HIGH-ACCURACY STOCHASTIC COMPUTING-BASED FIR FILTER DESIGN

Kazi J. Ahmed, Bo Yuan, Myung J. Lee

Department of Electrical Engineering, City College, City University of New York, USA

ABSTRACT

In this paper we propose a novel Stochastic Computing (SC) Finite Impulse Response (FIR) filter design to improve the overall accuracy. Since SC is generated probabilistically, it incurs inherent SC process error, i.e. the larger number of SC process blocks, the larger error probability. Moreover, generation of SC from lower binary radix value produces higher percentage of error as compared to generation from higher value. In our SC FIR design, we reduce the number of SC processing blocks by reducing the number of FIR stages using fast FIR algorithm (FFA). Further we improve the inner-product performance by re-ordering the FIR filter parameters to generate higher-value SC input. Confirmed by the numerical analysis and simulation results, our novel design can implement higher-order FIR filter without reducing its output performance.

Index Terms—Stochastic Computing (SC), Finite Impulse Response (FIR) filter, High order FIR filter, Linear Phase, Fast FIR Algorithm (FFA)

1. INTRODUCTION

Digital filter is considered as one of the major fundamental blocks for Digital Signal Processing (DSP). Out of the two filters, namely, 1) Infinite Impulse Response (IIR) and 2) Finite Impulse Response (FIR); FIR filter is deemed better because of its linear phase and non-recursive property. For the same reason, FIR filter is more resilient to phase shift and has higher stability than the IIR filter. However, FIR filter also has its own shortcomings. FIR filter requires higher order (tap) than IIR to achieve the same spectral performance. Hence, for practical implementation, higherorder FIR filter (more stages) is considered so as to meet the design specification.

Because of its simple hardware requirements, Stochastic Computing (SC) FIR filter is gradually gaining its pace. However, SC design innately incurs errors since it is generated probabilistically. As a result, high-order SC FIR design is inherently error prone irrespective of the performance of individual SC module. So far, most of the recent works focus mainly on the performance of SC unit such as inner-product or adder without considering much of the reduction of SC processing blocks. Consequently, the design of high-accuracy high-order SC FIR filter is still very challenging.

This research is supported by MSRDC (D01_W911SR-14-2-0001-0016)

In our proposed SC FIR design, however, we focus mainly to reduce the number of SC blocks so as to reduce SC error. We primarily reduce the number of SC blocks by applying fast FIR Algorithm (FFA) [9]. For a *q*-by-*q* FFA, the *m*-tap FIR filter could virtually be reduced to *m/q*-tap, resulting in lowering the SC error significantly. Moreover, our design smartly selects higher SC value over lower one, since lower value SC produces higher percentage error. Simulation results show that the proposed approaches can significantly increase the accuracy of higher-order SC FIR filter, thereby paving the way of promoting stochastic DSP modules in practical applications.

2. BACKGROUND ON SC AND FIR FILTER

In this section, we provide some background on SC and FIR filter before putting forth our proposed work.

2.1 Stochastic Computing(SC)

Stochastic Computing is attractive because of its low hardware complexity. Derived from the conventional binary radix computing (BC), it represents any value by a stream of bits ('1's and '0's). However, before converting, the BC value has to be made within [0,1] for unipolar and [-1, 1] for bipolar. In case of unipolar, any value x in BC is probabilistically converted to a length of *N*-bit stream containing *X* bits of '1's so that x = P(X) = X/N (Fig. 1(a))[1-3].



Figure 1. Some basic SC presentations and operations

On the other hand, any bipolar representation of x in SC for a length of N-bit stream is presented by X bits of '1's such that x = 2(X/N)-1(Fig.1 (c) [1-3]. As seen from Fig. 1 the multiplication and addition can be done using only digital logic gates. Moreover, SC is not limited to just multiplication and addition, for some other arithmetic operations readers are referred to [1-5].

2.2 Finite Impulse Response (FIR) filter

FIR filter is one of the major DSP modules used for wireless communication system. The output sequence of an *m*-tap FIR filter can be expressed in time domain as

$$y(n) = x(n)h_0 + x(n-1)h_1 + x(n-2)h_2 + \cdots$$

... + x(n-m+1)h_{m-1}
= $\sum_{i=0}^{m-1} x(n-i)h_i$ (1)

where x(n) and y(n) are the input and output of the filter respectively at time slot *n* and h_i are the filter coefficients.

Since an *m*-tap FIR filter requires *m* high- complex BC multipliers, the area as well as the power consumption is increased significantly. On the other hand, an SC multiplier requires only an AND gate to perform the same job. Motivated by this fact, both direct-form and lattice-form of SC FIR filters were proposed [2,6,7,8,11]. However, as the SC are generated and process in probabilistic domain, the accuracy of the design may be degraded significantly, especially in case of higher-order FIR filter.

3. PROPOSED WORK

To improve the accuracy of SC unit such as scaled down adder (Fig. 1 (e-f)) of FIR filter, several FIR filter designs were already proposed [2,6,7,8,11]. However, for a FIR filter with higher order, the accuracy of its filtering output is still not satisfactory. Even though newer designs provide improved accuracy, the core problem of SC FIR filter still lies on the probabilistic approach of calculation.

3.1. Previous works on SC FIR filter

3.1.1. Inner-product based SC FIR filter design

Since the basic SC adder (SA) produces scaled down output, the long-tap (higher-order) FIR filter based on SA suffers accuracy loss significantly. To alleviate the scaled down effect, inner-product-based FIR design was proposed in [2]. The inputs and output of a 2-input inner-product from [2] are shown in Fig. 2, where the sign of a is represented by s(a).



Figure 2. Inner-product module proposed in [2]

This module depends on the assumption that the denominator of the output $(|a_0|+|a_1|)$ is less than or close to 1. Moreover, the design also assumes that the value of *h* parameters is pre-known. However, from practical point of view, the values of *h* parameters toned to be adjusted time to time. Therefore, this design is not efficient for higher-order FIR filter.

3.1.2. Non-scaled Stochastic FIR filter design

Another FIR filter design that attempts to reduce the scaled down effect of SA was proposed in [8]. Here the SC value of x is presented using two lines of length-N bit streams, separating the magnitude $M(X_i)$ and sign $S(X_i)$ to represent $x = \frac{1}{N} \sum_{i=0}^{N-1} [1 - 2S(X_i)M(X_i)]$. This design considers the basic direct-form FIR filter using its two-lines SC Multiplier (TSM) and two-line SA (TSA). While the TSM uses simple XOR and AND gates, the TSA requires combinational logic circuit, counter, along with XOR and AND gates as shown in Fig. 3.



Figure 3. Adder circuit for TSA from [5]

The accuracy of the TSA claimed to be independent of the number of stage [8], however, each stage requires significant amount of area and power consumption. Consequently, it limits the number of stage of FIR filter and makes it inefficient for practical implementation.

3.1.3. Accumulative Parallel Counter based FIR filter

Yet another stochastic FIR filter was proposed in [11] using Accumulative Parallel Counter (APC) [12]. However, each of these APC adders requires many Full Adders (FAs) and Half Adders (HAs), resulting in much more hardware complex. Consequently, it consumes greater area and power and is not efficient for higher-order SC FIR filter design.

The above analysis for the state-of-the-art works shows that the practical FIR filter implementation (with higher order) using SC is still very challenging. Moreover, the prior efforts focused on higher accuracy of SC unit, not attempting to reduce the sources of SC errors.

3.2. Proposed SC FIR filter design

3.2.1. Stochastic Fast FIR Algorithm (FFA)

The major problem with all of the above SC FIR filter designs is that they suffer because of numerous stages of probabilistic SC processing. We proposed a FFA-based SC FIR filter that can reduce the number of stage significantly.

In general, the *q*-by-*q* FFA approach can reduce an *m*-tap FIR filter to m/q-tap parallel sub-filters. For instance, the *z*-transform of a 2-by-2 FFA filter with input *x*, filter parameters h_i and output *y* can be expressed as $X(z) = X_e(z^2) + z^{-1}X_o(z^2)$, $H(z) = H_e(z^2) + z^{-1}H_o(z^2)$ and $Y(z) = Y_e(z^2) + z^{-1}Y_o(z^2)$ respectively [9]. The schematic diagram for this FFA is shown in Fig. 4. Since even $X_e(z^2)$ and odd $X_o(z^2)$ time-series inputs are needed at the same time, the sampling time for the FFA filter is two time units (z^2) . However, both the even $Y_e(z^2)$ and the odd $Y_o(z^2)$ time-series outputs are available at each sampling time, resulting in no additional time delay.

For a 2-by-2 FFA filter, h parameters of each m/2-tap subfilters in z-domain can be expressed as



Figure 4: Schematic diagram of 2-by-2 FFA

As seen from Fig. 4, the outputs of each of the *H* blocks are produced from its even and odd *x* time-series inputs (X_e and X_o) and *h* parameters (H_e and H_o). Hence the outputs of these four *H* blocks in time domain can be expressed as:

$$\begin{split} X_e H_e &= x(0)h_0 + x(2)h_2 + x(4)h_4 + \cdots \\ X_e H_o &= x(0)h_1 + x(2)h_3 + x(4)h_5 + \cdots \\ X_o H_e &= x(1)h_0 + x(3)h_2 + x(5)h_4 + \cdots \\ X_o H_o &= x(1)h_1 + x(3)h_3 + x(5)h_5 + \cdots \end{split}$$

From the above expressions it is clear that the output of each block is the inner product of some x and h values. To generalize these expressions, we map the h parameters to a_1 , a_2 , ... and x inputs to x_1 , x_2 , ... as presented in the Table 1 and 2.

Table 1. Mapping of a with h of H_e and H_o

General a parameters	a_1	a_2	 $a_{m/2-1}$	$a_{\rm m/2}$
Mapped $H_{\rm e}$ parameters	h_0	h_2	 $h_{\text{m-4}}$	$h_{\text{m-2}}$
Mapped H_0 parameters	$h_{\rm m-1}$	$h_{\rm m-3}$	 h_3	h_1

Table 2.	Mapping	of $x(n)$	to x _r for	each <i>H</i> block
1 4010 2.	mapping	01 20(10)	10 10 101	

	1 0	<u> </u>		
General x_r inputs	x_1	<i>x</i> ₂	 $x_{m/2-1}$	$x_{m/2}$
Mapped for $X_{\rm e}H_{\rm e}$	<i>x</i> (0)	<i>x</i> (2)	 <i>x</i> (<i>m</i> -4)	<i>x</i> (<i>m</i> -2)
Mapped for $X_e H_o$	<i>x</i> (<i>m</i> -2)	<i>x</i> (<i>m</i> -4)	 <i>x</i> (2)	<i>x</i> (0)
Mapped for X_0H_e	<i>x</i> (1)	<i>x</i> (3)	 <i>x</i> (<i>m</i> -3)	<i>x</i> (<i>m</i> -1)
Mapped for X_0H_0	<i>x</i> (<i>m</i> -1)	<i>x</i> (<i>m</i> -3)	 <i>x</i> (3)	<i>x</i> (1)

Using the above generalized x_r inputs and a_r parameters, the output y_b of any H block can be expressed as

 $y_b = x_1a_1 + x_2a_2 + x_3a_3 \dots + x_{m/2-1}a_{m/2-1} + x_{m/2}a_{m/2}$ Since the output of each block has the same equation, the SC circuits for all *H* blocks are exactly the same. The SC circuit for each *H* block for the proposed 12-tap FFA filter is presented in Fig. 5.

The effect of the proposed FFA based FIR filter can be realized from the numerical results shown in Fig. 6. Here the cumulative output error is calculated considering 1% error in BC-to-SC conversion. The error corresponding to No-reorder is from the conventional direct-form inner-product design, while the error corresponding to No-reorderFFA2 is from the inner-product 2-by-2 FFA FIR filter. As seen from the figure, FFA FIR filter is very efficient for reducing the error, especially for the higher-order FIR filter.



Figure 5: Generalize SC FFA FIR circuit for each H block



Figure 6. Error in FIR output from different designs

3.2.2. Improved inner-product module

One drawback of SC is that the bit streams that represent low values are usually associated with high representation errors. As a result, we propose to alter the multiplexer (MUX) control input in SC inner product module from $\frac{|a_0|}{|a_0|+|a_1|}$ to $\frac{|a_1|}{|a_0|+|a_1|}$ (see Fig. 7) to reduce the error.



Figure 7: Inner-product block for $|a_0| < |a_1|$: (a) original Module, (b) modified Module

The effect of this change in the reduction of error is shown in Fig, 8. Notice that in the conventional design ("No *reorder*"), if the $|a_0|$ is getting lesser than $|a_1|$, the value of $\frac{|a_0|}{|a_0|+|a_1|}$ becoming smaller from 0.5 and consequently, generating increasingly higher error. However, in our proposed design this order is changed (*re-order*) selectively so that the value of the control input of MUX can always be maintained at the higher value (>=0.5). It can be seen from Fig. 8 that the re-order approach significantly reduces the output error.



Figure 8. Error in MUX control input

Fig. 6 shows the error performance of FFA-based SC FIR filter when FFA and re-ordering (re-orderFFA2) are jointly performed. It can be seen that the use of the proposed re-order further improves the overall error performance.

However, the comparative values of |a|'s may be not preknown. In that case, the following circuit (Fig. 9) can be used to produce proper |a|'s. Here when $|a_0| > |a_1|$, the comparator outputs is 1 (*Comp out*) and the order is $|a_0|$, $|a_1|$ (Fig. 9 (a)); when $|a_0| < |a_1|$, *Comp out* is 0 and the order is $|a_1|$, $|a_0|$ (Fig. 9 (b))



Figure 9: Circuit to alter $|a_0|$, $|a_1|$ depending on their values

The outputs of the above circuit (re-ordered |a|'s) will be fed to the circuit that will produce the MUX control inputs. Fig. 10 shows one of the circuits that produces MUX control inputs from |a|'s using JK flip-flops and AND gates.



Figure 10. MUX input from JK flip-flop and AND gates

4. SIMULATION RESULTS

To evaluate the performance of the proposed design, we simulate our FIR filter in C programming language. The FIR Impulse Response and the corresponding h parameters are generated in MATLAB R2017. We generate the Impulse Response of a band-pass FIR filter of 12 orders (12-tap) for the LTE downlink channel band 8 (925-960 MHz) [10]. The input of the filter is made from super positioning of

sinusoidal waves of different frequencies. To assess the performance, we also generate the filter outputs from MATLAB and from FIR filter proposed in [2]. To make a fair comparison, MUX control inputs of both the proposed and in [2] are generated using JK flip-flops and AND gates.

Fig. 7 shows the outputs from different FIR filters. Five greens spikes are the filter input, coming from the super position of five sinusoidal waves and the dotted shape is the filter Impulse Response. We generate two outputs from [2]: 1) keeping the *h* parameters in order ("No *re-order*"), 2) changing the order of *h* parameters so as to get high values of MUX control inputs (*re-order*).

Comparing with the outputs of the proposed design and the "No *re-order*" design , the proposed design produces significantly better results. As seen from the figure, the pass-band energy of the proposed design is significantly higher (-3 db) than that of the "No re-order" design (-40 dB). Moreover, the rejection ratio beyond the pass-band of proposed design (37 dB) is higher than that of the "No reorder" (2 dB).



Figure 11: Filter Output in dB for frequency 0 to 2 GHz

Re-ordering of h parameters also produce significantly better results as compared to "No re-order" as seen from Fig. 11, The energy at the pass-band of re-order design is higher(-38dB) than that of the "No re-order" design (-40dB). Again the rejection ratio beyond the band-pass for re-order design on is higher than that of the "No re-order" design (2dB).

5. CONCLUSION

In this paper we propose a new approach that addresses the errors of SC FIR filter and provide corresponding solution. Mainly, we reduce the SC errors by lowering the number of SC processing blocks using 2-by-2 FFA. We also improve the SC error performance by swapping the hparameters to get higher value of MUX control input. Simulation results shows that the combination of these two error reduction approaches can significantly improve the performance compare to existing ones. In future we will design higher order FIR filter using higher parallelism of FFA.

6. REFERENCES

- Y. Oppenheim, R. W. Schafer, J. R. Buck, et. al., Discretetime signal processing, vol. 2. Prentice-hall Englewood Cliffs, 1989.
- [2] Y. N. Chang and K. K. Parhi, "Architectures for digital filters using stochastic computing," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, 2013, pp. 2697-2701
- [3] B. Gaines, "Stochastic computing systems," Advances in Information Systems Science, vol. 2. no. 2. pp. 37-172, 1969
- [4] P. Li, D. J. Lilja, W. Qian, K. Bazargan and M. D. Riedel, "Computation on Stochastic Bit Streams Digital Image Processing Case Studies," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 449-462, March 2014.
- [5] W. Qian, X. Li, M. D. Riedel, K. Bazargan and D. J. Lilja, "An Architecture for Fault-Tolerant Computation with Stochastic Logic," in *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 93-105, Jan. 2011. doi: 10.1109/TC.2010.202
- [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), South Brisbane, QLD, 2015, pp. 1027-1031.
- [7] Y. Liu, K.K. Parhi, "Linear-Phase Lattice FIR Digital Filter Architectures Using Stochastic Logic," *Journal of Signal Processing Systems*, Springer, pp. 1-13, 25 Jan. 2017.
- [8] B. Yuan and Y. Wang, "High-Accuracy FIR Filter Design Using Stochastic Computing," 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, PA, 2016, pp. 128-133.
- [9] K.K. Parhi, VLSI Digital Signal Processing Systems, Design and Implementation, Wiley, Inter-Science, 1999.
- [10] https://en.wikipedia.org/wiki/LTE_frequency_bands, Last visited on Oct 18th, 2017.
- [11] A. Ardakani, F. Leduc-Primeau and W. J. Gross, "Hardware implementation of FIR/IIR digital filters using integral stochastic computation," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, 2016, pp. 6540-6544.
- [12] P. S. Ting and J. P. Hayes, "Stochastic Logic Realization of Matrix Operations," 2014 17th Euromicro Conference on Digital System Design, Verona, 2014, pp. 356-364.