Design and Evaluation of Stochastic FIR Filters

Ran Wang, Jie Han, Bruce Cockburn, and Duncan Elliott
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, AB T6G 2V4, Canada
{ran5, jhan8, cockburn, duncan.elliott}@ualberta.ca

Abstract—**The compact arithmetic units in stochastic computing can potentially lower the implementation cost with respect to silicon area and power consumption. In addition, stochastic computing provides inherent tolerance of transient errors at the cost of a less efficient signal encoding. In this paper, a novel FIR filter design using the stochastic approach based on multiplexers are proposed. The required stochastic sequence length is determined for different signal resolutions by matching the performance of the proposed FIR filter with that of the conventional binary design. Silicon area, power and maximum clock frequency are obtained to evaluate the throughput per area (TPA) and the energy per operation (EPO). For equivalent filtering performance, the stochastic FIR filter underperforms in terms of TPA and EPO compared to the conventional binary design, albeit with some advantages in circuit area and power consumption. The stochastic design, however, shows a graceful degradation in performance with a significant reduction in energy consumption as the stochastic sequences are shortened. The fault-tolerance of the stochastic circuit is compared with that of the binary circuit equipped with triple modular redundancy. It is shown that the stochastic circuit is more reliable than the conventional binary design and its triple modular redundancy (TMR) implementation with unreliable voters, but it is less reliable than the binary TMR implementation when the voters are fault-free.**

Keywords—Stochastic computing; FIR filter; throughput per area; energy per operation; fault tolerance.

I. INTRODUCTION

The importance of finite impulse response (FIR) filters in digital signal processing and the potential benefits of stochastic computing motivated us to explore the possibility of implementing stochastic FIR filters. Both manufacturing variations and transient errors pose additional challenges to reliable operation. Stochastic computing methods can be exploited to address the above issues and thus possibly allow operation with less reliable, leading edge processes in very low voltage and/or high noise operating conditions.

In image processing, Li *et al.* showed that stochastic circuits can outperform conventional binary designs for key algorithms with respect to important design metrics [1]. Specifically, sequential stochastic computational elements were built using finite state machines [2]. Alaghi *et al.* investigated an edge-detection algorithm for real-time image processing [3]. It was shown that the area-delay product of the stochastic edge detection circuit is only 8.7% of that of a conventional binary circuit. Qian and Riedel compared stochastic hardware implementations of polynomial arithmetic [4]. Chang and Parhi investigated novel designs for both FIR and infinite impulse response (IIR) filters based on stochastic logic [5]. Several low-pass and high-pass filters with different cut-off frequencies were considered.

In this paper, we investigate two stochastic FIR filter designs. The conventional weighted average (CWA) design exploits basic stochastic arithmetic elements such as the XNOR gate for multiplication and the multiplexer for addition. The hardwired weighted average (HWA) design leverages the fact that every input signal is selected with the same probability in a multiplexer. The weights are then given by the number of combined inputs of the multiplexer. Different resolutions were considered to determine the threshold (or break-even point) that defines the competitive resolution range for stochastic circuits. 3-bit to 16-bit FIR filters using both stochastic and binary approaches are initially implemented. Then the minimum required sequence length that enables the stochastic circuit to work as accurately as the binary one is determined. The metrics of throughput per area (TPA) and energy per operation (EPO) are used for characterizing and comparing the performance of stochastic and conventional circuits.

This paper makes the following original contributions.

- A new stochastic FIR filter is designed using a novel HWA structure. In the HWA design, the filter coefficients or weights are given by repeating inputs to the multiplexer. It is shown that an HWA-based FIR filter has improved performance in terms of area, power and speed, compared to the CWA design.
- A quantitative analysis of the frequency response of the filters, in terms of the passband ripple and stopband attenuation, is performed to determine the minimum stochastic sequence length required to ensure that the performance of a stochastic filter matches that of a conventional filter.
- A detailed comparison is provided with respect to the fault tolerance of the proposed stochastic and conventional binary filters. A fault-tolerant triple modular redundancy (TMR) implementation of the binary filter is also considered.

II. BACKGROUND

A. Stochastic Computing

Stochastic computing involves processing numbers that are encoded as real values, which are represented using stochastic bit-streams. N_1 1's in a bit stream containing N_s bits represents either the (unsigned) unipolar number N_1/ N_s or the (signed) bipolar number $(2N_1–N_s)/ N_s$ [6, 7]. For example, "0001100101" denotes 2/5 in unipolar and -1/5 in bipolar for $N_s = 10$ and $N_1 = 4$. To encode a binary number containing N_b bits, the minimum sequence length is $N_{s,min} = 2^{N_b}$. However, the required sequence length N_s is usually made larger for increased accuracy during stochastic processing. Therefore, a performance matching multiplier is introduced as $PMM = N_s/2^{N_b}$. The stochastic sequences are usually generated using linear feedback shift registers (LFSRs). Fig. 1 shows a general stochastic computing system [6].

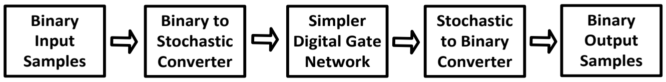

Binary Input Samples
Binary to Stochastic Converter
Simpler Digital Gate Network
Stochastic to Binary Converter
Binary Output Samples

Fig. 1. A basic stochastic computing system.

B. *Encoding Numbers as Unipolar and Bipolar Stochastic Sequences*

Stochastic number generation relies on pseudo-random bit generators such as LFSRs. For example, to generate the stochastic sequence for a 4-bit unsigned binary number, the stochastic number generator (SNG) in Fig. 2 is implemented with a 4-bit LFSR. The SNG in Fig. 2 converts a 4-bit unsigned binary number X to a stochastic number (sequence) of length 16. The SNG takes advantage of the weight generation. The bit streams named $W3$, $W2$, $W1$ and $W0$ represent the weights of 1/2, 1/4, 1/8 and 1/16, respectively. The binary number x is converted bit-by-bit with different weights assigned to them. Therefore, we have $P(S) = 1/2{\cdot}X[3] + 1/4{\cdot}X[2] + 1/8{\cdot}X[1] + 1/16{\cdot}X[0] = (8{\cdot}X[3] + 4{\cdot}X[2] + 2{\cdot}X[1] + 1{\cdot}X[0]) / 16 = X/16$, where S is the output sequence of the SNG and $P(S)$ is the probability that S represents. This S is the unipolar stochastic representation of the binary number X.

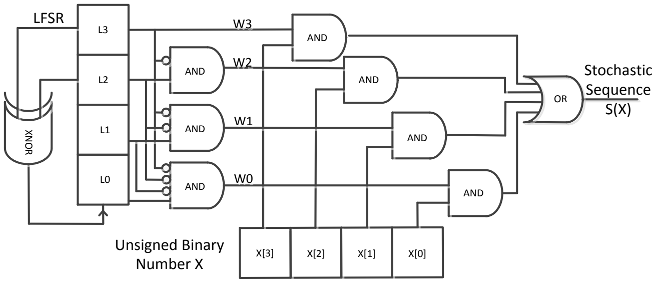


Fig. 2. Unipolar stochastic number generator for unsigned numbers [6].

For signed numbers, we use bipolar stochastic representations. An N_s-bit stochastic sequence with N_1 1's encodes the probability of $(2{\cdot}N_1 - N_s)/N_s$. To design an SNG for signed numbers, let us consider the mappings of a signed binary number to its stochastic representation. For example, for a 4-bit signed binary number in two's complement, Table I shows its relationship with the probability that every single bit in the stochastic sequence is '1' and the probability that the number is encoded in the bipolar representation, assuming that the sequence length is 16 bits. This relationship reveals that the stochastic conversion of a signed binary number can be implemented by the SNG for unsigned numbers, by simply inverting the sign bit and treating the remaining bits in the signed binary number as in an unsigned number. This SNG design is shown in Fig. 3. To invert the signal of the sign bit in the 4-bit signed number, a NOR gate is used to replace the AND gate connected to the sign bit and some inverters are reduced into one at the output of L3.

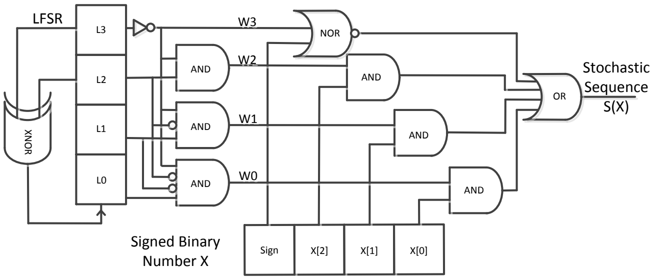


Fig. 3. Bipolar stochastic number generator for signed binary numbers in two's complement.

TABLE I. A MAPPING SCHEME OF SIGNED BINARY NUMBERS IN TWO'S COMPLEMENT AND THEIR STOCHASTIC REPRESENTATIONS.

Signed Binary Number in 2's complement	**Decimal**	**Probability of any bit being '1' in the 16-bit sequence**	**Stochastic Number in the bipolar representation: (2·N1-Ns)/Ns**
0111	7	15/16	(2×15−16)/16=7/8
0110	6	14/16	(2×14−16)/16=6/8
0101	5	13/16	(2×13−16)/16=5/8
0100	4	12/16	(2×12−16)/16=4/8
0011	3	11/16	(2×11−16)/16=3/8
0010	2	10/16	(2×10−16)/16=2/8
0001	1	9/16	(2×9−16)/16=1/8
0000	0	8/16	(2×8−16)/16=0/8
1111	−1	7/16	(2×7−16)/16=−1/8
1110	−2	6/16	(2×6−16)/16=−2/8
1101	−3	5/16	(2×5−16)/16=−3/8
1100	−4	4/16	(2×4−16)/16=−4/8
1011	−5	3/16	(2×3−16)/16=−5/8
1010	−6	2/16	(2×2−16)/16=−6/8
1001	−7	1/16	(2×1−16)/16=−7/8
1000	−8	0/16	(2×0−16)/16=−8/8

C. *FIR Filters*

An N_f –tap FIR filter implements a sum of products over a sliding window of the N_f most recent input samples, as specified in (1). The fixed filter coefficients $H[i]$ (i = 0, 1, …, $N_f - 1$) give the finite impulse response of the filter as

$$Y[n] = \sum_{i=0}^{N_f-1} H[i]\, X[n-i]. \tag{1}$$

The hardware implementation of an FIR filter consists of adders and multipliers as well as delay units implemented as D flip flops as in Fig. 4. To meet the resolution requirement, the stochastic sequence must be as long as $PMM \cdot 2^{N_b}$, where N_b is the binary bit resolution and $PMM \geq 1$. However, it requires huge storage which leads to excessive latency [5].

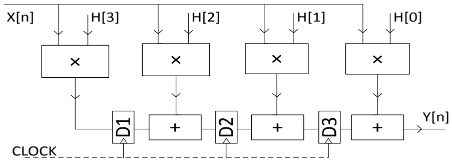


Fig. 4. A 4-tap FIR filter design (i.e., N_f = 4 in (1)).

One solution to the relatively long latency and large storage cost is to move the binary input signal samples through the DFFs before they are expanded into stochastic bit streams (Fig. 5). At every stochastic clock cycle there is one stochastic bit generated by each of the stochastic number generators (SNGs). In Fig. 5, $S(X[n\text{-}i])$ and $S(H[i])$ are the stochastic bit streams encoding the values of $X[n\text{-}i]$ and $H[i]$, respectively, where i = 0, 1, 2, 3. An N_s-bit stochastic sequence $S(Y[n])$ is produced as the filter output over N_s stochastic clock cycles. The design in Fig. 5 requires four expensive SNG modules but only three N_b-bit registers. It has relatively low cost and thus we chose it for further investigation.

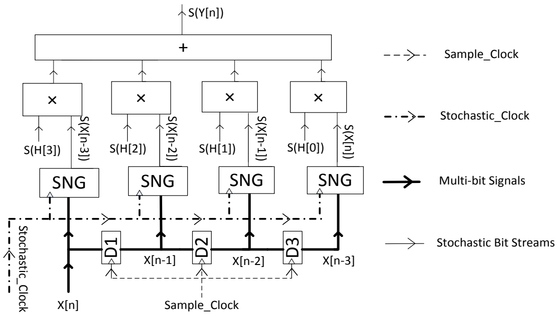


Fig. 5. A 4-tap stochastic FIR filter (i.e., N_f = 4 in (1)) [5].

III. Stochastic FIR Filter Design

A. Conventional Weighted Average (CWA) Design

At the core of an N_f-tap FIR filter is an N_f-input weighted average function. For a 16-tap FIR filter, this function is implemented using stochastic logic, see Fig. 6. The multiplexer is used as a simple adder. The XNOR gates implement bipolar multiplications provided that the two input sequences, i.e., the input sequence and the corresponding coefficient sequence, are statistically independent [6]. Two N_s-bit bipolar stochastic sequences $S1$ and $S2$ are said to be independent if $P(S1\ XNOR\ S2) = P(S1) \cdot P(S2)$, where $P(S1)$ and $P(S2)$ denote the probabilities encoded by $S1$ and $S2$, respectively. Note that the selecting signals (representing the probability of 0.5) are encoded as unipolar sequences by unipolar SNGs (denoted by SNG_u). All the numbers to data inputs are converted by bipolar SNGs (denoted by SNG_b).

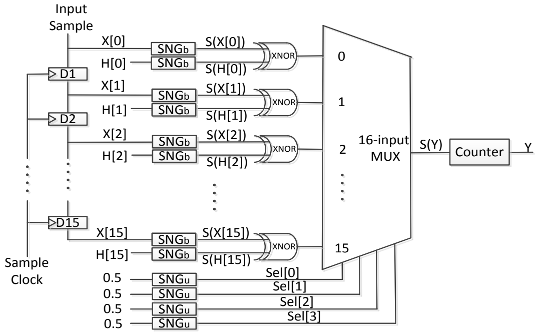


Fig. 6. A 16-tap FIR filter implemented using stochastic logic [5].

B. Hard-wired Weighted Average (HWA) Design

In the CWA design, the SNGs cannot be shared, due to the requirement of signal independency. In the hard-wired weighted average (HWA) design, however, the absolute values of the coefficients can be implemented by assigning unbiased stochastic sequences to the selecting inputs of the multiplexer. In an unbiased stochastic sequence, the probabilities of each bit being '1' and '0' are the same, i.e., 0.5. The probability is then the same for selecting each of the inputs. However, a particular data input can be given more weight in the multiplexer output by connecting the input to multiple multiplexer inputs. Note that the signs of the coefficients can be implemented by XOR gates at the data inputs of the multiplexer. XOR gates help invert the corresponding input when the coefficient is negative. When the coefficients are positive, XOR gates become buffers.

In Fig. 7, for example, Wires 8 to 15 are associated with the same input S(X[4]), where S(X[4]) is the stochastic bit stream encoding the value of X[4]. Thus the probability of selecting the input S(X[4]) is 8/16 or 1/2, which means the coefficient of X[4] is either 1/2 or -1/2. Similarly, all the other coefficients can be weighted by repeating inputs appropriately. The weighted average function in (2) requires a multiplexer with four selecting inputs. It can be implemented by a 16-input multiplexer with combined data inputs as in Fig. 7.

$$Y = sign(A[0]) \cdot \frac{1}{16} \cdot X[0] + sign(A[1]) \cdot \frac{1}{16} \cdot X[1] + sign(A[2]) \cdot \frac{1}{8} \cdot X[2] + sign(A[1]) \cdot \frac{1}{4} \cdot X[3] + sign(A[0]) \cdot \frac{1}{2} \cdot X[4]. \tag{2}$$

The HWA design potentially improves the CWA design in that the SNGs for the weights can be removed.

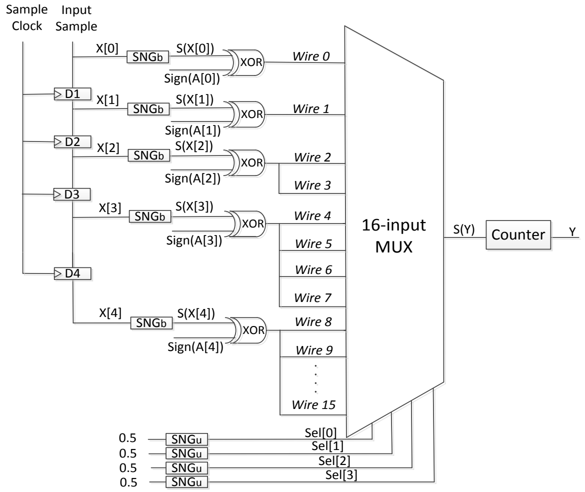


Fig. 7. The hard-wired weighted average design of a 5-tap FIR filter

In general, to implement the N_f-tap FIR filter in (1) using an N_b-bit resolution, the major steps are as follows:

1) Convert the floating point coefficients $H[i]$ in (1) to fixed point N_b-bit binary numbers $A[i]$, where i = 0, 1, …, $N_f - 1$.

2) Calculate the sum of all the absolute values of the coefficients $A = \sum_{i=0}^{N_f-1} |A[i]|$ where $A[i]$ is an N_b-bit binary number, for i = 0, 1, …, $N_f - 1$.

3) The multiplexer has 2^{N_m} data inputs and N_m selecting inputs, where N_m is determined by $N_m = \lceil log_2 A \rceil$ and $\lceil\ \ \rceil$ is the ceiling function. Each selecting input is an unbiased stochastic sequence encoding the probability of 0.5.

4) The number of inputs to be combined is given by $|A[i]|$ for input $X[i]$ (i = 0, 1, ..., $N_f - 1$). The sign of the coefficient A[i] (i = 0, 1, ..., $N_f - 1$) is one of the inputs of the corresponding XOR gate.

5) Use a synthesis tool to optimize the design.

IV. Performance Evaluation of the Conventional Binary and the Proposed Stochastic FIR Filters

A low-pass FIR filter is considered for evaluating the proposed stochastic and binary filter designs with specifications in Table II. The number of taps is $N_f = 267$. The sequence length N_s is given by $N_s = 2^{N_{LFSR}}$, where N_{LFSR} is the number of bits in an LFSR. Various sequence lengths have been investigated for different resolutions to compare with the conventional binary design. N_{LFSR} is varied from 3 up to 30 (i.e., $N_{LFSR} = 3, 4, \ldots, 30$) to determine the sequence length. The resolution N_b ranges from 3 bits to 16 bits. The magnitude responses of the filters are then obtained. Passband ripples (PRs) and stopband attenuations (SAs) are used to evaluate the performance of the binary design for various resolutions and the stochastic design for different sequence lengths. PR and SA are defined as the maximum magnitude response of a frequency in the passband and stopband, respectively.

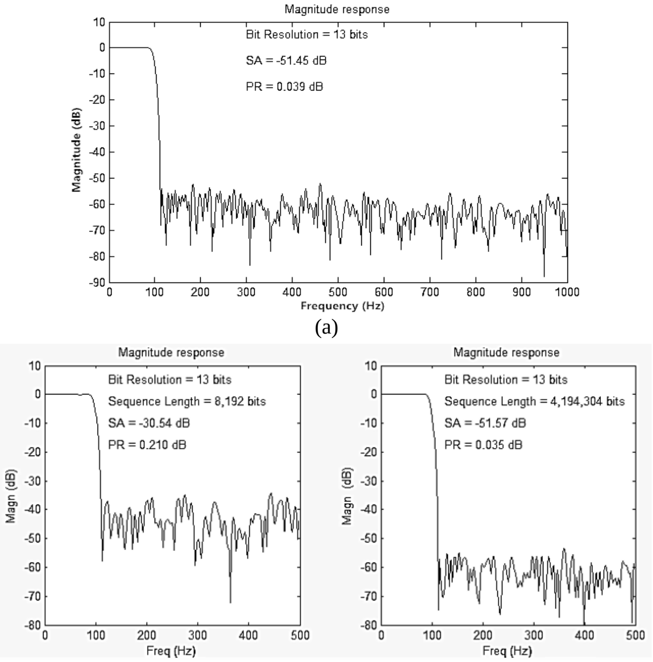
Table II. Low-pass FIR filter specifications.

Specifications	Values
Cut-off frequency f_c	100 Hz
Transition band bandwidth *BW*	30 Hz
Minimum stop-band attenuation	-50 dB
Maximum peak-to-peak pass-band ripple	0.1 dB

In Table III, the PRs and SAs are shown for different bit resolutions and sequence lengths. The HWA-based stochastic FIR filter suffers from both quantization errors and random fluctuations. The stochastic FIR filter has a gradually-improving performance as the sequence length increases. For each bit resolution, the minimum sequence length N_s was found that matched the performance of the stochastic FIR filter to that of the conventional N_b-bit binary FIR filter. The performance matching multiplier (PMM) is then calculated by $PMM = N_s/2^{N_b}$.

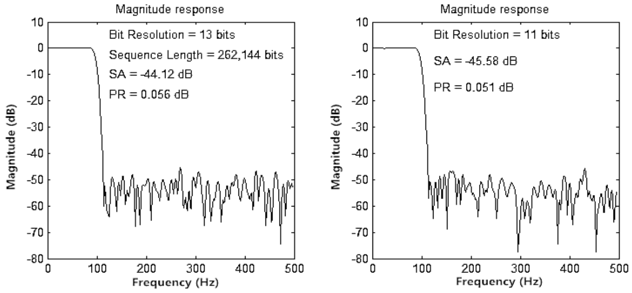
The minimum resolution to achieve the filter specifications in Table II is 13 bits for the binary design. The attenuation in the stopband is -51.45 dB and the passband ripple is 0.039 dB. The magnitude responses of the stochastic and binary filters are plotted in Fig. 8. The HWA-based design with the minimum sequence length suffers from a maximum PR of 0.210 dB, as shown in Fig. 8(b). The SA for the HWA-based FIR filter is only -31.54 dB (Fig. 8(b)). To match the performance of the binary filter, the oversampling rate has to be increased to 512 (Fig. 8(c)). The PR and SA are 0.035 dB and -51.57 dB, respectively, for the stochastic filter in this case.

A shorter sequence length can be used for the stochastic filter to obtain a possibly still useful degraded performance. For example, if the PMM is 32 instead of 512, the SA becomes -44.12 dB and the PR is 0.06 dB (see Fig. 9(a)). The time per operation is reduced to only 1/16 of the previous result, which indicates a graceful degradation in the performance of the stochastic design. In contrast, an 11-bit conventional binary implementation of the filter shows similarly degraded performances with an SA of -45.58 dB and a PR of 0.05 dB (see Fig. 9(b)), however with little saving in energy consumption. This trade-off will be further discussed in the next section.



(b) (c)

Fig. 8. Magnitude responses of 13-bit FIR filters: (a) Conventional binary design, (b) Stochastic HWA design without performance matching (PMM = 1), (c) Stochastic HWA design with performance matching (PMM = 512).



(a) (b)

Fig. 9. Magnitude responses of lower-quality FIR filters: (a) 13-bit stochastic HWA design with PMM = 32 and (b) 11-bit conventional binary design.

V. Simulation Results

The Synopsys design compiler with an STM 28-nm cell library [8] was used to synthesize a high-level design in VHDL to a standard cell ASIC design. The FIR filters specified in Table II were simulated for various resolutions from 3 bits up to 16 bits. In Table III, the circuit performance is compared with respect to silicon area, power consumption and delay. Although the core of the stochastic circuit is implemented using simple logic gates, the interfacing circuits require a relatively large number of SNGs and counters, especially for FIR filters with a large number of taps. The hardware cost of the binary circuits grows faster than the stochastic circuits, so for larger resolutions, the stochastic HWA circuit becomes increasingly advantageous over a binary design. The auxiliary circuits such as the SNGs and counters, however, make this advantage of stochastic circuits less significant.

TABLE III: PERFORMANCE OF THE CONVENTIONAL BINARY (B) AND THE PROPOSED STOCHASTIC (S) FIR FILTER. $\boldsymbol{N_b}$: BIT RESOLUTION, $\boldsymbol{N_s}$: SEQUENCE LENGTH, PR: PASSBAND RIPPLE, SA: STOPBAND ATTENUATION AND PMM: PERFORMANCE MATCHING MULTIPLIER.

N_b	N_s	PR (dB)		SA (dB)		PMM
		CB	HWA	CB	HWA	
3	64	3.078	2.582	-3.91	-7.14	8
4	256	4.788	1.414	-9.93	-11.49	16
5	512	2.860	1.324	-13.11	-14.70	16
6	1,024	1.462	0.801	-17.80	-17.90	16
7	2,048	0.722	0.552	-24.02	-20.65	16
8	8,192	0.391	0.209	-27.63	-27.53	32
9	16,384	0.18	0.282	-31.13	-31.95	32
10	65,536	0.103	0.071	-36.37	-36.02	64
11	524,288	0.052	0.050	-45.58	-44.60	256
12	1,048,576	0.032	0.037	-49.94	-48.32	256
13	4,194,304	0.039	0.034	-51.45	-51.57	512
14	16,777,216	0.035	0.036	-54.59	-55.51	1,024
15	134,217,728	0.036	0.035	-55.91	-54.89	4,096
16	536,870,912	0.036	0.036	-54.96	-55.32	8,192

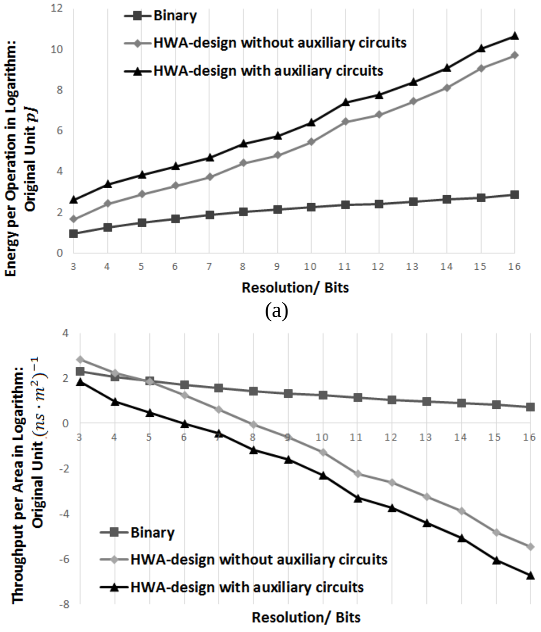
Note that both the binary and stochastic circuits have been optimized for maximum throughput by adding pipeline registers as determined by the Synopsys synthesis tool. The area could be over-estimated for this reason. Long latency has been a major challenge for stochastic circuits. Adopting a faster clock is a potential way to reduce latency. With the help of timing analysis, the clock can be pushed to the limit according to the slack time. The results are shown in Table IV. The required stochastic sequence length is given by $2^{N_b} \cdot PMM$, where N_b ($N_b = 3, 4, \ldots, 16$) is the binary resolution and PMM is the performance matching multiplier. The reported power consumptions are estimated at the fastest clocks for each of the resolutions. It can be seen that the stochastic circuits are more compact and they consume less energy per clock cycle than binary implementations. In the comparison of the two stochastic designs, the HWA-based FIR filter is more cost-efficient in terms of area, power and speed (see Table IV).

TABLE IV. COMPARISON OF HARDWARE COST, POWER CONSUMPTION AND MINIMUM CLOCK PERIOD FOR CONVENTIONAL BINARY (B) FILTERS, CWA-BASED STOCHASTIC (C) FIR FILTERS AND HWA-BASED STOCHASTIC (H) FIR FILTERS INCLUDING AUXILIARY CIRCUITS.

R	**Area (× 10^3 μm^2)**			**Power (mW)**			**Min Clock Period (× 10 ps)**		
	B	**C**	**H**	**B**	**C**	**H**	**B**	**C**	**H**
3	12.76	14.94	12.85	24.24	20.35	19.82	39	37	34
4	21.26	15.41	15.55	35.50	29.79	29.05	41	37	35
5	31.89	21.02	18.75	47.86	40.17	38.02	42	40	36
6	44.65	26.84	21.99	61.22	51.39	47.96	44	40	36
7	59.53	27.98	24.71	75.49	63.36	59.06	47	41	39
8	76.54	27.69	27.41	90.59	76.04	71.55	49	41	40
9	95.68	33.54	29.98	106.5	89.38	84.66	49	42	41
10	116.9	36.24	33.12	123.1	103.3	97.23	50	43	41
11	140.3	38.34	36.51	140.5	117.9	109.0	51	47	42
12	165.8	44.99	43.53	138.5	133.1	125.2	54	47	44
13	193.5	46.57	44.91	158.5	148.7	140.7	55	47	44
14	223.3	50.16	48.06	177.1	164.8	159.2	57	49	45
15	255.1	53.74	50.82	196.3	181.5	174.8	59	49	46
16	289.2	57.32	54.89	226.6	198.6	189.6	64	50	46

The stochastic circuits, however, suffer from the long latency caused by the required stochastic sequences, which makes the designs less competitive. Hence, a seemingly low-power design may take more clock cycles to run an arithmetic operation, which might in fact need more energy rather than saving it. The operational efficiency of such a design is captured by the throughput per area (TPA), defined as the number of operations per circuit area in a unit time, and the energy per operation (EPO), obtained as the product of the power and time required to complete one operation. In an evaluation, the stochastic FIR filter must work as effectively as the conventional binary FIR filter. The effectiveness is measured using passband ripple (PR) and stopband attenuation (SA) in Table III. The sequence length is a key factor that determines the efficacy of a stochastic design. When computing the TPA and the EPO, therefore, the required sequence lengths in Table III must be used, making the stochastic design even less competitive at higher resolutions.

Fig. 10 shows the plots of EPO and TPA, respectively, for the binary and stochastic circuits (with and without the auxiliary circuits). The *y*-axes in both plots are the base-10 logarithms of the original metrics. The *x*-axes are bit resolutions from 3 bits to 16 bits. The figures show that the stochastic approach is not competitive in terms of EPO and TPA. The higher the resolution, the less competitive the stochastic implementation becomes. This is caused by the required sequence length, which grows exponentially with the bit resolution. When the auxiliary circuits are not considered, the stochastic circuit performs better in terms of the TPA than the binary design for resolutions under 5 bits.



(a)

(b)

Fig. 10. (a) EPO and (b) TPA comparisons: HWA Design (with/without auxiliary circuits such as SNGs and counters) and Binary Design

Although the stochastic design suffers from a long latency, its performance degrades gracefully as the energy is reduced, which could be beneficial in a failing battery scenario. Take the 13-bit design as an example. A lower-quality stochastic filter (Fig. 9) is implemented using 262,144 bits (in contrast to 4,194,304 bits required by a good-quality stochastic filter that matches the performance of a 13-bit conventional binary filter). An 11-bit conventional binary filter shows similarly degraded performance. The EPOs of these lower-quality filters are shown in Table V. As the stochastic filter only requires 1/16 of the original sequence length, the EPO saving is 93.33% below that of the good-quality stochastic filter. However, the 11-bit binary filter saves only 17.81% of the energy per operation compared to the 13-bit binary filter.

TABLE V: ENERGY SAVINGS BY LOWER-QUALITY IMPLEMENTATIONS OF CONVENTIONAL BINARY (B) AND HWA-BASED FIR FILTERS.

Implementations	B	HWA
EPO of High Quality Filter (*pJ*)	87.15	253700027
EPO of Lower Quality Filter (*pJ*)	71.63	15856251
Energy Saving (%)	17.81	93.75

VI. FAULT TOLERANCE ANALYSIS

Stochastic computing has been known to be intrinsically fault-tolerant. When one bit in a binary circuit flips, it can cause a serious error if the erroneous bit is among the MSBs. However, any bit in a stochastic sequence has the same weight, so the effect of a single bit flip is insignificant in a relatively long stochastic sequence. To measure the reliability of a design, the average absolute error (AAE) is defined as $AAE = \frac{1}{M} \cdot \frac{1}{2^{2N_b+4}} \sum_{i=0}^{M-1} |X_i - X_i'|$, where X_i and X_i' are the expected correct output and the actual output, respectively, and M is the number of simulations. For simplicity, $\frac{1}{2^{2N_b+4}}$ is used as a constant coefficient so that the AAEs lie between 0 and 1 ($N_b = 13$ here). The AAE indicates how seriously the injected error affects the circuit output.

We investigate the AAE for the conventional binary 13-bit low-pass FIR filter with 267 taps as well as the stochastic HWA design using a sequence length of 4,194,304 bits (see Table III) under various injected error rates. In addition, redundant copies of the binary circuit can be used to obtain fault tolerance in the form of triple modular redundancy (TMR) with unreliable and fault-free voters. The stochastic computational models in [9] are used to facilitate the fault-tolerance analysis. XOR gates are used to inject errors into the circuit. The majority voters in the TMR circuits are bitwise not word-wise.

TABLE VI. AVERAGE ABSOLUTE ERROR OF THE STOCHASTIC (S) AND BINARY (B) CIRCUITS WITH AND WITHOUT REDUNDANCY AT VARIOUS INJECTED ERROR RATES.

Error Rate (%)	Average Absolute Error (%)			
	S	B	Binary TMR	
			Error-free Voter	Unreliable Voter
0	0	0	0	0
0.1	0.065	1.507	0.004	0.126
0.2	0.136	2.325	0.009	0.225
0.5	0.326	3.290	0.035	0.581
1	0.574	5.209	0.111	1.203
2	1.226	6.368	0.198	2.382
5	2.948	10.942	0.337	5.794
10	5.696	21.477	1.123	12.050

The AAEs obtained from 200 simulations with a sequence length of 100,000 bits are given in Table VI. It shows that the AAE increases as the injected error rate increases. The conventional binary circuit is not as fault-tolerant as the stochastic circuit. The binary TMR circuit with unreliable voters has increased reliability, but it is still not as reliable as the stochastic approach. However, the binary TMR circuit with reliable voters becomes more fault-tolerant than the stochastic circuit.

VII. CONCLUSIONS

In this paper, a stochastic hard-wired weighted average (HWA) design is proposed to implement FIR filters. The HWA design takes advantage of simply repeating the input wires of a multiplexer to implement the weights of different data inputs. The stochastic design has a smaller circuit area and lower power consumption, compared with the conventional stochastic design using simple arithmetic elements.

Compared to binary FIR filter circuits, the proposed stochastic design has a significant advantage in circuit area, especially for higher resolutions. In terms of throughput per area and energy per operation, however, the stochastic design does not show any advantages over its binary counterpart. This is primarily due to the significant latency in stochastic computing, because long sequences must be used to match the performance of a binary filter. A shorter stochastic sequence, however, would make the stochastic circuit degrade gracefully in performance compared to the binary design. Graceful degradation to reduce power in stochastic circuits is easily achieved by changing the stochastic bit counter limit, but is difficult to achieve in bit-parallel binary circuits.

A binary TMR circuit using error-free voters is more reliable than the stochastic design. Due to the intrinsic fault tolerance capacity, however, the proposed stochastic design shows significant advantages in reliability over the conventional binary design and its TMR implementation when the voters are subject to errors. These results suggest future work on sum-of-product based fault-tolerant circuit design using stochastic computing techniques. Massively parallel designs, where the overhead circuits can be shared across arrays of stochastic data paths, should also be considered for stochastic implementation.

REFERENCES

[1] P. Li, D. J. Lilja, "Using stochastic computing to implement digital image processing algorithms." *IEEE ICCD*, pp. 154-161, 2011.

[2] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. on VLSI Systems*, vol. 22, no. 3, pp. 449–462, March 2014.

[3] A. Alaghi, C. Li, and J. P. Hayes, "Stochastic circuits for real-time image-processing applications." *DAC*, pp. 1-6, 2013.

[4] W. Qian, M. D. Riedel, "The synthesis of robust polynomial arithmetic with stochastic logic." *DAC*, pp. 648-653. 2008.

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

[6] A. Alaghi, J. P. Hayes, "Survey of stochastic computing." *ACM Trans. on Embedded Computing Systems (TECS)* 12, no. 2s (2013): 92.

[7] B. R. Gaines, "Stochastic computing systems." In Advances in Information Systems Science, pp. 72-73, Springer US, 1969.

[8] D. W. Knapp, *Behavioral synthesis: digital system design using the Synopsys behavioral compiler*, Prentice-Hall, Inc., 1996.

[9] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, F. Lombardi, "A stochastic computational approach for accurate and efficient reliability evaluation." *IEEE Trans. on Computers*, vol. 63, no. 6, pp. 1336 – 1350, June 2014.

[10] R. Wang, J. Han, B. Cockburn and D. Elliott, "Stochastic Circuit Design and Performance Evaluation of Vector Quantization," in *Proc. IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (IEEE ASAP 2015)*, Toronto, Canada, July 27 - 29, 2015.