude of the inter-stage gain causing the INL in Figure 9.5? What are the most likely causes of the IGE error?
15. What is the magnitude of the inter-stage gain causing the INL in Figure 9.6? What are the most likely causes of the IGE error? What could be causing the breaks within each sub-range? Discuss the possible causes among the following:
    (a) Capacitor mismatch
    (b) Amplifier open loop error
    (c) Reference error
16. What are the possible causes of the DAC errors in the INL of Figure 9.11? What are the magnitudes of the errors on the analog side? Note the symmetry between the two sides and the increase in the magnitude of the errors as we move away from the center.

References

[1] A. N. Karanicolas, H.-S. Lee, and K. L. Barcrania, "A 15-bit 1-Msample/s Digitally Self-Calibrated Pipeline ADC," *IEEE Journal of Solid-State Circuits*, 28, pp. 1207–1215, Dec 1993.

[2] E. Siragusa and I. Galton, "Gain Error Correction Technique for Pipelined Analogue-to-Digital Converters," *Electronic Letters*, 36(7), pp. 617–618, 2000.

[3] E. Siragusa and I. Galton, "A Digitally Enhanced 1.8-V 15-bit 40-MSample/s CMOS Pipelined ADC," *IEEE Journal of Solid-State Circuits*, 39(12), pp. 2126–2138, Dec 2004.

[4] J. Ming and S.H. Lewis, "An 8-bit 80-Msample/s Pipelined Analog-to-Digital Converter with Background Calibration," *IEEE Journal of Solid-State Circuits*, 36(10), pp. 1489–1497, 2001.

[5] A.M.A. Ali, H. Dinc, P. Bhoraskar, *et al*., "A 14b 1GS/s RF Sampling Pipelined ADC with Background Calibration," *IEEE Journal of Solid-State Circuits*, 49(12), pp. 2857–2867, Dec 2014.

[6] A.M.A. Ali and K. Nagaraj, "Background Calibration of Operational Amplifier Gain Error in Pipelined A/D Converters," *IEEE Transactions on Circuits and Systems II*, 50(9), pp. 631–634, 2003.

[7] A.M.A. Ali, A. Morgan, C. Dillon, *et al*., "A 16-bit 250-MS/s IF Sampling Pipelined ADC with Background Calibration," *IEEE Journal of Solid-State Circuits*, 45(12), pp. 2602–2612, Dec 2010.

[8] A. Panigada and I. Galton, "A 130mW 100MS/s Pipelined ADC with 69dB SNDR Enabled by Digital Harmonic Distortion Correction," *IEEE ISSCC Digest of Technical Papers*, pp. 162–163, Feb 2009.

[9] A. Panigada and I. Galton, "Digital Background Correction of Harmonic Distortion in Pipelined ADCs," *IEEE Transactions on Circuits and Systems–I: Regular Papers*, 53(9), pp. 1885–1895, Sep 2006.

[10] A. Panigada and I. Galton, "A 130mW 100MS/s Pipelined ADC with 69dB SNDR Enabled by Digital Harmonic Distortion Correction," *IEEE Journal of Solid-0State Circuits*, 44(12), pp. 3314–3328, Dec 2009.

[11] N. Rakuljic and I. Galton, "Suppression of Quantization-Induced Convergence Error in Pipelined ADCs with Harmonic Distortion Correction," *IEEE Transactions on Circuits and Systems I*, 60(3), pp. 593–602, Mar 2013.

[12] A.M.A. Ali, "Methods and structures that reduce memory effects in analog-to-digital converters," US Patent No 6,861,969, Mar 2005.

[13] J.P. Keane, P.J. Hurst, and S.H. Lewis, "Digital Background Calibration for Memory Effects in Pipelined Analog-to-Digital Converters," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 53(3), pp. 511–525, Mar. 2006.

[14] A.M.A. Ali and A. Morgan, "Calibration methods and structures for pipelined converter systems," US Patent 8,068,045, Nov 2011.

[15] A.M.A. Ali, A.C. Morgan, and S.G. Bardsley, "Correlation-based background calibration of pipelined converters with reduced power penalty," US Patent 7,786,910, Aug 2010.
[16] A.M.A. Ali, "Pipelined converter systems with enhanced accuracy," US Patent 7,271,750, Sep 2007.
[17] A.M.A. Ali, "Method and device for improving convergence time in correlation-based algorithms," US Patent 8,836,558, Sep. 2014.
[18] S. Sonkusale, J. Van der Spiegel, and K. Nagaraj, "True Background Calibration Technique for Pipelined ADC," *Electronic Letters*, pp. 786–788, 36(9), 2000.
[19] Y. Chiu, C.W. Tsang, B. Nikolic, and P.R. Gray, "Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters, *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 51(1), pp. 38–46, Jan 2004.
[20] J.A. McNeill, J. McNeill, M.C.W. Coln, and B.J. Larivee, "Split ADC" Architecture for Deterministic Digital Background Calibration of a 16-bit 1-MS/s ADC, *IEEE Journal of Solid State Circuits*, 40(12), pp. 2437–2445, Dec 2005.
[21] E. Iroaga and B. Murmann, "A 12-Bit 75-MS/s Pipelined ADC Using Incomplete Settling," *IEEE Journal of Solid-State Circuits*, 42(4), pp. 748–756, Apr 2007.
[22] B. Murmann and B.E. Boser, "A 12 b 75 MS/s Pipelined ADC Using Open-Loop Residue Amplification," *IEEE ISSCC Digest of Technical Papers*, 1, pp. 328–497, 2003.
[23] B. Murmann and B.E. Boser, "A 12 b 75 MS/s Pipelined ADC Using Open-Loop Residue Amplification," *IEEE Journal of Solid-State Circuits*, 38(12), pp. 2040–2050, Dec 2003.
[24] B. Murmann and B.E. Boser, "Digital Domain Measurement and Cancellation of Residue Amplifier Nonlinearity in Pipelined ADCs," *IEEE Transactions on Instrumentation and Measurements*, 56(6), pp. 2504–2514, Dec 2007.
[25] A.M.A. Ali and H. Dinc, "Method and device for reducing inter-channel coupling in interleaved and multi-channel ADCs," US Patent 8,471,741, Jun 2013.
[26] S.M. Jamal, D. Fu, N.C.-J. Chang, *et al*., "A 10-b 120-Msample/s Time-Interleaved Analog-to-Digital Converter with Digital Background Calibration," *IEEE Journal of Solid-State Circuits*, 37(12), pp. 1618–1627, Dec 2002.
[27] D. Fu, K.C. Dyer, S.H. Lewis, and P.J. Hurst, "A Digital Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, 33(12), pp. 1904–1911, Dec 1998.
[28] K. Dyer, D. Fu, S.H. Lewis, and P.J. Hurst, "An Analog Background Calibration Technique for Time-Interleaved Analog-to-Digital Converters," *IEEE Journal of Solid-State Circuits*, 33(12), pp. 1912–1919, Dec 1998.

[29] J.A. McNeill, C. David, M. Coln, and R. Croughwell, "Split ADC" Calibration for All-Digital Correction of Time-Interleaved ADC Errors, *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 56(5), pp. 344–348, May 2009.

[30] J.A. McNeill, M.C.W. Coln, D.R. Brown, and B.J. Larivee, "Digital Background-Calibration Algorithm for "Split ADC" Architecture," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 56(2), pp. 294–306, Feb 2009.

[31] M. El-Chammas and B. Murmann, "A 12-GS/s 81-mW 5-bit Time-Interleaved Flash ADC with Background Timing Skew Calibration," *IEEE Journal of Solid-State Circuits*, 46 (4), pp. 838–847, Apr 2011.

[32] D. Stepanović and B. Nikolić, "A 2.8 GS/s 44.6 mW Time-Interleaved ADC Achieving 50.9 dB SNDR and 3 dB Effective Resolution Bandwidth of 1.5 GHz in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, 48(4), pp. 971–982, Apr 2013.

[33] G. Léger, E.J. Peralias, A. Rueda, and J.L. Huertas, "Impact of Random Channel Mismatch on the SNR and SFDR of Time-Interleaved ADCs," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 51(1), pp. 140–150, Jan 2004.

[34] C.H. Law, P.J. Hurst, and S.H. Lewis, "A Four-Channel Time-Interleaved ADC with Digital Calibration of Interchannel Timing and Memory Errors," *IEEE Journal of Solid-State Circuits*, 45(10), pp. 2091–2103, Oct 2010.

[35] T.-H. Tsai, P.J. Hurst, and S.H. Lewis, "Bandwidth Mismatch and Its Correction in Time-Interleaved Analog-to-Digital Converters," *IEEE Transactions on Circuits and Systems-II: Express Briefs*, 53(10), pp. 1133–1137, Oct 2006.

[36] W. Liu and Y. Chiu, "Time-Interleaved Analog-to-Digital Conversion with Online Adaptive Equalization," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, 59(7), pp. 1384–1395, Jul 2012.

[37] P. Huang, S. Hsien, V. Lu, *et al*., "SHA-Less Pipelined ADC with *In Situ* Background Clock-Skew Calibration," *IEEE Journal of Solid-State Circuits*, 46(8), pp. 1893–1903, Aug 2011.

[38] M. Brandolini, Y. Shin, K. Raviprakash, *et al*., "A 5GS/s 150mW 10b SHA-Less Pipelined/SAR Hybrid ADC in 28nm CMOS," *IEEE ISSCC Digest of Technical Papers*, pp. 468–469, Feb 2015.

[39] A.M.A. Ali, H. Dinc, P. Bhoraskar, *et al*., "A 14-bit 2.5GS/s and 5GS/s RF Sampling ADC with Background Calibration and Dither," *IEEE VLSI Circuits Symposium*, pp. 206–207, 2016.

[40] B. Razavi, "Design Considerations for Interleaved ADCs," *IEEE Journal of Solid-State Circuits*, 48(8), pp. 1806–1817, Aug 2013.

[41] B.D. Sahoo and B. Razavi, "A 12-Bit 200-MHz CMOS ADC," *IEEE Journal of Solid State Circuits*, 44(9), pp. 2366–2380, Sep. 2009.