An 11.0 bit ENOB, 9.8 fJ/conv.-step Noise-Shaping SAR ADC Calibrated by Least Squares Estimation

Harald Garvik, Carsten Wulff and Trond Ytterdal
Dept. of Electronics and Telecommunications, Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

***Abstract*—A noise-shaping SAR ADC implemented in 28 nm UTBB FDSOI is presented. A cascaded FIR-IIR loop filter topology is used, and implemented as an inverter-based switched-capacitor circuit. The loop filter also employs input buffers. To correct CDAC mismatch, an off-line calibration technique that estimates the CDAC calibration coefficients from a digitized test sequence is proposed and used. At 28 MS/s and Nyquist bandwidth of 1.75 MHz, the measured accuracy is 11.0 bit ENOB, and the Walden FOM 9.8 fJ/conv.-step.**

I. Introduction

Noise-shaping SAR ADCs (NS-SARs) [1]–[4] are good candidates for high-accuracy analog-to-digital conversion. That is because both quantization noise and comparator noise are efficiently alleviated through oversampling and noise-shaping, making it possible to use fewer bits in the SAR, and a small, simple comparator. When high conversion accuracy is required, however, distortion induced by mismatch in the CDAC quickly becomes the limiting factor, as the capacitor matching needs to be as good as the conversion accuracy. In [2], this problem was mitigated by integrating first-order mismatch-shaping in the CDAC, resulting in an NS-SAR with conversion accuracy above 100 dB SNDR. With mismatch shaping transfer function $(1 - z^{-1})$, however, the technique is only effective if the oversampling rate (OSR) is reasonably high, and this is not attractive if maximum bandwidth is demanded. Reference [3] points out that classical SAR ADC calibration [5] can be used for NS-SARs. As this calibration is unrelated to the oversampling, this approach works well independent of OSR.

In this work, we present an 11.0-bit ENOB, 1.75 MHz BW NS-SAR implemented in 28 nm UTBB FDSOI. Its loop filter topology is similar to the one introduced in [1], but implemented as a pseudo-differential, inverter-based circuit, similar to what is done in the $\Delta\Sigma$-modulators presented in [6]. Also, modified source-followers [7] are used as loop filter input buffers. Off-chip calibration is implemented using a proposed technique that estimates the CDAC mismatch errors from a digitized test sequence. This is done by the solution of a linear least squares problem, which fits the distortion components in the test sequence to a signal model, extracted from the same sequence. At run-time, the estimated mismatch errors are then used to correct the uncalibrated codes.

II. Design Overview

Figure 1 shows the architecture of the proposed NS-SAR. It consists of a 9-bit MOM CDAC, a dynamic comparator with two input pairs, a bootstrapped sampling switch, a SAR digital circuit, and a switched-capacitor loop filter. In addition, code correction is performed off-chip, using the calibration coefficients found by the method that will be introduced in section IV. The CDAC, comparator, sampling switch and digital circuit are variants of the ones in [8], and their layout is compiled using a closed source compiler. An open source version is available from [9].

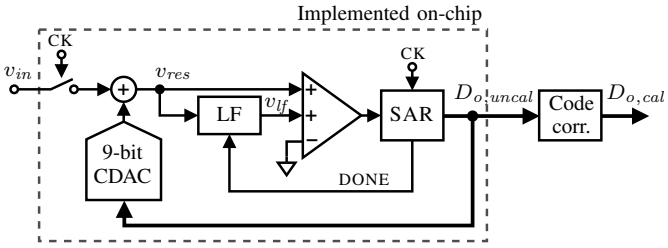


Fig. 1. Architecture of the proposed NS-SAR (actual implementation is differential).

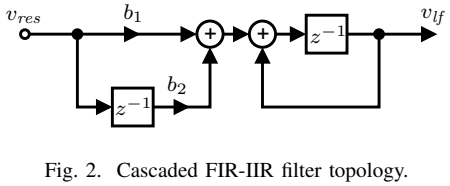


Fig. 2. Cascaded FIR-IIR filter topology.

The loop filter input is connected directly to the CDAC top plates, and its output controls the extra comparator input pair. It receives its clock from the SAR digital circuit, which rises the signal DONE when the bit cycling is finished. This triggers sampling of the residue voltage from the CDAC, followed by update of the loop filter output before the next conversion takes place. It is shown in [10] that by connecting a loop filter to a SAR ADC in this way, the quantization noise and comparator noise are noise-shaped by the noise transfer function $1/(1 + H(z))$, where $H(z)$ is the loop filter transfer function.

III. Loop Filter Implementation

The cascaded FIR-IIR loop filter topology introduced in [1] is adapted to this work. It is shown in figure 2, and consists of a delaying discrete-time integrator, and a two-tap FIR filter. The loop filter gives rise to the noise transfer function (NTF)

$$\mathrm{NTF}(z) = \frac{1 - z^{-1}}{1 + (b_1 - 1)z^{-1} + b_2 z^{-2}}. \tag{1}$$

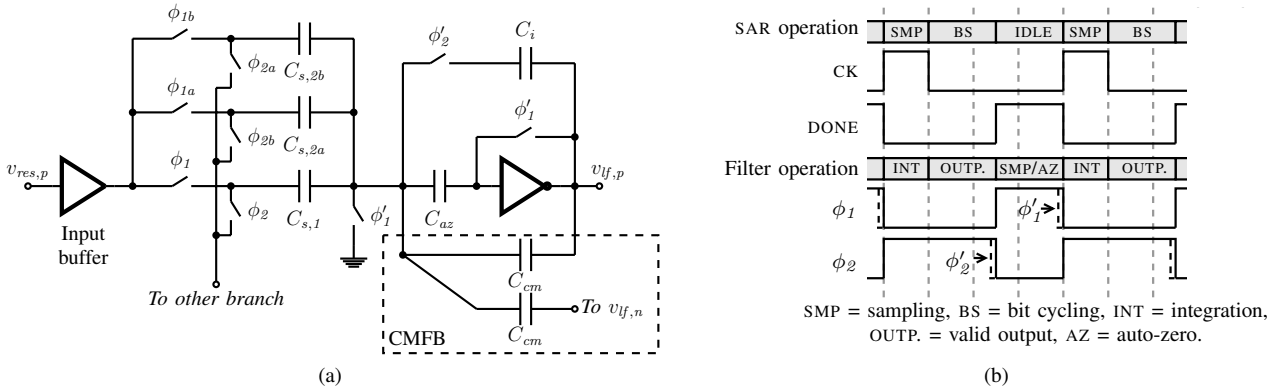


Fig. 3. (a): Loop filter schematic (only positive branch drawn). (b): Timing diagram.

It thus has one zero at DC, and two selectable poles. As the 9-bit quantization of the ADC results in inherently high loop filter stability, it is possible to choose the poles aggressively [10]. In this implementation, $b_1 = 2.9$ and $b_2 = 0.9$ were used, leading to better than first order noise-shaping.

The cascaded FIR-IIR filter is implemented as a pseudo-differential, inverter-based, switched-capacitor circuit, shown in figure 3a (only positive branch drawn). It has input buffers, and uses auto-zeroed inverters as gain elements. The two main non-overlapping clock phases are ϕ_1, and ϕ_2, derived from the DONE signal from the SAR circuit, as shown in the timing diagram in figure 3b. The main ADC sample clock CK is also shown, together with the operations being performed by the SAR and the loop filter at different points in time. In addition to these clocks, the loop filter also uses two other clock signal pairs to implement the delayed FIR filter tap. That is ϕ_{1a} and ϕ_{1b}, and ϕ_{2a} and ϕ_{2b}. These are similar to ϕ_1 and ϕ_2, but have every other clock pulse masked, in an interleaved fashion. Finally, ϕ'_1 and ϕ'_2 fall before the other clocks, and are used to mitigate signal dependent charge injection.

The filter and NS-SAR operate as follows: After the SAR has finished bit-cycling, the DONE signal is rised, and the loop filter enters the ϕ_1 phase. A buffered version of the CDAC residue voltage is then sampled on capacitor $C_{s,1}$, and either $C_{s,2a}$ or $C_{s,2b}$. The inverter is put in reset, and its reset voltage is sampled across the auto-zero capacitor C_{az}. The operation then proceeds to ϕ_2 when CK rises. Now the inverter is taken out of reset, and the integration capacitor C_i is connected in negative feedback. Also, $C_{s,1}$, and either $C_{s,2a}$ or $C_{s,2b}$ (depending on which that holds the oldest signal sample) are connected to the same capacitors in the negative circuit branch, and differential charge transfer to C_i takes place. This charge transfer has to settle sufficiently before the SAR starts bit cycling, as the updated loop filter output is then needed.

The common mode feedback network (CMFB) consists of four capacitors C_{cm} when both branches are considered. In ϕ_1, they sample the reset voltages of the inverters. Then in ϕ_2, the difference between the inverter reset voltages and the inverter output voltages is transferred to C_i, forcing the output common mode voltage towards the reset voltages. It can be shown by circuit analysis that the output common mode voltage of the circuit in the z-plane is

$$V_{cm}(z) = \frac{2C_{cm}}{C_i} \frac{1}{1 + \frac{2C_{cm}}{C_i} - z^{-1}} V_{reset,avg}, \qquad (2)$$

and it thus settles to the average of the positive and negative inverter reset voltage $V_{reset,avg}$, with a speed adjustable by C_{cm}/C_i. It should be pointed out that the CMFB used in this work is actually the same as the one with eight switches used in [6]. The difference is that in our work, the common mode reference voltage is sampled directly from the inverter output nodes during reset, and this makes all the switches redundant.

To mitigate flicker noise, the loop filter input buffers are chopped by the frequency $f_s/2$. The clocks needed for this are the already existing ϕ_{1a} and ϕ_{1b}. At the input, the chopping is implemented by a standard "butterfly style" switching network between the positive and negative buffers. At the output, the chopping is integrated with the residue sampling switches in figure 3a, but this is not shown for simplicity. In practice, the switches in the $C_{s,2b}$ branches are connected to the buffer outputs in the opposite loop filter branches, and the sampling switches for $C_{s,1}$ are replaced by a butterfly switch network.

A. Inverter amplifier

The inverter amplifiers used in the loop filter have a standard cascoded inverter topology. Cascoding is used both to increase the gain, and decrease parasitic input capacitance (by decreasing the Miller effect). The cascode devices are biased to V_{DD} and V_{SS} (through tie-cells). Regular threshold voltage devices are used for all transistors, rather than low threshold voltage devices. This increases the g_m/I_D ratio for all devices, and also makes it easier to put the cascode devices in saturation.

Inverter amplifiers have a unity gain frequency of $f_t = (2g_m)/(2\pi C_L)$, and this is achieved using only one current branch. The efficiency is thus high, and this was one of the main reasons why inverters were chosen for this work. One of the drawbacks is that large auto-zero capacitors are needed. However, this only increases area and not power, as the charge transfer to these capacitors is negligible.

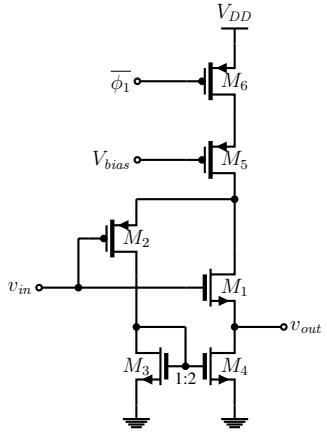


Fig. 4. Loop filter input buffer schematic.

B. Input Buffer

Most reported NS-SARs passively extract the residue from the CDAC by charge sharing with the loop filter input [1], [3], [4]. This may lead to significant attenuation. In this work, loop filter input buffers are introduced to mitigate this problem (a buffer was also used independently in [2]).

The buffer is shown in figure 4 (one is used in each loop filter branch). It is the modified source follower topology introduced in [7], with the addition of transistor M_6. This transistor makes it possible to enable the buffer only in the ϕ_1 phase, when residue sampling takes place. The bias currents are thus shut off for around two thirds of the ADC operation cycle. In short, the advantage of using the modified source follower topology is increased gain and linearity. This is accomplished by adding the extra source follower M_2, decreasing the V_{ds} variations across the main source follower M_1. Also, a negative feedback loop is added by $M_{3,4}$, counteracting current changes through the main source follower branch.

IV. Calibration Algorithm

(NS)-SAR ADCs rely upon binary scaled CDACs, and if the accuracy of this scaling is insufficient, distortion occurs. This is essentially because the binary weights of the digital output words change to unknown non-binary values, and to correct the non-linearity, these weights have to be captured or estimated. Good methods exist to measure and correct the weights directly [5], but in this work we propose and use a technique to estimate them externally, without doing any changes to the ADC.

To calibrate, the ADC input is first excited by a test sinusoid, and a resulting digital output sequence $d_{out}[n]$ of K samples is captured ($K = 2^{14}$ in this work). This sequence will contain both the sinusoid, and harmonic distortion components (in addition to noise). From $d_{out}[n]$, all the bit single patterns $b_0[n], b_1[n], ..., b_{N-1}[n]$ are extracted and stored in own vectors. Then, $d_{out}[n]$ is band-pass filtered using a linear-phase FIR filter to remove the input signal, and some of the high-frequency noise. This results in a signal $d_{dist}[n]$ containing the harmonic distortion, in addition to the noise in the same frequency region. The group delay introduced by the filter on $d_{dist}[n]$ must be removed.

Next, $d_{dist}[n]$ is approximately modeled as a linear combination of all the bit patterns extracted earlier. I.e. $d_{dist}[n] \approx d'_{dist}[n] = c_0 b_0[n] + c_1 b_1[n] + ... + c_{N-1} b_{N-1}[n]$. The coefficients c_i are then an estimate for the part of the binary weights that makes the harmonic distortion, and can thus be used to correct them. Let B be a $K \times N$ matrix containing all the N bit patterns as columns, and c the vector containing the unknown coefficients. The modeling can then be treated as a linear least squares problem. That is, we seek the approximate solution of the overdetermined linear equation set $Bc = d_{dist}$, that minimizes $||Bc - d_{dist}||_2 = ||d'_{dist} - d_{dist}||_2$, where $||\cdot||_2$ is the two-norm. This problem has the well-known solution $B^T Bc = B^T d_{dist}$, known as the normal equations. It can be solved automatically by tools like Matlab, used in this work.

The resulting coefficient set is then used to correct the binary weights. For digital ones, the new binary weights are $(b_i - c_i)$. Also, the weights for digital zeros are changed from 0, to $(0 + c_i)$. The weight correction essentially subtracts the estimate of the harmonic distortion from the total signal.

In practice, the calibration in this work was carried out by capturing the sequence $d_{out}[n]$, and then computing the calibration coefficients in Matlab. Then, these same coefficients were used directly (off-chip) for correction of all data sequences subsequently captured and used in section V of this paper. This shows that the method works when the input frequency and input amplitude are varied after the calibration coefficients are obtained. Also, it was found sufficiently to only correct the three largest bit weights. This is attractive if code-correction is to be applied on-chip, as the three coefficients can be combined into 2^3 correction terms and stored in a small look-up table, meaning that only a single table look-up and an addition are needed per sample to obtain the corrected codes.

The drawback of the proposed method is that an external test signal is needed to perform the calibration, and that a CPU is needed to calculate the coefficients. The advantage is that no change at all is needed to the ADC itself, meaning that the design complexity is reduced. Also, the Matlab program used to compute the coefficients in this work completes in 0.7 seconds on an ordinary laptop, suggesting that the method may be quick enough for use during production testing.

V. Measurement Results

The ADC prototype is fabricated in 28 nm UTBB FDSOI, and the die photo, together with the layout is shown in figure 5. The ADC measures $133.5\,\mu\text{m} \times 119\,\mu\text{m}$, giving it an area of $0.016\ \text{mm}^2$. Three chips were tested, and only small performance variations were found between them.

Figure 6 shows the spectrums for calibrated and uncalibrated ADC outputs, for a test sinusoid of 211 kHz. The spectrums were generated using Welch's method on output sequences of 2^{17} samples. The sampling frequency is 28 MS/s, and the signal bandwidth is 1.75 MHz (OSR 8). The calibration improves the SFDR by 23 dB at the chosen input frequency.

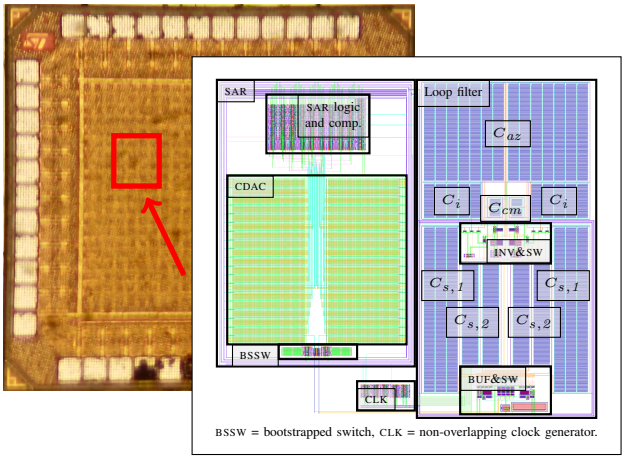


Fig. 5. Die photo with layout overlay.

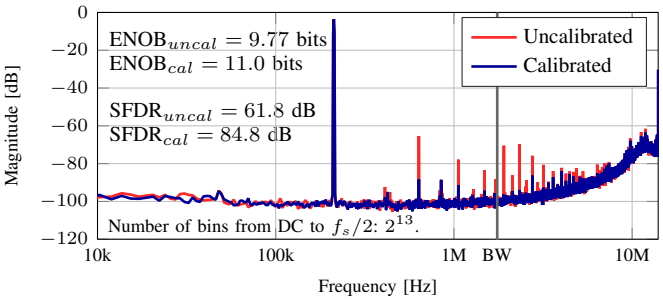


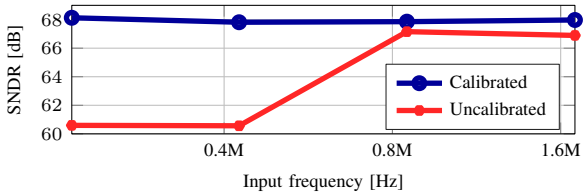
Fig. 6. Measured spectrums. $f_{in} = 211$ kHz.

The spur at $f_s/2$ is the chopped offset of the buffers. Figure 7 shows SNDR vs. input frequency and input amplitude, verifying that the calibrated performance is consistent across these variables. The uncalibrated SNDR increases with input frequency because more of the spurs are shifted out of the signal band. The presented ADC is compared against other NS-SARs in table V, and shows that this work performs energy efficiently both in terms of Walden FOM and Schreier FOM.

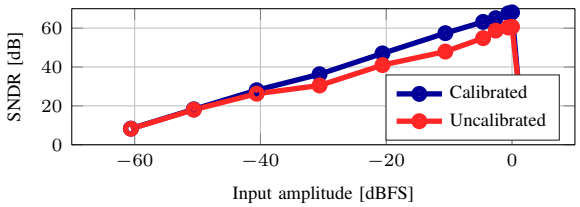
TABLE I
COMPARISON TO PRIOR NOISE-SHAPING SAR ADCS

	[1]	[2]	[3]	[4]	This work
Technology (nm)	65	55	130	65	28 FDSOI
f_s (MS/s)	90	1	2	50	28
B_w (MHz)	11	0.004	0.125	6.25	1.75
OSR	4	125	8	4	8
Area mm^2	0.03	0.072	0.13	0.012	0.016
Supply (V)	1.2	1.2	1.2	0.8	0.81
ENOB (bits)	10.0	15.7	12	9.35	11.0
SNDR (dB)	62.0	96.1	74	58.03	68.1
SFDR (dB)	72.0	105.1	95	–	84.8
Power (µW)	806	15.7	61	120.7	70.5
FOM$_w$ (fJ/c.step)[1]	35.8	36.9	59.6	14.8	9.8
FOM$_s$ (dB)[2]	163.4	180.0	167	165.2	172.0

[1] FOM$_w$ = (Power)/$(2^{\text{ENOB}} \cdot 2B_w)$
[2] FOM$_s$ = SNDR + 10 log $(B_w$/Power)



(a)



(b)

Fig. 7. SNDR: (a): Vs. input frequency, (b): Vs. input amplitude.

VI. CONCLUSION

A noise-shaping SAR ADC in 28 nm UTBB FDSOI has been presented. The cascaded FIR-IIR loop filter topology was adapted, but implemented as an inverter-based switched-capacitor circuit, using modified source-followers as input buffers. Off-chip calibration was employed to correct CDAC mismatch, using a proposed technique that estimates the calibration coefficients from a digitized test signal. The ADC achieves an accuracy of 11.0 bit ENOB, and a Walden FOM of 9.8 fJ/conv.-step, making its energy efficiency state-of-the-art.

REFERENCES

[1] J. Fredenburg and M. Flynn, "A 90-MS/s 11-MHz-Bandwidth 62-dB SNDR Noise-Shaping SAR ADC," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 2898–2904, Dec. 2012.

[2] Y. S. Shu, L. T. Kuo, and T. Y. Lo, "27.2 an oversampling SAR ADC with DAC mismatch error shaping achieving 105dB SFDR and 101dB SNDR over 1kHz BW in 55nm CMOS," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan. 2016, pp. 458–459.

[3] W. Guo and N. Sun, "A 12b-ENOB 61 uW noise-shaping SAR ADC with a passive integrator," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Sep. 2016, pp. 405–408.

[4] Z. Chen, M. Miyahara, and A. Matsuzawa, "A 9.35-ENOB, 14.8 fJ/conv.-step fully-passive noise-shaping SAR ADC," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*, Jun. 2015, pp. C64–C65.

[5] H. S. Lee, D. A. Hodges, and P. R. Gray, "A self-calibrating 15 bit CMOS A/D converter," *IEEE Journal of Solid-State Circuits*, vol. 19, no. 6, pp. 813–819, Dec. 1984.

[6] Y. Chae and G. Han, "Low Voltage, Low Power, Inverter-Based Switched-Capacitor Delta-Sigma Modulator," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 458–472, Feb. 2009.

[7] B. Hernes, J. Bjornsen, T. Andersen, A. Vinje, H. Korsvoll, F. Telsto, A. Briskemyr, C. Holdo, and O. Moldsvor, "A 92.5mW 205MS/s 10b Pipeline IF ADC Implemented in 1.2V/3.3V 0.13 um CMOS," in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, Feb. 2007, pp. 462–615.

[8] C. Wulff and T. Ytterdal, "A compiled 3.5fJ/conv.step 9b 20MS/s SAR ADC for wireless applications in 28nm FDSOI," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Sep. 2016, pp. 177–180.

[9] C. Wulff, "cIcCreator – Custom IC Creator," http://www.ciccreator.org/, 2016.

[10] H. Garvik, C. Wulff, and T. Ytterdal, "Noise transfer functions and loop filters especially suited for noise-shaping SAR ADCs," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 1034–1037.