Mismatch-Shaping Successive-Approximation ADC

Matthew Coe and Un-Ku Moon
Oregon State University
moon@ece.orst.edu

Abstract

A mismatch-shaping scheme is proposed for a 16-bit successive-approximation ADC based on a simple two capacitor charge-redistribution DAC where the mismatch estimates are accurate to the first order. A digital state machine is used to control the switching sequence of the D/A conversion. A second order delta-sigma loop was found to be effective in system simulations given a 0.1% capacitor mismatch. Spectral analysis of the ADC shows dramatic improvement in total harmonic distortion as well as 87 dB SNDR (signal to noise and distortion ratio) for an oversampling ratio (OSR) of 10.

Keywords: Mismatch-shaping, Capacitor mismatch, Data converters, ADC, Successive-approximation, Switched capacitor, Delta-sigma

1. Introduction

The performance of analog integrated circuits is generally limited to the matching accuracy of their components. A recent approach to systematically solving this problem is to use digital logic to perform a shuffling of nominally matched components so that the noise introduced by component mismatch can be controlled [1][2]. The mismatch noise is filtered to reduce noise energy within the signal band and move it to higher frequencies where it can be removed. This mismatch-shaping approach has been used successfully for the internal digital-to-analog converters (DAC) of delta-sigma analog-to-digital converters (ADC) and DACs, as well as for pipeline ADCs [3][4][5]. This paper will extend the mismatch-shaping technique to a simple successive-approximation ADC.

2. Successive-approximation ADC

A successive-approximation ADC employs the binary search algorithm in order to find the digital code which most closely matches the analog input value. In order to obtain an N bit digital word the binary search algorithm must be carried out N times, requiring N clock cycles. During this time, the input voltage must not change, which requires the use of one additional clock cycle to sample and hold the input value. N+1 clock cycles are required to determine N bits, starting with the MSB (most significant bit) and continuing until the LSB (least significant bit) is determined.

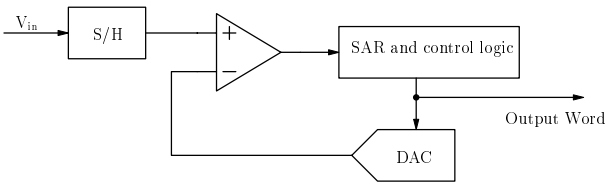

Figure 1: Block diagram of successive-approximation ADC

During the first of the N+1 clock cycles, the input voltage is sampled and held, as is the reference voltage for the conversion (refer to Figure 1 above). The reference charge is shared between two nominally matched capacitors at the beginning of each clock cycle (see Figure 2 below). In this way, the reference voltage available in each clock cycle can be half the reference value of the previous clock cycle.

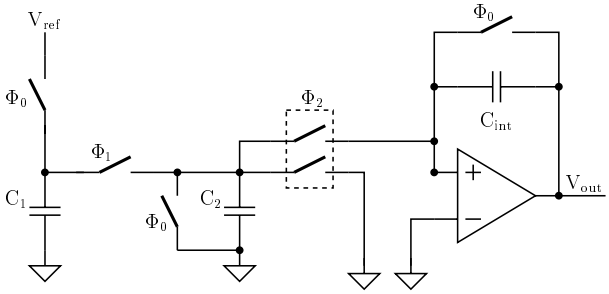

Figure 2: Two capacitor charge-redistribution DAC

The sampled input voltage is continually compared to the output of this internal DAC, which determines one bit in each of the remaining N clock cycles. In this type of ADC, the accuracy of the entire converter is mainly constrained by the accuracy of the internal DAC. Therefore, it is extremely important to maximize the accuracy of the DAC. The accuracy of this simple DAC is dependent upon the matching accuracy of the two charge-sharing capacitors C_1 and C_2.

3. Mismatch shaping

In order to shape (i.e. high-pass filter) the error resulting from the capacitor mismatch in the context of a successive-approximation ADC, we need to be able to add or extract error charge from the integration capacitor. Figure 3 shows the conceptual implementation of a circuit that meets these criteria. Capacitor mismatch error, which causes an inaccurate amount of charge to be stored on the integration capacitor, can be controlled by interchanging the roles of C_1 and C_2 during each clock cycle.

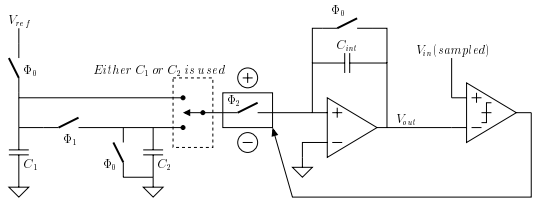

Figure 3: Conceptual representation of charge-redistribution ADC

A differential circuit representation for the conceptual ADC of Figure 3 is shown in Figure 4. Each box controlled by the sequence selector contains the set of switches that can either add or subtract charge to/from the integrating capacitors by either directly-coupling or cross-coupling the two branches of the fully differential circuit. By keeping track of the accumulated error over all previous conversion cycles, as well as the output of the comparator, the smart logic determines the switching sequence that will minimize the amount of error charge stored on the integration capacitors.

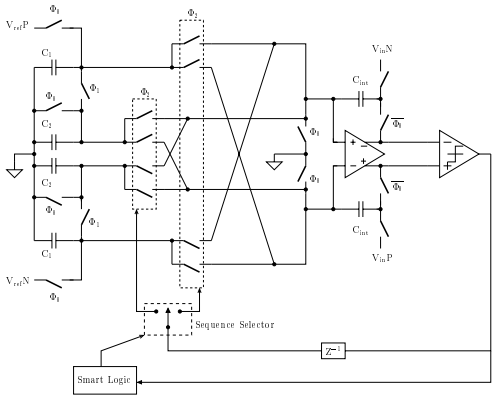

Figure 4: Differential implementation of mismatch-shaping ADC

We must be able to describe the capacitor mismatch error to successfully control it. Assuming that C_1 is slightly larger than C_2 we can normalize the capacitance values to $1+\alpha$ and $1-\alpha$, respectively. We can now determine the amount of error charge that is injected into the integrating capacitor in terms of α. The error map shown in Figure 5 describes the mismatch errors resulting in each clock cycle in terms of α. The error quantity is approximated to the first order (α^2, α^3, … terms are ignored).

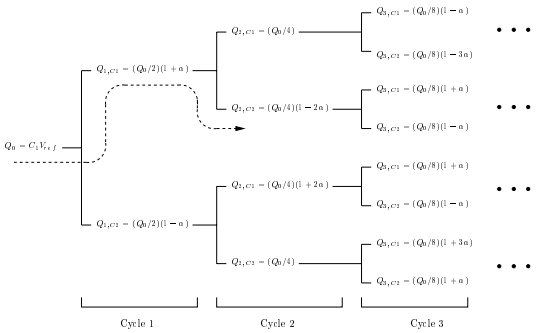

Figure 5: Error map

The example path denoted by the dotted arrow represents the switching sequence in the first two clock cycles. In this case, C_1 was used to dump charge to the integrating capacitor in the first cycle and C_2 in the second. Note that the error charge accumulated so far in the example is $(Q_0\alpha)/2 + (-Q_0 2\alpha)/4 = 0$. For each data conversion, it can be shown that the net error accumulated up to any location in the error map is fully described (to the first order) by the equation

$$Error \cong \alpha \sum_{k=1}^{N} p(k) 2^{-k} \left[t(k) - \sum_{j=1}^{k-1} t(j) \right] \quad (1)$$

where α represents the capacitor mismatch, k=1 is the MSB, $t(k) = \pm 1$ controls the choice of C_1 or C_2, and $p(k) = \pm 1$ represents the comparator output in each clock cycle. This equation is similar to that found in [2].

The smart logic block will implement the above equation to estimate the magnitude and sense of the errors committed. The logic block will attempt to minimize the total error committed in each conversion cycle and then use a fully digital second-order delta-sigma loop to shape the remaining error which will be removed by a decimation filter. The algorithm can only act on available information which, in the context of a successive-approximation ADC, will be an incomplete set of data. In the kth cycle, the information is available only up to the k-1 index. Given the latest comparator output, which defines the polarity of p(k), Equation 1 is used to determine the polarity of t(k). Thus the mismatch minimizing algorithm needs to try to optimize the choice of either C_1 or C_2 at each clock cycle. To shape the overall error (the error committed over many conversion cycles) the delta-sigma loop output

is used to select the capacitor used in the first clock cycle of each conversion cycle.

The analog portion of the system as described requires 1 opamp, 1 comparator, 2 reference generators, 6 capacitors, and 20 switches. kT/C noise analysis indicates that the minimum capacitor size, given an oversampling ratio of 10, should be 0.5 pF. The switches should be sized to have a maximum on resistance of 13 kΩ. The digital portion of the system was synthesized from structural Verilog code. The unoptimized layout of the digital circuit occupied 0.5 mm^2 when targeted to the TSMC 0.35μm process.

The system was simulated for a capacitor mismatch of 0.1% with and without the proposed mismatch shaping. The input signal was a $0.707V_{ref}$ peak-to-peak sinusoid with a frequency 32 times lower than the Nyquist rate. Figure 6 shows the uncompensated output. The simulated SNDR (signal to noise and distortion ratio) for this converter is 63dB. The response is highly tonal with the total harmonic distortion at –70dB.

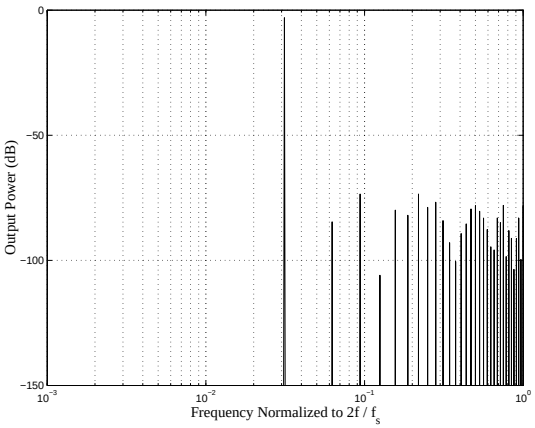

Figure 6: Uncompensated ADC output

Figure 7 shows the compensated output (prior to decimation) for an oversampling ratio of 10. The simulated SNDR improves to 87dB and tones are disbursed. Note that the noise shaping response is evident, but not as well behaved as the response of a similarly compensated DAC. In the case of the ADC, the following LSBs are undetermined when a decision is made for the MSB, resulting in error in each conversion period that cannot be well controlled. This is an inherent disadvantage for ADCs, as compared to any DAC, which would have the data/word at hand for each conversion period. The shaping of the error is hence somewhat less accurate in the case of cyclic ADCs, and the total error made is bound to be more irregular (resulting in nonlinear quantization) and larger (giving more rms noise).

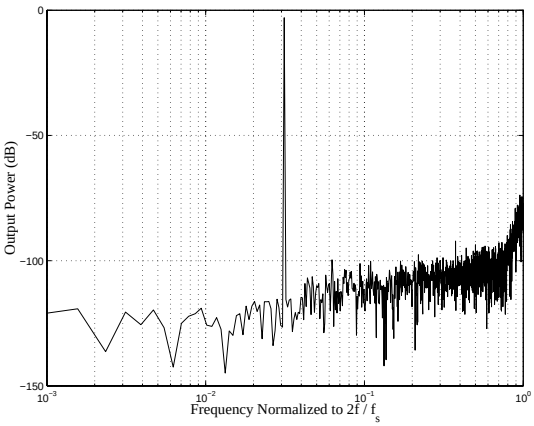

Figure 7: Compensated ADC output (undecimated)

4. Conclusions

For the mismatch-shaping successive-approximation ADC described, simulations indicate successful operation even with modest oversampling ratios. The fully digital second-order delta-sigma loop operates

with mathematical estimates of the mismatch errors, which are accurate to a first-order approximation. Higher order errors that are not accounted for are essentially negligible.

5. References

1. P. Rombouts, L. Weyten, J. Raman, and S. Audenaert, “Capacitor mismatch compensation for quasi-passive switched-capacitor DAC,” IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol. 45, pp. 68-71, January 1998.

2. L. Weyten and S. Audenaert, “Two-capacitor DAC with compensative switching,” Electronics Letters, vol. 31, pp. 1435-1436, August 1995.

3. J. Steensgaard, U. Moon, and G. C. Temes, “Mismatch shaping switching for two-capacitor DAC,” Electronics Letters, vol. 34, pp. 1633-1634, August 1998.

4. J. Steensgaard, U. Moon, and G. C. Temes, “Mismatch-shaping serial digital-to-analog converter,” Proceedings of the 1999 IEEE Symposium on Circuits and Systems, vol. 2, pp. 5-8, May 1999.

5. L. Hernandez, “Digital implementation of mismatch shaping in oversampled pipeline A/D converters,” Electronics Letters, vol. 34, pp. 616-617, April 1998.