ON THE USE OF REDUNDANCY IN SUCCESSIVE APPROXIMATION A/D CONVERTERS

Boris Murmann

Department of Electrical Engineering, Stanford University, Stanford, CA, USA
Email: murmann@stanford.edu

ABSTRACT

In practical realizations of sequential (or pipelined) A/D converters, some form of redundancy is typically employed to help absorb imperfections in the underlying circuits. The purpose of this paper is to review the various ways in which redundancy has been used in successive approximating register (SAR) ADCs, and to connect findings from the information theory community to ideas that drive modern hardware realizations.

Keywords— A/D conversion, redundancy, successive approximation, beta expansion

1. INTRODUCTION

Analog-to-digital (A/D) converters map continuous-time, continuous-amplitude signals into a discretized representation via sampling and quantization. In a typical hardware implementation, the precision of this mapping is impaired by nonidealities of the underlying electronic circuit, as for instance mismatch between nominally identical components. In practice, these nonidealities can be mitigated via a number of design techniques that can be categorized into the following groups: (1) precision analog design, (2) analog or digital calibration techniques, and (3) redundancy.

Precision analog design techniques aim at designing (or sizing) the circuit such that its precision matches the desired specifications by construction. While this approach can be practical, it sometimes causes significant overhead, for example in terms of power dissipation. To address this issue, calibrated A/D converters correct circuit imperfections by measuring the induced errors and by adjusting a correction circuit in the analog or digital domain. Introducing redundancy in the A/D conversion process is another popular solution, but it differs fundamentally from calibration in the sense that the errors are neither measured, nor corrected, but simply tolerated and rejected by the conversion algorithm. Many modern A/D converters utilize a combination of calibration and redundancy and employing redundancy is often required to make certain calibration techniques work.

To this author's best knowledge, the use of redundancy in A/D converters dates back to 1964 [1]. Since then, many variants of the idea have been proposed and used in practice. Most recently, however, there has been renewed interest in research on this topic for the successive approximation register (SAR) architecture, which has gained popularity due to its compatibility with nano-scale integrated circuit technologies [2]. As we will explain below, SAR ADCs can benefit from redundancy in a variety of intriguing ways, some of which have been discovered or applied only recently. Within this context, the purpose of this paper is to summarize the state-of-the-art in the design of SAR ADCs with redundancy.

2. IDEAL A/D CONVERSION AND BETA-EXPANSION

Ideal A/D conversion of a continuous input variable $0 \le x < 1$ can be viewed as a binary expansion of the form

$$\hat{x} = \sum_{k=1}^{N} b_k 2^{-k} \tag{1}$$

Here, $b_1, \ldots, b_N \in \{0, 1\}$ are the bits of the binary representation and $\hat{x} - x$ is the quantization error. The bits can be determined using a binary search algorithm that uses the initial guess $x_1 = 1/2$ and the recursion

$$x_k = x_{k-1} + s_k 2^{-k} \tag{2}$$

where

$$s_k = \begin{cases} +1 & x > x_k \\ -1 & x \le x_k \end{cases} \tag{3}$$

and $b_k = (s_k + 1)/2$. This process can be interpreted graphically using the decision tree shown in Figure 1 [3]. The dotted lines represent all possible paths for x_k, and the solid lines correspond to an example path for a specific input x. An important property of this conversion algorithm is that the path that leads to $\hat{x}$ is unique. This also implies that there exists a unique bit pattern for each input, and more importantly, any error in the bit decisions given by (3) will prevent us from achieving the best possible approximation.

Consider now a modification of (1) such that

$$\hat{x} = \alpha \sum_{k=1}^{N} b_k \beta^{-k} \tag{4}$$

where $1 < \beta < 2$ and $\alpha = \beta - 1$ is a scale factor that sets the full-scale range to unity. As explained in [4], this "beta-expansion" [5] contains redundancy, in the sense that multiple bit patterns can lead to an approximation within a certain error bound. This is illustrated graphically in Figure 2. Here, $\beta = 2^{3/4}$ and the algorithm uses $N = 4$ steps. After the last step, the obtained approximation is digitally mapped onto the closest level of an ideal 3-bit A/D converter.

As we can see from the pattern of all possible paths, there are multiple trajectories that terminate at the same $\hat{x}$. This means that certain decision errors can be absorbed without affecting the conversion result. For instance, as shown using the bold dashed line, a decision error in the

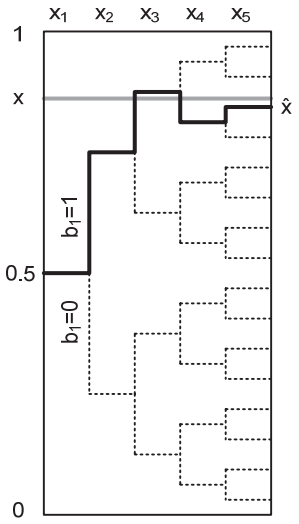

Figure 1: Graphical illustration of ideal sequential A/D conversion with 4-bit resolution. The algorithm resolves 4 bits using 4 steps (no redundancy).

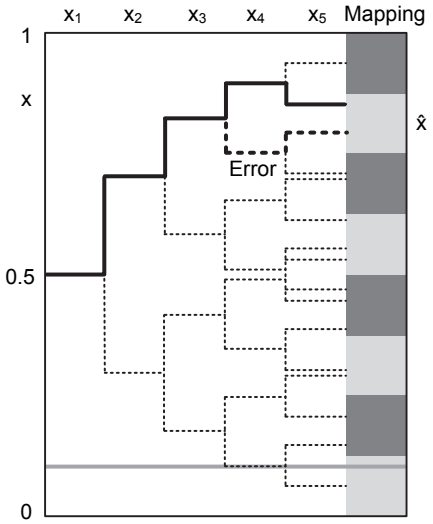

Figure 2: Graphical illustration of sequential A/D conversion with redundancy. The conversion resolves 3 bits using 4 steps.

third step will still lead to the correct conversion result. The cost for this error tolerance is two-fold: (1) the number of steps must be larger than the number of bits that are being resolved and (2) extra hardware is needed to map the raw bit pattern into the usual binary output.

The magnitude of the tolerable decision error in each step can be estimated by computing the difference between the current bit weight and the sum of all remaining weights. For example, consider the above-described converter with $\beta = 3/4$, resolving three bits in 4 steps. The first bit enters (1) with a weight of $\beta^{-1} = 0.595$. The sum of the remaining weights is $\beta^{-2} + \beta^{-3} + \beta^{-4} = 0.689$. As long as a close approximation (within the quantization error) is reachable by the sum of the last three weights, an error in the first bit decision will be inconsequential. The same idea applies to later bit decisions, with the main difference that the sum of the remaining weights, and therefore the correction range, is decreasing with each step. Detailed calculations of the tolerable decision errors for a variety of bit configurations are tabulated in [3]. For example, in a 10-bit, 12-step ADC, the tolerable decision errors normalized to the quantization step size are: 90, 51, 28, 16, 9, 5, 3, 1, 1 and 0 for all remaining decisions.

In recent literature, it is often overlooked that the concept of using $\beta < 2$ (or "radix < 2") has been used in hardware implementations long before detailed mathematical results – such as Daubechies' 2002 paper [4] – were available. In the context of SAR ADCs, using a reduced radix was first proposed in 1981 [6], and further popularized in [7], [8]. The latter reference is sometimes cited as the "first" even though it appeared more than twenty years after the original idea. What is even less known is that the original idea of using redundancy dates back to 1964 [1]. In this work, redundancy was introduced not by using $\beta < 2$, but instead by creating extra decision levels in (3). We will summarize this idea and other approaches that have evolved in the context of hardware design in the following section.

3. REDUNDANCY IN TODAY'S DESIGNS

A. Radix=2 Designs with Redundant Decision Levels

In the original work of [1], one extra decision level was used to create overlapping trajectories as in Figure 2. The design resolved two bits per step, which normally requires three decision levels. The added fourth decision level allowed the algorithm to absorb large comparison errors, enumerated in more detail in [1].

The idea of introducing redundant decision levels is still used today, and most widely exploited in pipeline ADCs [9], which can be described by a set of equations similar to (1) – (4). In this context, designers speak of a "1.5-bit" quantizer when one extra level is added to (3), since $\log_2(2+1) = 1.58$. The concept is also called "redundant signed digit (RSD)" conversion [10], akin to the redundant binary number system sometimes used in digital adders. The 1.5-bit concept has been re-introduced recently in SAR conversion, as described in [11].

B. Radix=2 Designs with Redundant Steps

Redundancy primarily helps absorb errors in the bit decisions (equation (3)). However, it is important to distinguish between two different ways in which such errors may be introduced. The first and most obvious is a direct error in the evaluation of the inequality. The second possibility is an error in x_k, which may occur in hardware realizations due to the finite speed at which (2) is computed ("DAC settling error" – see also Section IV). Such errors can be tolerated by designs with redundant decision levels, but it was shown in [12] that the introduction of one redundant step (and no extra decision levels) is also sufficient. The idea exploits the exponential nature of the settling errors and the fact that the impact of the errors reduces from cycle to cycle.

The work of [13] uses one redundant step in an even more intriguing way to mitigate the impact of random decision errors ("thermal noise"). It is noted that at most two out of all decisions (equation (2)) must resolve a very small difference that may be corrupted by noise. One of these critical decisions must be the last one, and the other one can be in any prior cycle. As shown in [13], this latter error can be elegantly corrected by introducing one extra conversion step. In hardware, this feature is then exploited by running all but the last two conversions with a very low-energy (but noisy) comparator, and expending significant energy to overcome noise only in the final decisions.

C. Radix<2 Designs

As discussed previously, the idea of using a radix of less than two ("beta-expansion") goes back to 1981 and is still used today [3]. One common challenge to this form of redundancy is that the radix must be known precisely to construct the proper conversion result. Of course, the radix must also be precisely set in radix = 2 topologies, but here this is naturally achieved by employing integer multiples of well-matched and nominally identical integrated circuit components.

In practice, the radix is typically measured using some form of calibration. In [14], it was shown that the radix (β) can be estimated by comparing the output of the converter for the inputs x and 1-x. In a practical realization, such a calibration step would have to be performed with controlled input signals, thus interrupting the normal conversion operation. Such an approach is commonly called foreground or start-up calibration.

Reference [15] describes a method by which the radix can be continuously measured ("in the background") without interrupting normal conversion. The method is based on running two conversions of the same input with different additive perturbations. Based on the difference between the two results and its ideal value, an LMS loop updates the radix in the digital bit mapping until convergence is achieved. At first glance it seems expensive to run extra conversions for the sole purpose of measuring of the radix. However, the two measurements allow averaging of the thermal noise and hence the calibration is energy neutral (to first order).

With redundancy and radix calibration in place, the only remaining precision requirement in the hardware is that the computation of (2) must be sufficiently linear. However, as pointed out in [16], even nonlinearity could be compensated through calibration. Still, in typical realizations of SAR ADCs, where the computation of (2) relies on high-quality passive components, such issues have not yet proven to be significant. The situation is different in pipeline ADCs, where digital linearization techniques have been proposed to combat nonlinear effects in passives [17] and amplifiers [18].

5. A CLOSER LOOK AT DAC SETTLING ERRORS

Figure 3 shows a conceptional block diagram of a typical SAR ADC. The comparison level x_k in (2) is generated by a D/A converter, which is controlled by digital circuitry that implements the approximation

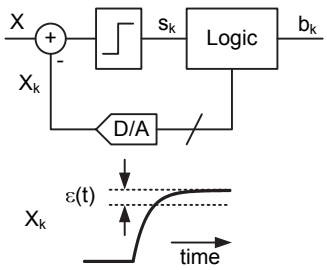

Figure 3: Conceptual block diagram of a SAR ADC.

algorithm. Since the speed of practical D/A converters is finite, x_k is usually not fully settled at the time the bit decision is made and this can lead to bit errors. Fortunately, and as already mentioned above, such errors are inconsequential with sufficient redundancy in place. The DAC error is indistinguishable from errors made in the quantizer itself. Especially for high-speed designs, this feature is being heavily exploited in today's designs [12].

In this context, it is interesting to invoke a comparison to pipeline ADCs, which also employ redundancy in their underlying quantizers. Figure 4(a) shows a block diagram of a pipeline ADC, which can be conceptually thought of as a "loop-unrolled" version of a SAR ADC. In other words, instead of performing (2) sequentially, the hardware is parallelized and pipelined to increase throughput. An interesting and important difference between the shown pipelined architecture and a SAR ADC is that DAC settling errors *cannot* be absorbed through redundancy. The reason is that the settling error is sampled and forward-propagated such that it results in a direct error that has no further time to decay. A clever workaround for this problem was only proposed recently in [19]. As shown in Figure 4(b), this design uses a feedforward path, which, after some delay injects a precise version of the fully settled DAC signal into the following stage. The feedforward path has extra time to settle, since its output is only needed after the succeeding stage's quantizer and DAC have processed their inputs. With this modification, the pipelined architecture can potentially benefit as much from redundancy as a SAR ADC, and high conversion rates are possible with relatively slow sub-D/A converters and amplifiers.

6. CONCLUSION

This paper has reviewed the state-of-the-art and historical background on the use of redundancy in SAR A/D converters. A general observation for most of the work in this area is that the practical exploration of ideas typically occurs well before the underlying mathematics has been thoroughly described. The development of a holistic theoretical framework that captures all variants of redundancy would be beneficial to the field.

REFERENCES

[1] T. C. Verster, "A Method to Increase the Accuracy of Fast-Serial-Parallel Analog-to-Digital Converters," *IEEE Trans. on Electronic Computers*, vol. EC-13, no. 4, pp. 471–473, Aug. 1964.

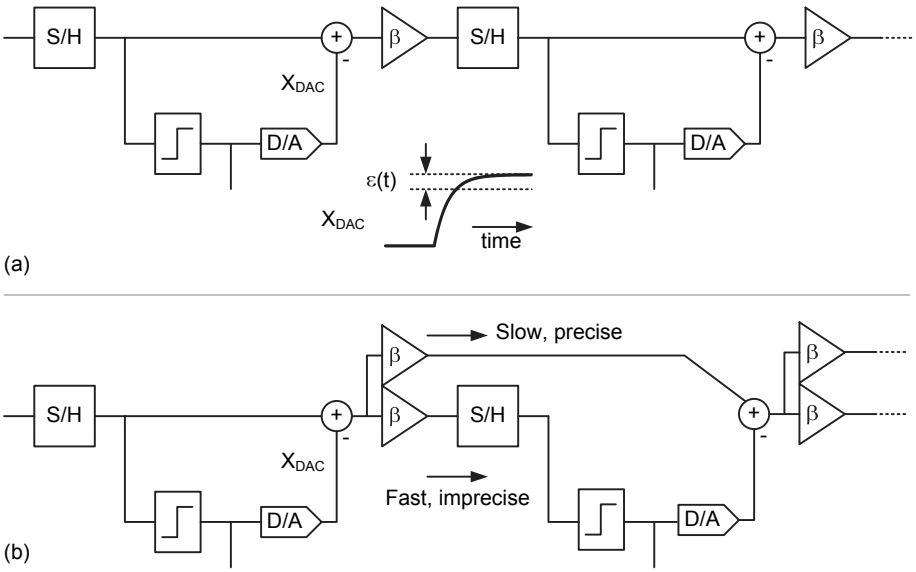

Figure 4: (a) Conventional pipeline ADC. (b) Addition of a feedforward path that allows the absorption of DAC settling errors in the converter's redundancy.

[2] D. Draxelmayr, "A 6b 600MHz 10mW ADC array in digital 90nm CMOS," in *ISSCC Dig. Techn. Papers*, 2006, pp. 264–265.

[3] T. Ogawa, H. Kobayashi, Y. Takahashi, N. Takai, M. Hotta, H. San, T. Matsuura, A. Abe, K. Yagi, and T. Mori, "SAR ADC Algorithm with Redundancy and Digital Error Correction," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E93.A, no. 2, pp. 415–423, Feb. 2010.

[4] I. Daubechies, R. DeVore, C. S. Gunturk, and V. Vaishampayan, "Beta expansions: a new approach to digitally corrected A/D conversion," in *Proc. IEEE ISCAS*, 2002, pp. 784–787.

[5] W. Parry, "On the Beta-Expansions of Real Numbers," *Acta Math. Acad. Sci. Hungar.*, vol. 11, pp. 401–416, 1960.

[6] Z. Boyacigiller, B. Weir, and P. Bradshaw, "An error-correcting 14b/20µs CMOS A/D converter," in *ISSCC Dig. Techn. Papers*, 1981, pp. 62–63.

[7] D. Draxelmayr, "A Self Calibration Technique for Redundant A/D Converters Providing 16b Accuracy," in *ISSCC Dig. Techn. Papers*, 1988, pp. 204–205.

[8] F. Kuttner, "A 1.2V 10b 20MSample/s non-binary successive approximation ADC in 0.13µm CMOS," in *ISSCC Dig. Techn. Papers*, 2002, pp. 136–137.

[9] S. H. Lewis, H. S. Fetterman, G. F. Gross, R. Ramachandran, and T. R. Viswanathan, "A 10-b 20-Msample/s analog-to-digital converter," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 3, pp. 351–358, Mar. 1992.

[10] B. Ginetti, A. Vandemeulebroecke, and P. Jespers, "RSD cyclic analog-to-digital converter," in *Symp. VLSI Circuits Dig.*, 1988, pp. 125–126.

[11] R. Vitek, E. Gordon, S. Maerkovich, and A. Beidas, "A 0.015mm2 63fJ/conversion-step 10-bit 220MS/s SAR ADC with 1.5b/step redundancy and digital metastability correction," in *Proc. IEEE Custom Integrated Circuits Conference*, 2012, pp. 1–4.

[12] C.-C. Liu, S.-J. Chang, G.-Y. Huang, Y.-Z. Lin, C.-M. Huang, C.-H. Huang, L. Bu, and C.-C. Tsai, "A 10b 100MS/s 1.13mW SAR ADC with binary-scaled error compensation," in *ISSCC Dig. Techn. Papers*, 2010, pp. 386–387.

[13] V. Giannini, P. Nuzzo, V. Chironi, A. Baschirotto, G. Van der Plas, and J. Craninckx, "An 820µW 9b 40MS/s Noise-Tolerant Dynamic-SAR ADC in 90nm Digital CMOS," in *ISSCC Dig. Techn. Papers*, 2008, pp. 238–239.

[14] I. Daubechies and O. Yilmaz, "Robust and Practical Analog-to-Digital Conversion With Exponential Precision," *IEEE Trans. Information Theory*, vol. 52, no. 8, pp. 3533–3545, 2006.

[15] W. Liu, P. Huang, and Y. Chiu, "A 12b 22.5/45MS/s 3.0mW 0.059mm2 CMOS SAR ADC achieving over 90dB SFDR," in *ISSCC Dig. Techn. Papers*, 2010, pp. 380–381.

[16] J. Biveroni and H.-A. Loeliger, "On sequential analog-to-digital conversion with low-precision components," in *2008 Information Theory and Applications Workshop*, 2008.

[17] M. K. Mayes and S. W. Chin, "A 200 mW, 1 Msample/s, 16-b pipelined A/D converter with on-chip 32-b microcontroller," *IEEE J. Solid-State Circuits*, vol. 31, no. 12, pp. 1862–1872, Dec. 1996.

[18] B. Murmann and B. E. Boser, "A 12-bit 75-MS/s pipelined ADC using open-loop residue amplification," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 12, pp. 2040–2050, Dec. 2003.

[19] Y. Chai and J.-T. Wu, "A 5.37mW 10b 200MS/s dual-path pipelined ADC," in *ISSCC Dig. Techn. Papers*, 2012, pp. 462–464.