A New Stochastic Inner Product Core Design for Digital FIR Filters

A New Stochastic Inner Product Core Design for Digital FIR Filters

Ming Ming Wong[1,⋆], *M. L. Dennis* Wong[2], *Cishen* Zhang[3], and *Ismat* Hijazin[3]

[1] *Faculty of Engineering, Computing and Science, Swinburne University of Technology, Sarawak Campus, Malaysia.*
[2] *Heriot Watt University Malaysia, Wilayah Persekutuan Putrajaya, Malaysia.*
[3] *School of Software and Electrical Engineeing, Swinburne University of Technology, Hawthorn VIC 3122, Australia.*

Abstract. Stochastic computing (SC) is a computational technique with computational operations governed by probability instead of arithmetic rules. It recently found promising applications in digital and image processing areas and attracted attentions of researchers. In this paper, a new stochastic inner product (multiply and accumulate) core with an improved scaling scheme is presented for improving the accuracy and fault tolerance performance of SC based finite impulse response (FIR) digital filters. The proposed inner product core is designed using tree structured multiplexers which is capable of reducing the critical path and fault propagation in the stochastic circuitry. The designed inner product core can lead to construction of SC based light weight and multiplierless FIR digital filters. As a result, an SC based FIR digital FIR filter is implemented on Altera Cyclone V FPGA which operates on stochastic sequences of 256-bits length (8-bits precision level). Experimental results show that the developed filter has lower hardware cost, better accuracy and higher fault tolerance level compared with other stochastic implementations.

1 Introduction

Stochastic computing (SC) [1] is a computational technique with operations based on probability instead of arithmetic rules [2–4]. This technique can simplify mathematical functions, which are computationally demanding in binary computation, with simple logic operations and reduced hardware requirement. It is robust against noise. And it has a progressive precision characteristic that the precision of stochastic numbers (bit streams) increases as computation proceeds [2]. These advantages enabled SC's recent applications in signal and image processing, in particular, in realization and implementation of digital filters [5, 6]

In this paper, a new stochastic inner product core with an improved scaling scheme is presented for improving the accuracy and fault tolerance performance of SC based finite impulse response (FIR) digital filters. The proposed inner product core is designed using tree structured multiplexers which is capable of reducing the critical path and fault propagation in the stochastic circuitry. The designed inner product core can lead to construction of SC based light weight and multiplierless FIR digital filters.

The performance of the proposed SC based FIR filter is tested via hardware implementation on an FPGA using a case study on a $6th$-order FIR digital filter. With the filter's order varying from $4th$ to $8th$ order and each having cutoff frequencies ranging from 0.2π to 0.8π, empirical analysis is performed to evaluate the proposed FIR filter's accuracy levels, fault tolerance capabilities as well as the associated hardware costs. The obtained results show that our proposed SC filter design outperforms other existing higher precision stochastic FIR filter designs.

The rest of the paper is organized as follows. Section 2 reviews stochastic computational elements of SC. The motivation and problem statement of proposed SC based FIR filter design are presented in Section 3. Next, an improved stochastic inner product is presented in Section 4. The proposed function is later employed to design a new SC digital FIR design and the case study is reported in Section 5. The experimental results (accuracy and fault tolerance analysis) as well as the hardware implementation for the application is reported and discussed in Section 6. Finally, some conclusion remarks are drawn in Section 7.

2 Basic Theory of Stochastic Computation

The basic rule of SC is that the computational data (in bit-streams) are represented as stochastic sequences and are processed in form of digitized probabilities [3]. Naturally, the representations and all the involved computations always lie within the real-number unit interval [0, 1]. Stochastic representation can be coded in two formats: *SC-unipolar* and *SC-bipolar* [1].

In SC-unipolar format, the input s is a real number within the unit interval, i.e. $0 \leq s \leq 1$. As an example, a 2's complement binary input bit-stream $\{0100\}_2$ is represented in stochastic bit-streams S, consisting of 4 of bit '1' out of $2^4 = 16$ bits (remaining bits are zeros). This stochastic bit-streams S, is also interpreted as $p = P(S = 1) = 4/16$. On the other hand, in SC-bipolar format, the range of the real number input, s, is extended to

⋆Corresponding author: wmingming7@gmail.com

$-1 \leq s \leq 1$. Consider a 2's complement binary input bit-stream $\{1100\}_2$. In SC-bipolar bit-streams S, the deterministic value is mapped to $p = P(S = 1)/2 = 12/(16 \times 2) = 6/16$.

In other words, stochastic representation observes the probability of 1s at arbitrary bit position in S. Such representation serves as the main reason for having high fault tolerance in SC. A single bit-flip in a long bit-stream causes a minor change in original logical value. On the contrary, a single bit-flip in the conventional 2's complement computation will result in huge error especially if the bit-flip occurs on the higher-order bit.

Multiplication of two inputs streams, which is computational intensive in conventional signed binary computing, can be performed using single logical gate in SC. Consider two stochastic input bit-streams, X_1 and X_2 and the output for their multiplication, Y, is derived as,

$$\begin{aligned} y &= P(Y = 1) \\ &= P(X_1 = 1)P(X_2 = 1) \\ &\quad +(1 - P(X_1 = 1))(1 - P(X_2 = 1)) \end{aligned}$$

Stochastic multiplication in bipolar format is clearly a logical XNOR operation between input bit-streams, X_1 and X_2 in digital circuit. For unipolar format, the multiplication is performed using a logical AND operation instead. Stochastic multiplier for both unipolar and bipolar formats are as depicted in Figure 1.

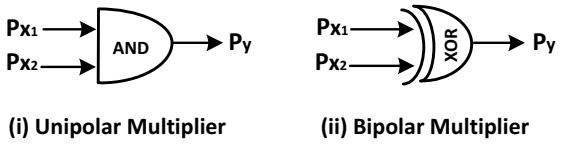

Figure 1. Stochastic Multiplier for (i) SC-unipolar and (ii) SC-bipolar formats.

Addition in SC is performed using a special operation, termed as scaled addition. The addition is scaled such that the value always lies between the probability interval $[0, 1]$. With S being a constant scale, the sum of two independent stochastic bit-streams X_1 and X_2, produces Y, defined as,

$$\begin{aligned} y &= P(Y = 1) \\ &= P(S = 1)P(X_1 = 1) + (1 - P(S))(P(X_2 = 1)) \\ &= SX_1 + (1 - S)X_2 \end{aligned}$$

Thus, multiplexer with conditional select line S, set as $P(S) = \frac{1}{2}$ can be used to realize the scaled addition of two stochastic bit-streams in digital circuit. **Subtraction** in SC is similar to the adder except that the stochastic scaled substractor requires an additional inverter and this only feasible in SC-bipolar format. Both the stochastic scaled adder and scaled substractor are illustrated in Figure 2.

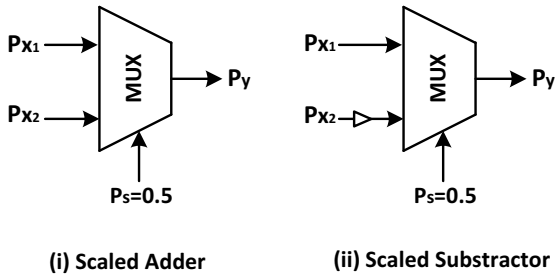

Figure 2. Stochastic scaled adder/substractor

3 Problem Statement

FIR filter is widely used in many classical DSP applications in order to achieve filtering stability and linear-phase property. FIR filters are generally characterized by their impulse response coefficients which perform multiplication with the input signals, i.e. the inner product.

Alternatively, these computationally expensive operations can be well approximated through SC. To be precise, the summation of the multiplication between input vectors $\{X_0, X_1\}$ and filter coefficients $\{a_0, a_1\}$ can be derived using a single stochastic operation, the *stochastic scaled addition*, i.e. $\frac{1}{2}\left(a_0X_0 + a_1X_1\right)$.

Through SC, the computational data are represented in stochastic bit-streams of 2^n bits (n is precision level) and are processed in the form on digitized probabilities [2]. In terms of hardware, a stochastic scaled addition can be realized using a *multiplexer* with its conditional select line, S set as the scaling factor [2].

Unfortunately, when repetitive computations are involved, the implicit scaling of $\frac{1}{2}$ in stochastic scaled addition will severly degrade the filter's output accuracy [5]. An alternative stochastic inner product architecture was reported in [5, 6] where the scaling is performed using with unevenly weightings. However, significant accuracy degradation is observed as the filter's order increases. Therefore, to address this issue, a new scaling scheme in stochastic inner product is required.

4 An Improved Stochastic Inner Product

In this work, an improved stochastic inner product is designed using a new scaling scheme which considers the weight distribution of the filter's coefficients. Under this scheme, the coefficients of equal (or near equivalent) weightings are paired together and form the scaling factor in the stochastic scaled addition. For instance, coefficients $\{a_0, a_1\}$ is paired when $a_0 \approx a_1$ and this produce scaling factor $\frac{|a_0|}{|a_0|+|a_1|}$.

A causal discrete-time FIR filter of order N (length $M = N+1$) is generally described as $y[n] = \sum_{k=0}^{N} a[k]x[n-k]$, with $y[n]$ as the output signal, $x[n]$ is the input signal and $\{a_0, a_1, a_2, a_3, \ldots, a_N\}$ are the filter coefficients. All the 4 types of linear phase FIR filters have symmetric parameters in absolute value. Therefore, the distribution of the absolute value of the filter's impulse response

coefficients resembles a bell curve. The largest value is weighted at the center of the distribution and decreases gradually towards the first and the last coefficients, i.e $a_0 \approx a_N < a_1 \approx a_{N-1} < a_2 \approx a_{N-2} < \cdots < a_{\frac{N-1}{2}}$. Hence, the scaled additions are performed according to the pairs of the FIR filter coefficients arranged such as follows.

- $P_0 = \{a_0, a_N\}, P_1 = \{a_1, a_{N-1}\}, P_2 = \{a_2, a_{N-2}\}, \ldots$ and
- $P_{N'=\frac{N-1}{2}} = \left\{a_{\frac{N-1}{2}}, a_{\frac{N+1}{2}}\right\}$ for even length (such as FIR Types II and IV),
- $P_{N'=\frac{N}{2}} = \left\{a_{\frac{N}{2}}\right\}$ for odd length (such as FIR Types I and III).

Furthermore, note that $P_0 < P_1 < P_2 < \cdots < P_0 < P_{N'}$ with $N' = \frac{N-1}{2}$ for even filter length and $N' = \frac{N}{2}$ for odd filter length. With that, the next round of scaled additions are performed following the pairing shown below.

- $P'_0 = \{P_0, P_1\}, P'_1 = \{P_2, P_3\}, \ldots$ and
- $P'_{N''=\frac{N'-1}{2}} = \left\{P_{N'-1}, P_{N'}\right\}$ for N' is odd,
- $P'_{N''=\frac{N'}{2}} = \left\{P_{N'}\right\}$ for N' is even.

The similar addition process is repeated until the inner product computation is completed. An example of the resultant stochastic inner product using the proposed approach is shown in (7).

5 Case Study of New SC FIR Filter Design

Consider a 6*th*-order linear phase Type I FIR filter with its taps coefficients labeled as $\{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$ and the input vectors listed as $\{X_0, X_1, X_2, X_3, X_4, X_5, X_6\}$. The inner product of the filter is derived as $y = a_0X_0 + a_1X_1 + a_2X_2 + a_3X_3 + a_4X_4 + a_5X_5 + a_6X_6$.

Using the proposed stochastic inner product, the final computation is described in (7) and is illustrated in Figure 3. Note that, both of the input vectors and filter coefficients are first converted into stochastic bit-streams using SNG modules [1], which are not shown in the figure.

$$Y_{11} = \Big(\frac{|a_0|}{|a_0| + |a_6|}\Big)X_0 + \Big(\frac{|a_6|}{|a_0| + |a_6|}\Big)X_6 \quad (1)$$

$$Y_{12} = \Big(\frac{|a_1|}{|a_1| + |a_5|}\Big)X_1 + \Big(\frac{|a_5|}{|a_1| + |a_5|}\Big)X_5 \quad (2)$$

$$Y_{13} = \Big(\frac{|a_2|}{|a_2| + |a_4|}\Big)X_2 + \Big(\frac{|a_4|}{|a_2| + |a_4|}\Big)X_4 \quad (3)$$

$$Y_{14} = \Big(|a_3|X_3\Big) \quad (4)$$

$$Y_{21} = \Big(\frac{|a_0| + |a_6|}{|a_0| + |a_1| + |a_5| + |a_6|}\Big)Y_{11} + \Big(\frac{|a_1| + |a_5|}{|a_0| + |a_1| + |a_5| + |a_6|}\Big)Y_{12} \quad (5)$$

$$Y_{22} = \Big(\frac{|a_2| + |a_4|}{|a_2| + |a_3| + |a_4|}\Big)Y_{13} + \Big(\frac{|a_3|}{|a_2| + |a_3| + |a_4|}\Big)Y_{14} \quad (6)$$

$$Y = \Big(\frac{|a_0| + |a_1| + |a_5| + |a_6|}{|a_0| + |a_1| + |a_2| + |a_3| + |a_4| + |a_5| + |a_6|}\Big)Y_{21} + \Big(\frac{|a_2| + |a_3| + |a_4|}{|a_0| + |a_1| + |a_2| + |a_3| + |a_4| + |a_5| + |a_6|}\Big)Y_{22} \quad (7)$$

With such filter's coefficients, $a_0 \approx a_6, a_1 \approx a_5$ and $a_2 \approx a_4$, the scaled addition in (1), (2) and (3) can be performed using fixed scaling factor $\frac{1}{2}$. Therefore, the conditional probability selection line (which is determined by the scaling factor) of the correspondence multiplexers can share the same SNG modules to promote hardware cost reduction. The savings will be more prominent in higher order filter where there is a large amount of identical coefficients. In addition, our SC FIR filter is designed with precision level of 8-bits, whereby the computations are performed using $2^8 = 256$ bits only. The filter designs in [5, 6] are computed using $2^{10} = 1024$ bits instead.

6 Experimental Results

Several simulations were performed to test the effectiveness and the efficiency of the proposed SC FIR filter. The metric of measurements included the output accuracy (error-to-signal power ratio), the fault tolerance and the hardware requirement and performance in FPGA implementation.

6.1 Accuracy Analysis

The new SC low-pass FIR filters, implemented in three different orders and each having four different cutoff frequencies, are evaluated for their accuracy levels. A total of 256 samples of input test signal is used in the test simulation. The test signal consists of a mixture of four sinusoidal waves of different frequencies padded with white noise. The accuracies of the proposed filters are measured in term of the error-to-signal power ratio and are benchmarked with the work reported in [5]. These results are as summarized in Table 1.

The results from [5] showed the error ratio increases with higher filters' order. In constrast, our SC FIR filter presents consistently lower error ratio regardless of the filters' order. Further accuracy justification can be deduced by comparing the frequency response and the power spectral density (PSD) of the output signal deduced from both our SC filter and the ideal filter (see Figure 4). It is observed that the spectrum of our SC filter is very close to that of the ideal filter.

6.2 Fault Tolerance Analysis

Apart from low hardware cost, SC is well recognized for being insusceptible towards fault as opposed to the conventional binary computing. Fault tolerance testing is conducted on our proposed SC 6*th*-order FIR filter with cutoff frequency at 0.4π. The test is performed by randomly injecting various percentage of bit-flipping error in the input signals and the corresponding error-to-signal power ratio is measured and summarized in Table 2.

The results show that percentage of random bit-flipping error ranging from 0.5% to 3.0% has minimal

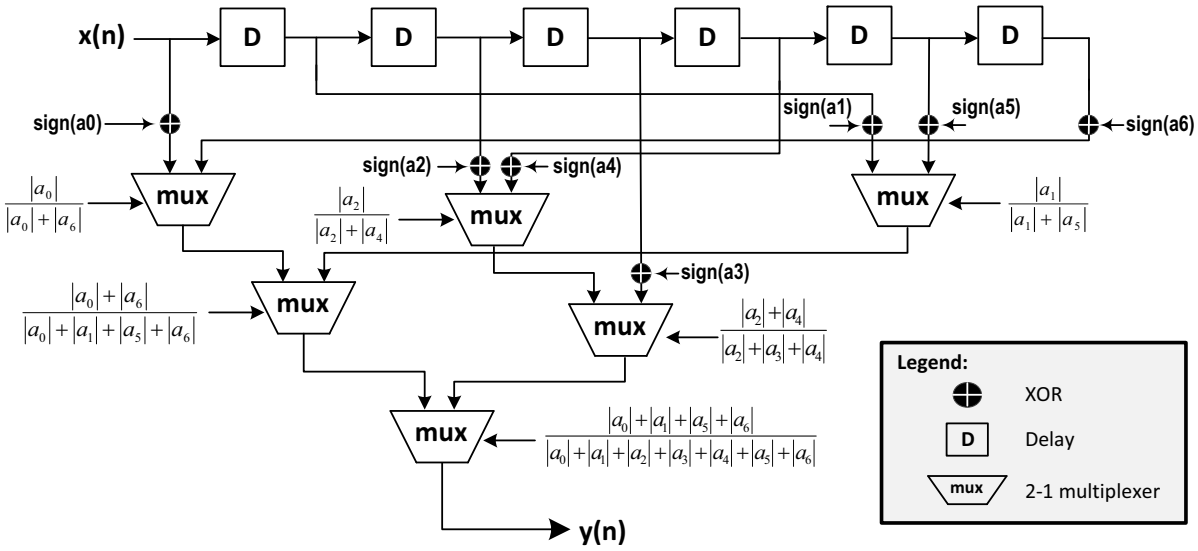

Figure 3. The new SC 6*th*-order FIR filter.

Table 1. Accuracy test results of (i) our proposed design (precision level of 8-bits) compared with (ii) [5] (precision level of 10-bits).

Filter Order	Filter Cutoff Frequency							
	0.2π		0.4π		0.6π		0.8π	
	(i)	(ii)	(i)	(ii)	(i)	(ii)	(i)	(ii)
2	0.0050	0.0037	0.0046	0.0025	0.0058	0.0013	0.0069	0.0004
4	0.0192	0.0597	0.0123	0.0314	0.0120	0.0465	0.0070	0.0145
6	0.0136	0.0648	0.0075	0.0462	0.0080	0.0637	0.0048	0.0626
8	0.0095	-	0.0107	-	0.0114	-	0.0076	-

Table 2. Error-to-signal power ratio analysis resultant from various percentage of random bit-flipping error in 6*th*-order FIR filter with cutoff frequency at 0.4π.

Filter Implementation	Percentage of Bit-Flipping					
	0.5%	1.0%	1.5%	2.0%	2.5%	3.0%
Our Work	0.0076	0.0160	0.0209	0.0292	0.0355	0.0514
Conventional Filter	0.1180	0.1874	0.3033	0.3351	0.4409	0.4843
Direct Form [6]	0.0488	0.0540	-	-	-	-
Lattice Form [6]	0.0563	0.0820	-	-	-	-

Table 3. Hardware review for the FPGA implementation of the SC 6*th*-order FIR filter with cutoff frequency at 0.4π.

Hardware Requirement/ Performance	**SC FIR Filter**	Inner Product Core	SNG Module
Combinational ALUTs (112,960)	**128**	4	26
Memory ALUTs (56,480)	**0**	0	0
Dedicated Logic Register (225,920)	**159**	1	33
Total Register (225,920)	**159**	1	33
Fmax (MHz)	**306.0**	0	429.92
Dynamic Thermal Power Dissipation (mW)	**2.11**	0.04	0.94

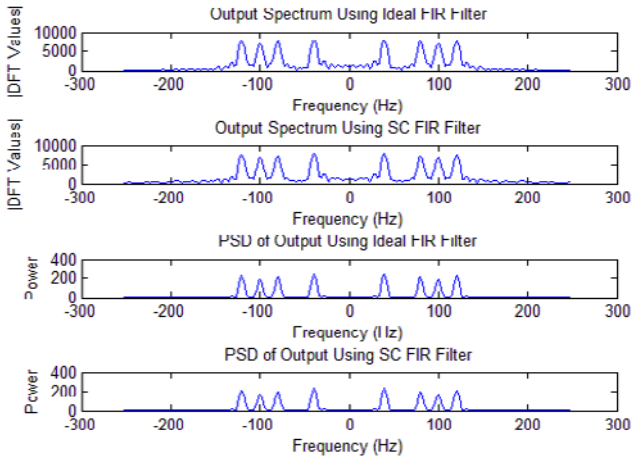

Figure 4. Output spectrum and PSD derived using the FIR ideal filter and our SC FIR filter. Both filters are low-pass with 6*th*-order and the cutoff frequency at 0.4π.

impact on the output accuracy of our proposed SC FIR filter. On the contrary, the conventional ideal FIR filter shows significant accuracy degradation as the injected bit-flipping error increases at every 0.5%. These results are further benchmarked with the work reported in [6]. The authors presented two SC 7*th*-order FIR filter with cutoff frequency at 0.1π using direct form and lattice form. Both of their filters also exhibited higher error percentages in comparison to our work.

The multiplexers in our proposed inner product core are positioned in tree structure to avoid error propagation that tends to occur in long critical path. Therefore, with short critical path, the presented SC FIR filter has higher fault tolerance in comparison to the conventional FIR filter as well as the existing SC FIR filters.

6.3 Hardware Complexity

The proposed SC FIR filter is implemented in Cyclone V 5CGXFC7D6F31C6 using Quartus II 11.1. The full hardware synthesis result of the filter as well as its core units; the inner product and the SNG Module are summarized in Table 3.

7 Conclusion

A case study of a new SC FIR filter design using an improved stochastic inner product core was presented in this paper. Without the use of multiplier, the inner product core unit employed stochastic scaled addition with a new scaling scheme that paired the filter's coefficients in according to their weightage. The computation was realized using multiplexers positioned in tree structure, which in turn reduces the critical path as well as the fault propagation in the stochastic circuitry. Such design enhanced the computational accuracy and offered high fault tolerance in SC filter system. For hardware evaluation, a new SC 6*th*-order FIR filter with the cutoff frequency at 0.4π on FPGA platform has been implemented and tested. Experimental results have shown that the presented SC FIR filter outperforms the conventional filter and the existing works in both metrics and also has low hardware cost.

References

[1] B. R. Gaines,'Stochastic computing', *Proceedings of the Spring Joint Computer Conference, New York, NY, USA*, pp. 149-156,(1967).

[2] A. Alaghi, and J. P. Hayes,'Survey of Stochastic Computing', *ACM Trans. Embed. Comput. Syst.*, vol. **12**, no. 2, pp. 19, (2013), .

[3] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, 'An architecture for fault-tolerant computation with stochastic logic',*IEEE Transactions on Computers*, vol. **60**, no. 1, pp. 93-105, (2011).

[4] B. Moons, and M. Verhelst, 'Energy-efficiency and accuracy of stochastic computing circuits in emerging technologies', *IEEE Journal on E,merging and Selected Topics in Circuits and Systems*, vol. **4**, no. 4, pp. 475-486, (2014).

[5] Y. N. Chang, and K. K. Parhi, 'Architectures for digital filters using stochastic computing', *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2697-2701, (2013).

[6] Y. Liu, and K. K. Parhi,'Lattice fir digital filter architectures using stochastic computing', *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1027-1031, (2015).