2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST) | 978-1-6654-1847-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/MOCAST52088.2021.9493368

Exploring the Effectiveness of Sigma-Delta Modulators in Stochastic Computing-Based FIR Filtering

Anastasios Vlachos, Nikos Temenos and Paul P. Sotiriadis
Department of Electrical and Computer Engineering
National Technical University of Athens, Greece
E-mail: vlahosanastasis@gmail.com, ntemenos@gmail.com

***Abstract*—A soft-filtering processing architecture based on Sigma-Delta Modulation and Stochastic Computing is proposed. It converts a high-resolution signal using a first order digital Sigma-Delta Modulator into a single-bit one and then exploits Stochastic Computing's encoding to perform area-efficient multiplications. The Sigma-Delta Modulator allows for the input signal to be oversampled at a much higher frequency rate, offering improved performance in terms of SNR, which is not possible with standard Stochastic Computing filter realizations. Spectral simulations results demonstrate the proper signal quantization and operation of the filter, including the filter's roll-off behavior. FPGA synthesis results of the proposed architecture, illustrate its area advantages in comparison to conventional binary filtering.**

***Index Terms*—Stochastic Computing, Digital Sigma-Delta Modulators, Soft-Filtering, Stochastic Computing Filters**

I. Introduction

The continuously increasing computational processing demands in modern-day Digital Signal Processors (DSPs), have pushed the research towards efficient alternatives to standard binary computing. Among many unconventional techniques used, Stochastic Computing (SC) is considered a well promising one [1].

SC processes signals by encoding the information of real-valued binary numbers into stochastic sequences [2]. Therefore, its probabilistic nature makes it highly-tolerant to soft-errors, for instance bit-flips, that originate from various noisy sources. Moreover, SC has the ability to realize effectively both fundamental arithmetic operations and complex functions at a relatively negligible logic cell cost, when compared to the binary counterparts. These advantages favor several applications that rely heavily on parallel computations including the fields of Neural Networks [3], Digital Image Processing [4], as well as soft-filtering [5]–[8].

Emphasizing on standard Nyquist-rate digital filters, SC-based implementations are proven to be hardware-efficient [5]–[7]. This is due to the fact that binary multipliers are replaced by XNOR gates, thus leading to increased area savings from the intensive multiply-and-add operations. The summation part used in [6], [7] is implemented by MUXs, which is the standard circuit used to implement the addition in SC.

To further increase the spectral characteristics of digital filters, Sigma-Delta Modutlators (SDMs) can be used. They convert a higher-resolution signal into a lower one and employ the technique of oversampling, i.e. the sampling of the input signal at a frequency much higher than the Nyquist, to push the in-band noise outside the desired frequency band. Filtering of SDM encoded signals using SC-based implementations, has been investigated in [8]. However, it is approached solely from the SC filter implementation perspective, meaning that the SDM stage's benefits are not investigated and its output is assumed to be available for processing.

Motivated by the work in [8], we extend the combination of the SDM encoding and the SC filtering into a general signal processing scheme and focus on both of its design aspects. The SDM-SC configuration utilizes a single-bit SDM to encode a multi-bit input signal into a single-bit one and then performs multiplications using SC elements. In contrast to the filtering methods used in [6], [7], we replace the hardware-taxing MUX-adders [9], [10] with a binary adder, so as to further reduce the area requirements.

The remainder of this paper is organized as follows. In Section II, we provide a background to the stochastic number representation, the first order SDM and the general notation used. In section III, we present the SDM-SC processing scheme and explain its principle operation. Section IV includes simulation as well as preliminary FPGA synthesis results accompanied by comparisons with the standard binary approach Finally, section V provides the conclusion.

II. Stochastic Computing and Sigma-Delta Modulation Basics

In this section, we provide with a brief background on the stochastic numbers and their notation as well as the first order Sigma-Delta Modulator (SDM).

A. Stochastic Number Encoding

The conversion process of a binary number into a stochastic one, is typically implemented by the stochastic number generator (SNG) shown in Fig. 1. It operates as a sampling unit by comparing on each clock cycle a k-bit Linear-Feedback Shift Register (LFSR) uniformly distributed in $\mathcal{R}_S = \{0, 1, \dots, 2^k - 1\}$ with the desired binary number $B \in [0, 1]$ of the same length k. After $N = 2^k$ clock cycles, the bit generation is completed and corresponds to the length

978-1-6654-1847-8/21/$31.00 ©2021 IEEE

of the sequence. It is important to note that sampling beyond N clock cycles, the LFSR will cycle again through all of its values which will consequently introduce correlation to the generated sequence [6].

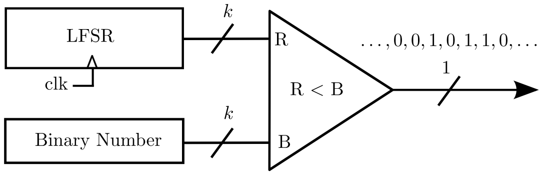

Fig. 1. Stochastic Number Generator (SNG) [1]

Formally, the N-bit sequence generated from the SNG, i.e. $\{X_n\}_{n=1}^{N}$, where n denotes the current time index (or clock cycle), is assumed to approximate a Bernoulli process. Therefore, the SN's value is positive in the range $[0, 1]$, known as unipolar format, with probability defined as $X \triangleq P(X_n = 1)$ and a mean value given as

$$\tilde{X}_N = \frac{1}{N}(X_1 + X_2 + \cdots + X_N). \tag{1}$$

Negative number representation is also supported in SC by applying the transformation $X \mapsto 2X - 1$, expanding the range of the SN to $[-1, 1]$. According to the format used, certain operations are realized by different logic gates, for instance the multiplication is realized by an AND gate in the unipolar format whereas in the bipolar by an XNOR gate.

B. The First Order Sigma-Delta Modulator

The Sigma-Delta Modulator of Fig. 2 converts a multi-bit signal to a single-bit one by exploiting the technique of oversampling to push the in-band quantization noise outside the signal's frequency band.

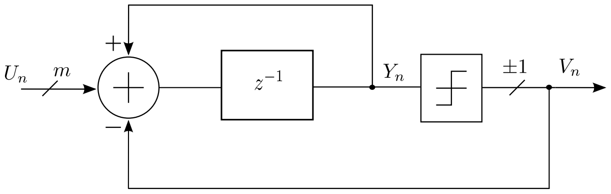

Fig. 2. First Order Sigma-Delta Modulator

It comprises of an adder and an integrator followed by a two-step quantizer block. The modulator's input denoted as U_n, is of m-bits length, whereas its output V_n is single-bit ± 1, determined by the quantizer. According to Fig. 2 the equation that captures the SDM's operation is given as

$$V_n = Q\left(Y_{n-1} + U_n - V_{n-1}\right), \tag{2}$$

where $Q(\cdot)$ denotes the two-step quantization function.

With respect to the SDM's oversampling characteristics, if its sampling frequency is f_s and f_B is the maximum input signal's frequency, then the oversampling ratio (OSR) is defined as

$$OSR \triangleq \frac{f_s}{2f_B}. \tag{3}$$

In the following section, we use the SDM's basic principles to explain the proposed processing scheme.

III. Proposed SDM-SC Processing Scheme

In this section we present the proposed SDM-SC processing scheme shown in Fig. 3. It consists of two blocks which are a first order single-bit digital SDM followed by an M tap FIR filter. The key concept of the architecture is to convert a multi-bit input signal into a single-bit one by using the SDM. Then, the encoded signal is further processed by stochastic computing elements and benefits from SC's area advantages. We proceed with the detailed its explanation starting from the SDM.

A. Digital SDM Encoding

The SDM shown in the architecture of Fig. 3 is the digital representation of the system-level one shown in Fig. 2. The multi-bit input is denoted as U_n while V_n is the single-bit output. Here, the integrator of Fig. 2 is replaced by a register of c-bits, so as to implement the SDM's iterative operation according to (2). We note that the register's size c should be greater than m to account for the accumulation, with typical value of $c = m + 1$.

The non-linear operation of quantization is modeled simply as the register's most significant bit (MSB). This is because the input U_n determines if the accumulator's current value is greater than or equal to 0. Hence, no further information besides the MSB is required. Finally, given the fact that V_n outputs 0 instead of -1, the conversion is done using sign extensions methods.

It is important to note at this point that the SDM's maximum operating frequency f_s (corresponding to the register's clock) determines two parameters according to (3): 1) the maximum OSR that can be selected and 2) the input signal's maximum frequency f_B that the architecture is able to process.

B. Stochastic FIR Filter

A general M-tap, or $(M-1)^{th}$ order, Finite Impulse Respone (FIR) filter computes each value of its output sequence as a weighted sum of the M most recent input values as follows

$$Z_n = \sum_{i=0}^{M} w_i V_{n-i} \tag{4}$$

where V_n, Z_n are the input and output signals respectively and w_i are the weights of the filter.

The conventional binary implementation of the M-tap FIR filter of Fig. 3, requires $M - 1$ delay elements, M binary multipliers of $m{+}l$ bit-length, where m and l are the input signal's and the coefficient's bit resolution respectively, followed by an adder of sufficient number of bits to account for the accumulation. Typically, this value is set to $m{+}l{+}\log_2(M){-}1$ according to [11].

In our case, the SDM's $0, 1$ encoding of the input signal allows for the M binary multiplications to be replaced by M XNOR gates. Regarding the summation, the standard method

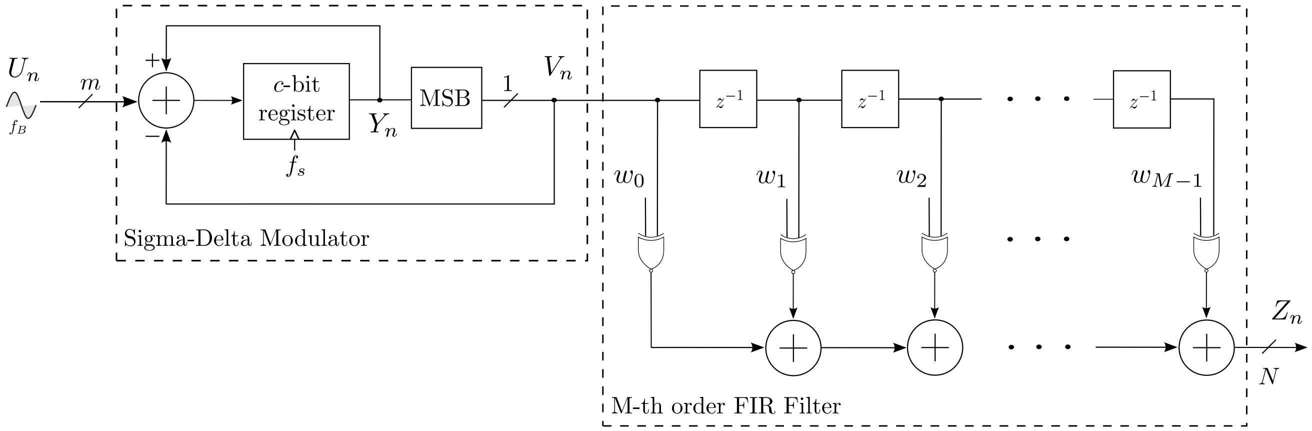

Fig. 3. Proposed SDM-SC processing scheme. The first order digital SDM encodes a multi-bit input signal into a single-bit one where 0 and 1 carry the information of -1 and 1 respectively. The sequence is then processed by a SC-based M-tap FIR filter

used in SC is the MUX, which is a hardware-demanding block due to the additional SNG required for each adder. Instead, we consider a binary adder of $N = \lceil log_2 M \rceil$-bits given the fact that Z_n belongs in $\{0, 1, \ldots, M-1\}$.

C. Stochastic Coefficient Generation

To represent the filter coefficients w_i of the architecture of Fig. 3 using stochastic sequences, M SNGs are required. The M SNGs though, require M LFSRs of k-bits, which alongside with their respective comparators will increase dramatically the overall utilization. Simply sharing LFSRs, i.e. using a single LFSR as the random number generator for all SNGs, is proven to introduce maximal correlation [6] between the generated sequences and as expected it will degrade the signal's quality.

To optimize the architecture's hardware resources, we employ the LFSR circular shifting scheme of [6] shown in Fig. 4. The circular shifting exploits the fact the LFSR cycles through all its values within $\mathcal{R}_S$ only once. As such, sequences are generated in parallel without being maximally correlated. Assuming that the LFSR's current binary value is $R_{n,i}$ and recalling that it takes values within $\mathcal{R}_S$, each circular shift block of Fig. 4 produces the next value $R_{n,i+1}$ as

$$R_{n,i+1} \triangleq R_{(n-s,i)_N} = R_{n-s,i} \, (\bmod N) \,, \tag{5}$$

where $i = 1, 2, \ldots, M$ and $s \in \mathbb{N}^*$ as long as $s < k$ holds.

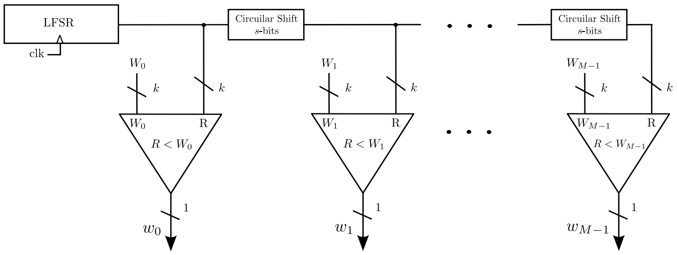

Fig. 4. SNG sharing scheme with circular shifting [6]

IV. Simulation Results

In this section, we evaluate the proposed SDM-SC processing scheme with simulations using Matlab. We use a test sinusoidal input $U_n = \sin(2\pi f_B n)$, where $f_B = 4kHz$ and an oversampling rate of $OSR = 256$. The FIR filter we consider here has 5-taps with corresponding weights selected as $w_0 = 0.7, w_1 = 0.6, w_2 = 0.9, w_3 = 0.6$ and $w_4 = 0.7$.

Understandably, the choice of the sequence length $N = 2^k$ plays an important role in the quality of the processed signal, as it is directly associated with 1) the number of samples taken into account in the output Z_n and 2) the resolution of the generated stochastic numbers. Therefore, increasing the number of samples leads to increased performance. The value of the sequence length selected here is $N = 2^{15}$.

To investigate the proposed processing scheme's performance in the frequency domain, we conducted spectral analysis on the filter's output. In Fig. 5 we present the comparison of the one-sided output amplitude spectrum and the amplitude response of the SDM-SC filtering and the conventional one.

It is observed that the fundamental spike of the proposed scheme coincides with the conventional's, as the fundamental harmonic of the output is spotted at $4kHz$, which is exactly the input signal's frequency. Furthermore, its amplitude peak has a value of 3.4837 which is extremely close to the theoretical expected value of $\sum_{i=0}^{4} w_i = 3.5$. With respect to the frequency response, the SDM-SC filter follows the conventional implementation and achieves correctly the -3dB drop in the cut-off frequency.

To investigate the OSR's influence in the SDM-SC processing scheme's performance, we increased its values in powers of two and calculated the corresponding Signal-to-Noise Ratio (SNR) in dB. The results are depicted in Fig. 6. As you may see, the SDM's OSR selection, allows for the SNR to be increased by up to 10dB, achieving approximately 48dB for the OSR value of $1,024$. The binary 5-tap filter counterpart, achieves a 97.21dB performance, considering the round-off noise in the coefficients and the input signal when their bit-resolution is selected to be 15.

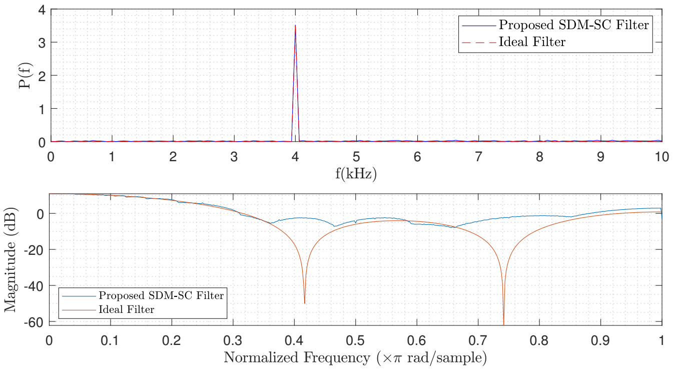

Fig. 5. Comparison of the amplitude spectrum (upper) and amplitude response (bottom) between the proposed SDM-SC processing scheme and the conventional filtering with sequence length $N = 2^{15}$ and OSR $= 256$

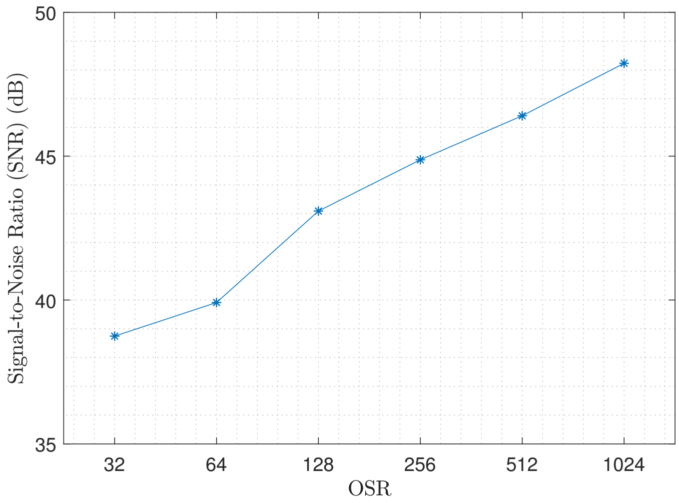

Fig. 6. Proposed SDM-SC scheme's performance in Signal-to-Noise ratio (SNR) in dB for increasing values of OSR in selected powers of two

To highlight the SDM-SC's low-area advantage, in Table I we compare the hardware utilization between proposed approach the conventional binary filter, synthesized in a Kintex-7 KC705 evaluation device. Here we considered a k = 15-bit resolution that corresponds to 1) the input signal U_n, 2) the coefficients w_i and 3) the LFSR on the proposed approach. Also, the oversampling ratio selected is OSR = 256.

As expected, the SDM-SC filtering utilizes negligible area, namely 28 slice LUTs and 35 slice registers compared to the conventional binary of 698 slice LUTs and 60 registers, which is due to the replacement of the binary multipliers with the XNOR gates. Moreover, it has also the advantage of maintaining the same max operating frequency of 667MHz. Note that for the conventional FIR filtering, the DSP blocks are converted into LUT equivalents so as to have a uniform comparison between the two approaches.

V. Conclusion

In this work we presented a soft-filtering DSP architecture based on SDMs and SC. It was shown that the encoding of a multi-bit signal into a single-bit one by exploiting a first order digital SDM, allows to use SC elements and to benefit from their small area advantages. Simulation results showed that the SDM-SC processing scheme offers satisfactory performance in terms of SNR, while the FPGA synthesis results demonstrated negligible FPGA area occupation compared to the traditional binary approach. To conclude, the proposed SDM-SC architecture's trade-off can be considered in large designs where the total area is constrained by the rest processing blocks.

TABLE I
FPGA Hardware Utilization Comparison between the SDM-SC and the Conventional 5-Tap FIR Implementation

	SDM-SC Filter	**Conv. Filter**
Max Operating Frequency (MHz)	667	667
Slice LUTs [Used / Util.]	29 / 0.01%	698 / 0.34%
Slice Registers [Used / Util.]	35 / 0.01%	60 / 0.02%

Acknowledgment

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the HFRI PhD Fellowship grant (Fellowship Number:1216).

References

[1] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 8, pp. 1515 – 1531, Aug. 2018.

[2] B. R. Gaines, *Stochastic Computing Systems*. Springer, Boston, MA, 1967.

[3] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, "A survey of stochastic computing neural networks for machine learning applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. early access, pp. 1 – 16, Aug. 2020.

[4] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Transactions on Very Large Scale Integration Systems (VLSI)*, vol. 2, no. 3, pp. 449–462, Apr. 2014.

[5] Y. Liu and K. K. Parhi, "Architectures for recursive digital filters using stochastic computing," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3705 – 3718, Jul. 2016.

[6] H. Ichihara, T. Sugino, S. Ishii, T. Iwagaki, and T. Inoue, "Compact and accurate digital filters based on stochastic computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 1, pp. 31 – 43, 2019.

[7] M. Alawad and M. Lin, "Fir filter based on stochastic computing with reconfigurable digital fabric," in *IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, Vancouver, BC, Canada, May 2015.

[8] N. Saraf, K. Bazargan, D. J. Lilja, and M. D. Riedel, "Iir filters using stochastic arithmetic," in *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, Germany, Mar. 2014.

[9] M. H. Najafi, D. Jenson, D. J. Lilja, and M. D. Riedel, "Performing stochastic computation deterministically," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 12, pp. 2925 – 2938, Dec. 2019.

[10] N. Temenos and P. P. Sotiriadis, "Deterministic finite state machines for stochastic division in unipolar format," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, Oct. 2020.

[11] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 4th ed. Springer, 2014.