Digital assistance for energy reduction in analogue-to-digital converters using a signal prediction algorithm

D. Avila, E. Alvarez and A. Abusleme

With the adoption of new technologies for analogue circuits, different digital techniques have been created to enhance their performance. Among the existing methods, a promising approach is to dynamically adapt the circuit operation considering the application characteristics. In the field of analogue-to-digital converters (ADCs), typically this approach is carried out by taking advantage of application-dependent signal properties, hence their use is limited. In this work, a digital assistance method for power reduction in ADCs is presented. The proposed method consists of an algorithm based on the input signal variation between samples, which allows to reduce the mean energy consumption per conversion in a variety of ADC architectures, regardless of the application.

Introduction: Digital assistance is used to expand the scope of analogue circuits by taking advantage of the fast growing performance of digital circuits resulting from the technology scaling [1]. Among the different methods of digital assistance, a promising approach is to dynamically adapt the circuit operation according to the characteristics of its input signal, in order to, for example, reduce the overall energy consumption. Several applications using this approach can be found in the design of ADCs for high-efficiency, low-power applications [2], [3], [4]. Typically, this approach is carried out by using application-dependent signal characteristics, such as periodical changes in the input signal rate or the input signal known shape. For this reason, these techniques lack of generality and can only be used in some specific applications. In this work, a digital assistance method for energy reduction in general-purpose ADCs is presented.

When designing an on-chip ADC for a custom SoC, the maximum rate of change of the ADC input signal is assumed to be a known variable, determined by the frequency bandwidth of the stage prior to the ADC, typically a signal-conditioning stage. Using this parameter and the ADC sampling frequency, an expression for the maximum variation between samples of the ADC output can be computed. Based on this variation, the most significant bits (MSBs) that will remain unchanged in the following conversion can be determined. Thus, for the next conversion, the ADC only needs to compute the remaining changing bits, reducing the mean energy consumption per conversion. For instance, in flash ADCs, the application of this method allows to turn off 50 % of the comparators per unchanged bit, whereas in bit-at-a-time ADCs, this method allows for a higher sampling rate when operating asynchronously, and an energy saving proportional to the mean number of unchanged bits.

This work is structured as follows: first, the computation of the maximum variation between samples of the ADC output as a function of the previous stage frequency bandwidth is shown, and then the mean number of unchanged bits per conversion is determined. Finally, behavioural simulation results are presented and the conclusions are drawn.

Maximum variation of the ADC output: The maximum variation between consecutive samples of the ADC input voltage, $\Delta v_{\text{in}}|_{\max}$, can be calculated considering that v_{in} is maintained at its maximum rate of change for a complete sampling period, as illustrated in Fig. 1. Mathematically, $\Delta v_{\text{in}}|_{\max}$ can be written as

$$\Delta v_{\text{in}}|_{\max} = \left.\frac{\mathrm{d}v_{\text{in}}}{\mathrm{d}t}(t)\right|_{\max} \times \frac{1}{f_s} \tag{1}$$

where $\left.\mathrm{d}v_{\text{in}}/\mathrm{d}t\,(t)\right|_{\max}$ is the maximum rate of change of v_{in} and f_s is ADC sampling frequency. The maximum variation of the ADC output between consecutive samples, N, can be written as a function of $\Delta v_{\text{in}}|_{\max}$ as follows

$$N = \left\lceil \frac{\Delta v_{\text{in}}|_{\max}}{LSB} \right\rceil \tag{2}$$

where $LSB = FSR/2^B$ is the least significant bit, FSR is the ADC full scale range, B is the ADC number of bits, and $\lceil\cdot\rceil$ is the ceiling function.

For illustration purposes, let us suppose that the bandwidth of v_{in} is limited by a brick-wall filter with cut-off frequency of $f = BW$.

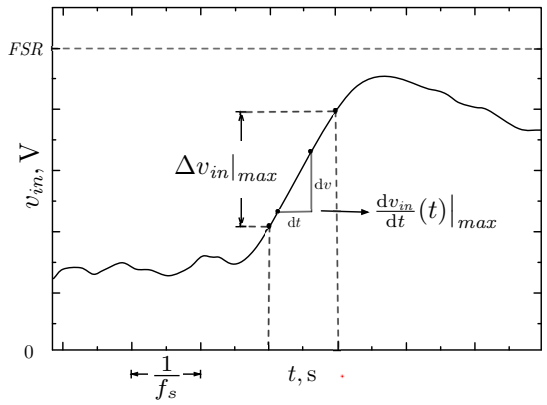

Fig. 1. Illustration of the maximum voltage variation at the ADC input.

Considering this, the maximum rate of change of v_{in} is given by the maximum slope of a sine wave of frequency BW and amplitude $FSR/2$. Therefore, (2) can be written as a function of the ADC input voltage bandwidth as

$$N = \left\lceil \frac{\pi \times BW \times FSR}{LSB} \times \frac{1}{f_s} \right\rceil = \left\lceil \frac{2^{B-1} \times \pi}{OSR} \right\rceil \tag{3}$$

where $OSR = f_s/(2 \times BW)$ is the oversampling ratio. In a practical design, N must be calculated considering the actual frequency response of the stage prior to the ADC.

Algorithm: Considering that the maximum variation of the input signal between two consecutive samples is $\pm N$, the proposed algorithm works as follows: once the current digital output L is computed, a variation of $\pm N$ is applied to L and the results given by $\min(L+N, 2^B-1)$ and $\max(L-N, 0)$ are bitwise compared with L. Once these comparisons have been made, the number of MSBs that will remain unchanged in the following conversion, UB, can be determined. Fig. 2 shows a pseudo code implementation for the proposed algorithm, which has been formulated to be written in a hardware description language to synthesize a combinational digital circuit. In this code, the last digital output and N are manipulated to create a mask that positional encode the number of bits that remains unchanged, then, a count leading zeros function is used to translate the positional value into a binary value.

```
a = max(L-N,0) ⊕ L
b = min(L+N,2^B-1) ⊕ L
c = a ⊕ b
UB = clz(c)
```

Fig. 2 *Pseudo code for the proposed algorithm. The operand ⊕ is the bitwise exclusive OR operator and clz the count leading zeros function.*

Using this algorithm, the operation of an ADC can be adapted considering that the UB MSBs of the following conversion are already known, so there is no need to compute them again and the mean energy consumption per conversion can be reduced.

The general expression for the number of codes whose i MSBs remain unchanged after a variation of $\pm N$ LSBs, B_i, is given by

$$B_i = \begin{cases} 2(2^x - N) & i = B - x \\ 2(2^{B-i} - N) + (2^i - 2)(2^{B-i} - 2N) - \sum_{j=i+1}^{B-x} B_j & i = 2 \ldots B - x - 1 \\ 2^B - 2N - \sum_{j=2}^{B-x} B_j & i = 1 \end{cases} \tag{4}$$

where $x = \lceil \log_2(N+1) \rceil$.

Assuming an uniformly distributed input, the mean number of unchanged bits per conversion, $\overline{UB}$, can be calculated as

$$\overline{UB} = \frac{1}{2^B} \sum_{i=1}^{B-x} i \times B_i \tag{5}$$

Replacing (4) in (5), it can be seen that $\overline{UB}$ only depends on the OSR, and not on the ADC number of bits.

A variation of the proposed algorithm can be formulated for asynchronous bit-at-a-time ADCs. In these ADCs, the time of conversion depends on UB. Therefore, N is not constant and can be recalculated at the end of each conversion. Assuming that the reset of the ADC takes one clock period, at the i-th conversion, N_i can be computed dynamically as

$$N_i = \left\lceil \frac{2^{B-1} \times \pi}{OSR} \times \frac{B - UB_{i-1} + 1}{B+1} \right\rceil . \quad (6)$$

It can be seen that under this variation, the number of unchanged bits depends on the signal shape and not on the signal statistics, thus, $\overline{UB}$ cannot be calculated for asynchronous operation.

Behavioural simulations results: The proposed algorithm, along with its variation with dynamic N calculation for asynchronous operation, were simulated for a 10-bit ADC. Both algorithms were tested using an input ramp from 0 V to FSR V, and using an input sine wave with an amplitude of $FSR/2$ and a frequency of $f = BW$.

Table 1 shows $\overline{UB}$ for different values of OSR. As mentioned before, $\overline{UB}$ does not depend on B, so the results shown in Table 1 are also valid for a generic B-bit converter. Using these results and considering the ADC architecture, the mean saved energy consumption per conversion can be determined. For instance, in a Flash ADC each unchanged bit allows to turn off 50 % of the comparators, thus, a mean energy reduction of 18.37 % can be reached for $OSR = 4$, whereas a mean energy reduction of 45.32 % can be reached for $OSR = 8$. Both values of OSR are considered as Nyquist rates [5].

OSR	2	4	8	16
$\overline{UB}$ from (5)	0	0.29	0.87	1.56
FSR ramp, static N	0	0.21	0.71	1.27
FSR ramp, dynamic N	0	0.27	0.85	1.50
$FSR \sin(2 \times \pi \times BW)/2$, static N	0	0.73	1.59	2.42
$FSR \sin(2 \times \pi \times BW)/2$, dynamic N	0	0.77	1.67	2.56

Table 1: Simulated $\overline{UB}$ as a function of OSR.

Conclusion: An algorithm to reduce the mean number of bits calculated per conversion is presented. Since the algorithm can be implemented with a small combinatorial digital circuit, the introduced power is marginal compared with the saved power resulting by using the algorithm. Besides, the algorithm allows an speed improvement in asynchronous bit-at-a-time ADCs.

Acknowledgment: The authors thank the National Comission for Scientific and Technological Research (CONICYT) of Chile, through project FONDECYT 11110165, for making it possible to continue with this research.

D. Avila, E. Alvarez and A. Abusleme (*Department of Electrical Engineering Pontificia Universidad Católica de Chile, PO Box 306 - Correo 22, Santiago 7820436, Chile*)

E-mail: dlavila@uc.cl

References

1 Murmann, B.: 'Digitally Assisted Analog Circuits', *Micro. IEEE* , 2006, **26**, no.2, pp. 38-47.
2 Trakimas, M.; Sonkusale, S.R., 'An Adaptive Resolution Asynchronous ADC Architecture for Data Compression in Energy Constrained Sensing Applications', *Tran. on Circ. and Sys I*, 2011, **58**, no.5, pp. 921-934.
3 O'Driscoll, S.; Shenoy, K.V.; Meng, T.H., 'Adaptive Resolution ADC Array for an Implantable Neural Sensor', *IEEE Trans. on Biom. Cir. and Sys* , 2005, **5**, no.2, pp. 120-130.
4 Zaare, M.; Sepehrian, H.; Maymandi-Nejad, M.; 'A New Non-Uniform Adaptive-Sampling Successive Approximation ADC for Biomedical Sparse Signals', 2014, **74**, no.2, pp. 317-330.
5 Maloberti, F.: 'Data Converters', *Springer London*, 2007.