

# High-Accuracy and Area-Efficient Stochastic FIR Digital Filters Based on Hybrid Computation\*

Shunsuke KOSHITA<sup>†a)</sup>, Naoya ONIZAWA<sup>††,†††b)</sup>, *Members*, Masahide ABE<sup>†</sup>, *Senior Member*, Takahiro HANYU<sup>†††</sup>, *Member*, and Masayuki KAWAMATA<sup>†</sup>, *Fellow*

**SUMMARY** This paper presents FIR digital filters based on stochastic/binary hybrid computation with reduced hardware complexity and high computational accuracy. Recently, some attempts have been made to apply stochastic computation to realization of digital filters. Such realization methods lead to significant reduction of hardware complexity over the conventional filter realizations based on binary computation. However, the stochastic digital filters suffer from lower computational accuracy than the digital filters based on binary computation because of the random error fluctuations that are generated in stochastic bit streams, stochastic multipliers, and stochastic adders. This becomes a serious problem in the case of FIR filter realizations compared with the IIR counterparts because FIR filters usually require larger number of multiplications and additions than IIR filters. To improve the computational accuracy, this paper presents a stochastic/binary hybrid realization, where multipliers are realized using stochastic computation but adders are realized using binary computation. In addition, a coefficient-scaling technique is proposed to further improve the computational accuracy of stochastic FIR filters. Furthermore, the transposed structure is applied to the FIR filter realization, leading to reduction of hardware complexity. Evaluation results demonstrate that our method achieves at most 40dB improvement in minimum stopband attenuation compared with the conventional pure stochastic design.

**key words:** FIR digital filter, stochastic computation, computational accuracy, digital circuit implementation

## 1. Introduction

Digital filters [1]–[3] are well known as an important fundamental technique in signal processing. Applications of digital filters can be found in a number of fields such as audio/speech processing, image/video processing, communications, control systems, and biomedical signal processing. Digital filters are classified into FIR (Finite Impulse Response) filters and IIR (Infinite Impulse Response) filters. In this paper we focus on FIR filters. It is well known that FIR filters are superior to IIR filters in that the stability is always guaranteed.

A digital filter consists of three building blocks: adders,

multipliers, and unit delay elements. Since the number of these building blocks depends on the filter order, higher-order digital filters require larger number of the building blocks and unfortunately increase the hardware complexity. In particular, hardware implementation of a multiplier is very costly and high-order digital filters with many multipliers lead to significant increase of the hardware complexity, causing large power dissipation and large area. Although there exist many techniques to improve digital filter architectures with reduced hardware complexity, such techniques have limitations in meeting current stringent requirements such as ultra-low power dissipation and extremely small size. Therefore, in order to solve this challenging task, there is a need to reconsider not only architectures but also computing methods in digital filters.

It is well known that digital filters are conventionally realized based on binary computation such as two's complement arithmetic. On the other hand, *stochastic computation* [4], [5] has recently received considerable attention [6]–[12] as another promising candidate for digital filter realization. Stochastic computation represents data as streams of random bits. Using stochastic computation, a multiplier can be implemented using a simple logic gate and an adder can be also simply implemented using a multiplexer, leading to area-efficient hardware. Hence the implementation cost of digital filters based on stochastic computation can be significantly reduced in comparison with digital filters based on the conventional binary computation. However, digital filters based on stochastic computation suffer from lower computational accuracy than binary-computation-based digital filters. One of the reasons of this problem is the error caused by the randomness of bit streams in stochastic computation. Another reason lies in the fact that an increase in the precision of stochastic computation requires much larger number of bit streams than binary computation. This problem becomes very serious in realization of FIR filters. Since FIR filters usually require higher order than IIR filters to meet a desired specification of frequency response, such high-order filters involve large number of adders and multipliers. This means that the number of error sources caused by stochastic adders/multipliers also becomes large in the case of FIR filters, causing severe degradation of frequency response.

In this paper we present stochastic FIR digital filters that simultaneously attain high computational accuracy and low hardware complexity. In order to improve the computational accuracy, we make use of stochastic/binary hy-

Manuscript received October 13, 2016.

Manuscript revised February 28, 2017.

Manuscript publicized May 22, 2017.

<sup>†</sup>The authors are with Graduate School of Engineering, Tohoku University, Sendai-shi, 980–8579 Japan.

<sup>††</sup>The author is with Frontier Research Institute for Interdisciplinary Sciences, Tohoku University, Sendai-shi, 980–8578 Japan.

<sup>†††</sup>The authors are with Research Institute of Electrical Communication, Tohoku University, Sendai-shi, 980–8577 Japan.

\*A part of this paper was presented at 2016 IEEE 46th International Symposium on Multiple-Valued Logic.

a) E-mail: kosita@mk.ecei.tohoku.ac.jp

b) E-mail: nonizawa@m.tohoku.ac.jp

DOI: 10.1587/transinf.2016LOP0011

brid computation, where multipliers are realized based on stochastic computation but adders are realized based on binary computation to avoid the error given in stochastic adders. In addition, we present a coefficient-scaling technique that further improves the computational accuracy in the FIR filtering based on stochastic computation. Furthermore, we realize stochastic FIR filters using the transposed structure, leading to significant reduction of hardware complexity in our stochastic/binary hybrid method.

The short version of this work was presented in [11], where limited results on the stochastic/binary hybrid computation were presented at algorithm level. On the other hand, the extended version presented in this paper includes: (a) transposed structure to reduce the hardware complexity of stochastic/binary hybrid computation, (b) coefficient-scaling technique to further improve the computational accuracy, and (c) hardware evaluation of the proposed stochastic FIR filters using Verilog-HDL and TSMC 65nm CMOS technology.

The rest of this paper is organized as follows. Section 2 reviews stochastic computation. Section 3 briefly reviews FIR digital filters and addresses the problems in the conventional realization method of stochastic FIR filters. Section 4 gives the proposed method. Section 5 evaluates the computational accuracy and the hardware complexity of the proposed stochastic FIR filters in comparison with the conventional stochastic FIR filters. Section 6 concludes this paper.

## 2. Stochastic Computation

Stochastic computation was first introduced in [4] and it represented information by sequences of random bits. It is a purely-digital implementation that provides complicated calculations such as multiplication and exponential function with simple area-efficient hardware [13]. Due to this hardware simplicity, stochastic computation has recently attracted considerable interest in many practical applications that include not only digital filters, but also neural network, communication, image processing, and so on [5], [14]–[18].

In stochastic computation, data bit-streams are represented by one of the two formats: *unipolar coding* and *bipolar coding*. This section first introduces these two formats and then explains data multiplication and data addition based on stochastic computation.

### 2.1 Unipolar Coding and Bipolar Coding

Given a data  $s$ , let  $S$  be a bit-stream that represents  $s$  by stochastic computation with  $N_{\text{sto}}$  bits, i.e.

$$S = S_0 S_1 \cdots S_{N_{\text{sto}}-1} \\ S_i \in \{0, 1\} \quad (0 \leq i \leq N_{\text{sto}} - 1). \quad (1)$$

Now, let  $P_s$  be the probability of observing ‘1’ in  $S_i$ . In *unipolar coding* we consider  $s = P_s$  under the assumption that the value of the given data  $s$  satisfies  $0 \leq s \leq 1$ , and we generate each bit  $S_i$  based on the probability  $P_s$  to obtain the

**Table 1** Bit-stream  $S$  represented by unipolar coding. ( $N_{\text{sto}} = 3$ )

$s$	0	1/3	2/3	1
$S$	(000)	(001) (010) (100)	(011) (101) (110)	(111)

**Table 2** Bit-stream  $S$  represented by unipolar coding and bipolar coding. ( $N_{\text{sto}} = 3$ )

$s$	0	1/3	2/3	1
$s'$	-1	-1/3	1/3	1
$S$	(000)	(001) (010) (100)	(011) (101) (110)	(111)

entire bit-stream  $S$ . The value of the resultant bit-stream  $S$  corresponds to the following data  $\hat{s}$ :

$$\hat{s} = \frac{1}{N_{\text{sto}}} \sum_{i=0}^{N_{\text{sto}}-1} S_i. \quad (2)$$

This equation shows that, in a bit-stream represented by the unipolar coding, the weight of each bit  $S_i$  is independent of  $i$  and hence becomes  $1/N_{\text{sto}}$  for any  $i$ . Table 1 shows an example of construction of the bit-streams  $S$  in the case of  $N_{\text{sto}} = 3$ . Since the weight of a bit ‘1’ becomes  $1/3$  in this case,  $S$  is non-unique for  $s = 1/3$  and  $s = 2/3$  and has three patterns of bit-streams.

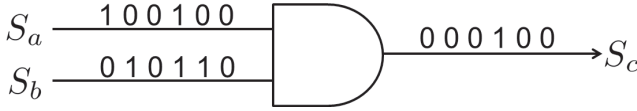
Next, *bipolar coding* is explained. Since the unipolar coding must impose the assumption of  $0 \leq s \leq 1$ , negative-valued data cannot be represented by the unipolar coding. On the other hand, the bipolar coding can deal with negative-valued data as well as the positive-valued data. The bipolar coding represents a data  $s'$  by using the following relationship

$$s' = 2P_s - 1 \quad (3)$$

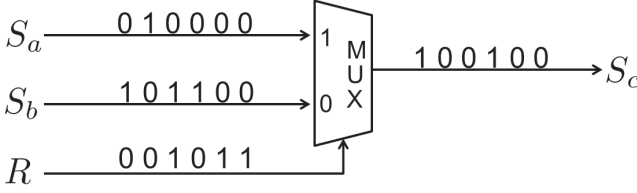
where  $P_s$  is the probability that is used in the unipolar coding. The range of  $s'$  becomes  $-1 \leq s' \leq 1$  because  $P_s$  must satisfy  $0 \leq P_s \leq 1$ . Table 2 shows the bit-streams represented by the bipolar coding and the unipolar coding in the case of  $N_{\text{sto}} = 3$ . The bipolar coding corresponds to the data  $s'$  and the unipolar coding corresponds to  $s$ . In this 3-bit case, the bipolar coding can represent four patterns of data  $\{-1, -1/3, 1/3, 1\}$  that correspond to the patterns  $\{0, 1/3, 2/3, 1\}$  for the unipolar coding.

### 2.2 Multiplication Based on Stochastic Computation

We first explain data multiplication for the unipolar coding. Let  $S_a = S_{a0} S_{a1} \cdots S_{aN_{\text{sto}}-1}$  and  $S_b = S_{b0} S_{b1} \cdots S_{bN_{\text{sto}}-1}$  be bit-streams represented by the unipolar coding with  $N_{\text{sto}}$  bits, and let  $P_{s_a}$  and  $P_{s_b}$  be the probabilities of observing ‘1’ in  $S_{ai}$  and  $S_{bj}$  for  $0 \leq i \leq N_{\text{sto}} - 1$  and  $0 \leq j \leq N_{\text{sto}} - 1$ , respectively. Also, let  $S_c = S_{c0} S_{c1} \cdots S_{cN_{\text{sto}}-1}$  be a bit-stream representing the product of  $S_a$  and  $S_b$ , and let  $P_{s_c}$  be the probability of observing ‘1’ in  $S_{ck}$  for  $0 \leq k \leq N_{\text{sto}} - 1$ . If  $S_a$  and  $S_b$  are statistically independent to each other,  $P_{s_c}$



**Fig. 1** Realization of multiplier based on unipolar coding. ( $N_{sto} = 6$ ,  $s_a = 2/6$ ,  $s_b = 3/6$ )



**Fig. 2** Realization of adder based on unipolar coding. ( $N_{sto} = 6$ ,  $s_a = 1/6$ ,  $s_b = 3/6$ )

becomes the product of  $P_{s_a}$  and  $P_{s_b}$ , i.e.

$$P_{s_c} = P_{s_a} P_{s_b}. \quad (4)$$

Based on this relationship, a multiplier based on the unipolar coding is simply realized by a two-input AND gate. An example of the multiplication for  $N_{sto} = 6$ ,  $s_a = 2/6$  and  $s_b = 3/6$  is shown in Fig. 1. It is clear from this figure that the output signal of the two-input AND gate becomes the product  $P_{s_a} P_{s_b}$ .

In the case of the bipolar coding, a multiplier can be also easily realized by a simple logic gate: when two bit-streams are represented by the bipolar coding, their product is obtained as the output of a two-input XNOR gate.

### 2.3 Addition Based on Stochastic Computation

In both the unipolar and the bipolar codings, an adder is realized by a multiplexer. To explain this, let  $S_a$ ,  $S_b$ ,  $P_{s_a}$  and  $P_{s_b}$  be as above, and let  $S_c$  be a bit-stream representing the sum of  $S_a$  and  $S_b$ . Also, let  $P_{s_c}$  be the probability of observing '1' in  $S_{ck}$  for  $0 \leq k \leq N_{sto} - 1$ . Then,  $P_{s_c}$  is obtained by

$$P_{s_c} = P_r P_{s_a} + (1 - P_r) P_{s_b} \quad (5)$$

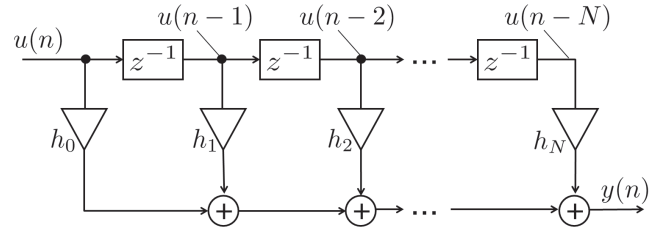
where  $P_r$  is the probability of observing '1's in the control input  $R$  of the multiplexer. In order to realize data addition,  $P_r$  is set to be  $1/2$ . Hence  $P_{s_c}$  becomes

$$P_{s_c} = \frac{1}{2}(P_{s_a} + P_{s_b}) \quad (6)$$

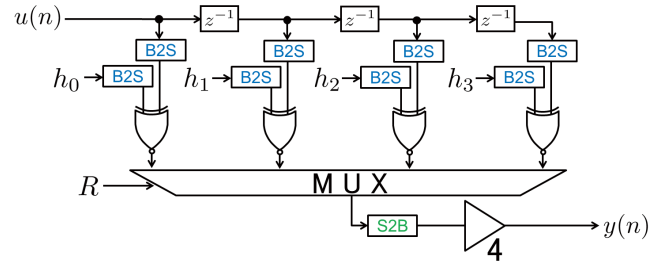
which means that the output of the multiplexer becomes the sum of  $P_{s_a}$  and  $P_{s_b}$ , but this sum is scaled down by two. An example of this scaled addition based on the unipolar coding for  $s_a = 1/6$  and  $s_b = 3/6$  with  $N_{sto} = 6$  is shown in Fig. 2. From this figure we easily see that  $s_c = (s_a + s_b)/2 = 2/6$  is obtained at the output of the multiplexer.

### 3. Stochastic FIR Digital Filters Based on Conventional Method

Given a discrete-time signal  $u(n)$  with  $n$  being an integer



**Fig. 3** Block diagram of  $N$ -th order FIR filter.



**Fig. 4** Conventional realization of 3rd-order FIR filter using stochastic computation.

that represents the sample time, FIR digital filters perform the following convolution to produce the output signal  $y(n)$ :

$$y(n) = \sum_{k=0}^N h_k u(n-k) \quad (7)$$

where  $N$  is the filter order and the set  $\{h_k\}$  for  $0 \leq k \leq N$  denotes the filter coefficients. The block diagram of this FIR filter is shown in Fig. 3, where the building block indicated by  $z^{-1}$  stands for the unit delay element.

Next, the conventional method for FIR filter realization based on stochastic computation is addressed. Throughout this paper, the bipolar coding is used for realization of stochastic FIR filters. Here, we consider a 3rd-order FIR filter as an example. Using the conventional stochastic computation, this filter is realized as in Fig. 4. In this figure, the filter input,  $u(n)$ , the filter output,  $y(n)$ , and the filter coefficients,  $h_0$ ,  $h_1$ ,  $h_2$  and  $h_3$  are represented by signed binary numbers. The system 'B2S' stands for the binary-to-stochastic converter that converts an input signal represented by signed binary numbers into the signal represented by stochastic numbers based on the bipolar coding. The system 'S2B' is the stochastic-to-binary converter that converts an input signal in the stochastic format into the signal in the binary format. Since the bipolar coding is used here, the multipliers in this filter are realized using XNOR gates, and the adder is realized using multiplexer. Note that the four-input multiplexer is used here because a 3rd-order FIR filter requires the sum of four terms:  $h_0 u(n)$ ,  $h_1 u(n-1)$ ,  $h_2 u(n-2)$  and  $h_3 u(n-3)$ . Also, note that the output of 'S2B' is multiplied by 4 to produce  $y(n)$  because the four-input multiplexer is the scaled adder of which output signal represents  $(h_0 u(n) + h_1 u(n-1) + h_2 u(n-2) + h_3 u(n-3))/4$  in the bipolar coding. This multiplication can be easily realized by means of logical shift left by two bits.

Although  $N$ -th order stochastic FIR filters can be also realized in this way, this conventional method unfortunately results in severe degradation of frequency response compared with binary-computation-based filters. Such degradation is due to the significant loss of computational accuracy caused by the following two problems.

1. As is apparent from (7), FIR filters require many data additions. Since data additions in stochastic computation involve scaling, many scalings must be carried out to calculate the filter output  $y(n)$ . This means that calculation of  $y(n)$  must deal with very small data values. Needless to say, stochastic computation requires very large number of bit streams to accurately represent such small values. Under the limited bit length, therefore, the error generated in calculation of  $y(n)$  becomes very large.
2. Usually, most of the values of filter coefficients  $h_k$  for  $0 \leq k \leq N$  become close to zero. Hence the error generated at each output of 'B2S' for  $h_k$  becomes very large.

In the next section, we shall present a realization method to overcome these problems and to improve the computational accuracy.

#### 4. Proposed Method

The proposed method consists of three strategies: stochastic/binary hybrid computation, transposed structure, and coefficient-scaling technique. The details of these strategies are described in this section.

##### 4.1 Stochastic/Binary Hybrid Computation

The stochastic/binary hybrid computation, which was presented in our previous work [11], replaces adders based on stochastic computation with the ones based on binary computation. In other words, we perform data multiplications based on stochastic computation, but we perform data additions based on binary computation instead of using the stochastic scaled adders. The 3rd-order stochastic FIR filter realized by stochastic/binary hybrid computation is shown in Fig. 5. In order to perform data additions based on binary computation, we apply 'S2B' to all of the outputs of the XNOR gates and we obtain the four terms  $h_0u(n)$ ,  $h_1u(n-1)$ ,  $h_2u(n-2)$  and  $h_3u(n-3)$  in the binary format. The sum of these four terms directly becomes the filter output  $y(n)$  because the scaling is not required at each data addition due to the use of binary computation. In this way, we avoid the scaling at data additions and improve the accuracy of filter operation. Hence the first problem addressed in Sect. 3 is solved by this strategy.

**Remark 1:** Although adders based on binary computation can be realized without scaling, the outputs of adders without scaling may sometimes suffer from overflow. However, the overflow is not a serious problem in the proposed

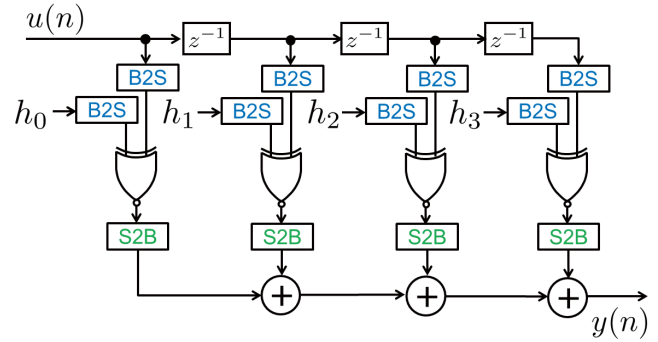


Fig. 5 3rd-order FIR filter using stochastic/binary hybrid computation.

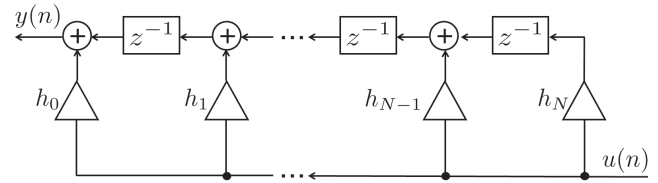


Fig. 6 Block diagram of  $N$ -th order FIR filter with transposed structure.

method. As is well known, overflow at adder outputs does not have any effect on the filter output  $y(n)$  and hence no degradation occurs if all of the following conditions are satisfied:

- An FIR filter to be realized is linear and its peak gain does not exceed unity.
- Two's complement representation is used in binary computation.

Fortunately these conditions are satisfied for most FIR filters, and hence we conclude that our method can be used without the problem of the overflow.

##### 4.2 Transposed Structure

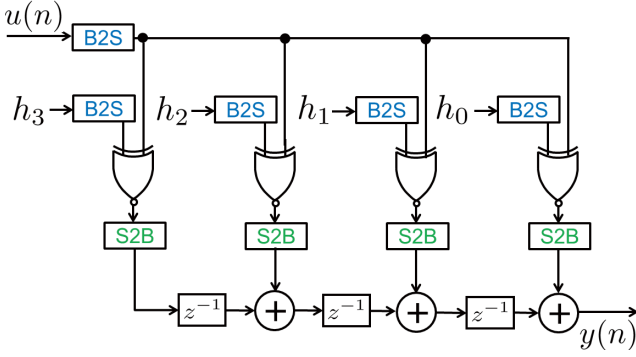
As is apparent from Figs. 4 and 5, the FIR filters given by stochastic/binary hybrid computation require  $N$  additional 'S2B's, causing higher hardware complexity than the conventional stochastic FIR filters. In order to overcome this drawback, we make use of the transposed structure [1]–[3].

The FIR filter with the transposed structure is shown in Fig. 6. This structure is derived from the original structure by applying the following procedures to Fig. 3:

- Exchange the input signal  $u(n)$  and the output signal  $y(n)$ .
- Invert the direction of each signal flow.
- Replace each summation node with a branch node. Also, replace each branch node with a summation node.

The transposed structure is different from Fig. 3 in that each delay element is placed at the output of the corresponding product-sum operation. However it is easy to see that the transfer function of this structure is the same as that of





**Fig. 7** 3rd-order FIR filter using transposed structure and stochastic/binary hybrid computation.

**Table 3** Comparisons of 'B2S's and 'S2B's in  $N$ -th order stochastic FIR filters.

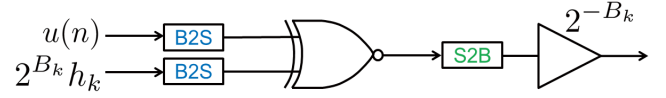
	Conventional	Previous [11]	Transposed
Number of 'B2S's	$2(N + 1)$	$2(N + 1)$	$N + 2$
Number of 'S2B's	1	$N + 1$	$N + 1$

Fig. 3.

Now, the 3rd-order FIR filter using the transposed structure and stochastic/binary hybrid computation is given in Fig. 7. This figure shows that the resultant filter reduces three 'B2S's compared with the previous method of Fig. 5. This means that the use of the transposed structure can reduce  $N$  'B2S's in realization of  $N$ -th order FIR filters, leading to considerable reduction of hardware complexity. This result is due to the fact that the input signal  $u(n)$  in the transposed structure is fed into all of the stochastic multipliers, where only one 'B2S' is required to convert this signal into the one based on stochastic computation.

Table 3 summarizes the required numbers of 'B2S's and 'S2B's in each  $N$ -th order stochastic FIR filter. The conventional method requires  $2(N + 1)$  'B2S's and one 'S2B'. Our previous work [11] also requires  $2(N + 1)$  'B2S's, but the number of 'S2B's is increased to  $N + 1$  because of the use of binary adders. Although the proposed strategy using the transposed structure has the same disadvantage as [11], this strategy is superior to both of the conventional and the previous methods in that the number of 'B2S's is significantly reduced from  $2(N + 1)$  to  $N + 2$ .

**Remark 2:** Instead of using the transposed structure, it is also possible to reduce the number of 'B2S's by another approach: in Fig. 5, if the data in each delay element follows stochastic format instead of binary format, it is easy to see that the resultant filter structure can reduce the number of 'B2S's into  $N + 2$ . Hence the required numbers of 'B2S's and 'S2B's in this approach respectively become the same as those in the method presented in this subsection. However, this approach is impractical from the viewpoint of implementation. As shown in [12], the structure using stochastic-format-based delay elements results in 10-fold increase in hardware complexity compared with the structure using binary-format-based delay elements. Therefore,



**Fig. 8** Coefficient-scaling technique.

throughout this paper it is assumed that each delay element follows binary format, and hence we conclude that the use of the transposed structure is an effective approach to reduce the hardware complexity of our previous method [11].

**Remark 3:** In the case of linear phase FIR filters, it is well known that the filter coefficients satisfy symmetry or anti-symmetry property [1]. In such a case, it is possible to share the multiplier for each pair of symmetric/antisymmetric coefficients, leading to further reduction of hardware complexity. However, we shall not deal with this issue because our proposed method is intended to be applicable to not only linear phase FIR filters but also other FIR filters of which coefficients do not satisfy symmetry/antisymmetry property.

#### 4.3 Coefficient-Scaling Technique

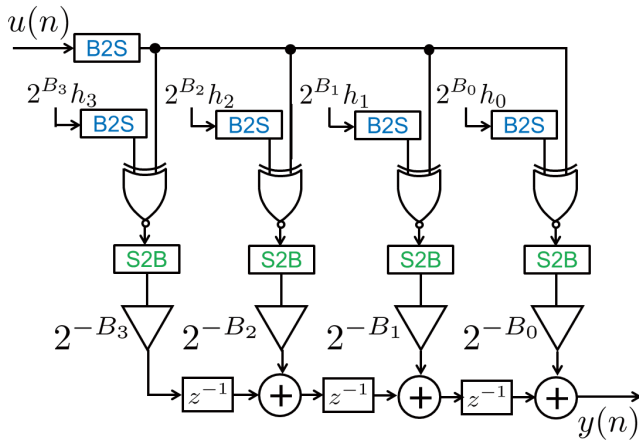
In order to solve the second problem addressed in Sect. 3, a coefficient-scaling technique is presented. This strategy is similar to the weight-scaling [19] that is intended for deep neural networks. In the method of [19], the weights are scaled up before performing stochastic multiplication in order to avoid large error caused by near-zero weights, and the products are scaled back down after performing accumulation.

Our coefficient-scaling technique is shown in Fig. 8. Here, each filter coefficient  $h_k$  given in binary format is first scaled up to be  $2^{B_k} h_k$  in order to avoid using near-zero coefficients in FIR filtering. The value  $B_k$  for  $0 \leq k \leq N$  is chosen to be the maximum non-negative integer such that the absolute value of  $2^{B_k} h_k$  becomes as close as unity under the constraint of  $|2^{B_k} h_k| < 1$ . For example, if  $h_k = 0.09$ , then  $B_k$  is set to be 3 and the filter coefficient is scaled up to be  $2^3 h_k = 0.72$ . Then, the scaled coefficient  $2^{B_k} h_k$  is converted to the stochastic data and multiplied by  $u(n)$  through the XNOR gate, and the product is converted back to the binary data. Finally, this binary data is scaled down by  $2^{-B_k}$ . This technique will significantly reduce the errors generated at the stochastic multiplications.

The 3rd-order FIR filter using this strategy together with the aforementioned two strategies is shown in Fig. 9. It should be noted that the coefficient-scaling technique does not increase the hardware complexity because the upscaled coefficients  $2^{B_k} h_k$  are given a priori, and the multipliers of  $2^{-B_k}$  after 'S2B's can be realized by logical shift right by  $B_k$  bits.

#### 5. Performance Evaluations

Here we shall carry out two performance evaluations for the



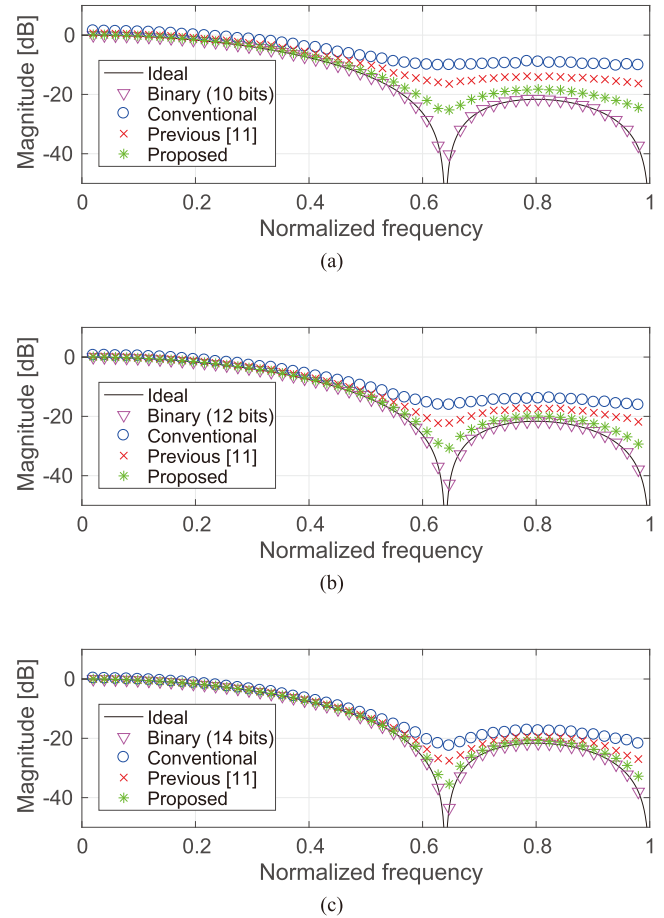
**Fig. 9** 3rd-order FIR filter using coefficient-scaling, transposed structure and stochastic/binary hybrid computation.

conventional and the proposed methods. First, the computational accuracy is evaluated by means of MATLAB simulations. Then, hardware evaluation is carried out through logic synthesis of each stochastic FIR filter using Synopsys Design Compiler on TSMC 65nm CMOS Technology.

### 5.1 Computational Accuracy Evaluation

Evaluation of the computational accuracy is carried out in terms of the magnitude responses of stochastic FIR filters. The magnitude response of each stochastic FIR filter is calculated using a sinusoidal wave as an input signal.

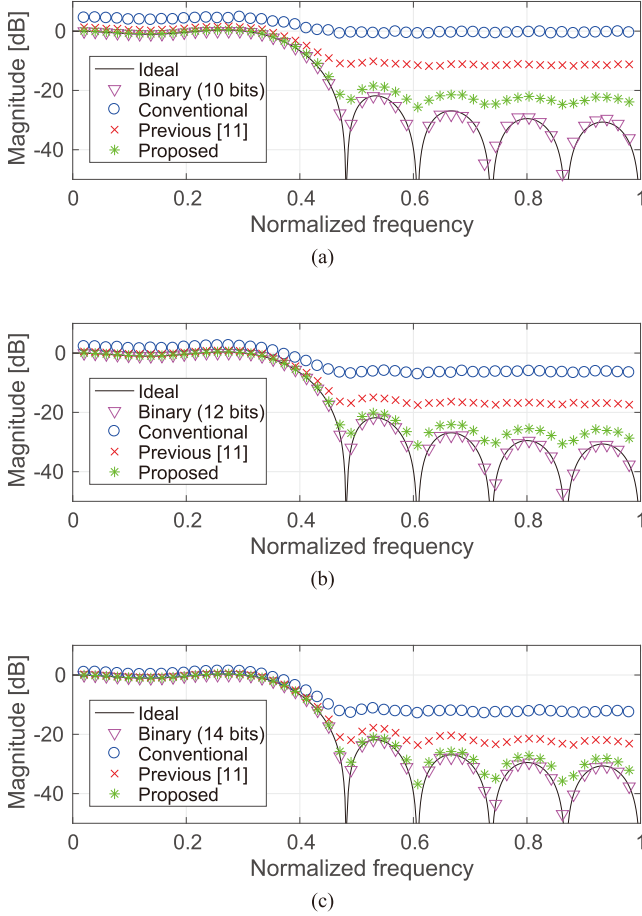
The evaluation results for a 3rd-order FIR filter are shown in Fig. 10, where ‘Ideal’, ‘Conventional’, ‘Previous [11]’, and ‘Proposed’ respectively correspond to the theoretical magnitude response, the evaluation result of the purely stochastic design, the one of the stochastic/binary hybrid design without coefficient-scaling and transposed structure [11], and the one for the stochastic/binary hybrid design using transposed structure and coefficient-scaling. Also, ‘Binary (10 bits)’, ‘Binary (12 bits)’ and ‘Binary (14 bits)’ that are respectively shown in Fig. 10(a), (b) and (c) correspond to the evaluation results for purely binary design in fixed-point arithmetic. The 3rd-order FIR filter given here is the low-pass filter designed by the window method, where the rectangular window is used and the cutoff frequency is set to be  $0.4\pi$  rad. Figure 10 tells us that the magnitude responses for purely binary design are very close to the ideal response, whereas the conventional purely stochastic design yields severe degradation of the magnitude response from the ideal one even if the bit length  $N_{sto}$  is increased to  $2^{14}$  bits. On the other hand, our previous work successfully improves the computational accuracy: in the case of  $N_{sto} = 2^{10}$  our previous work achieves a 5dB improvement in minimum stopband attenuation compared with the conventional method, and in the case of  $N_{sto} = 2^{14}$  a 2dB improvement is achieved. In addition, our proposed method further improves the computational accuracy: compared with the conventional method, the proposed method achieves a 10dB



**Fig. 10** Magnitude responses of 3rd-order stochastic FIR filters: (a)  $N_{sto} = 2^{10}$ , (b)  $N_{sto} = 2^{12}$ , and (c)  $N_{sto} = 2^{14}$ .

improvement for  $N_{sto} = 2^{10}$  and a 3dB improvement for  $N_{sto} = 2^{14}$  in minimum stopband attenuation.

Next, the evaluation results for a 15th-order FIR low-pass filter are shown in Fig. 11, where the filter design method, the filter cutoff frequency, and the value of  $N_{sto}$  are the same as in the previous example. In this case, the purely binary design also attains very accurate results, but the stochastic design results in more severe degradation of the magnitude responses than the 3rd-order filter. In particular, the conventional method fails to provide low-pass characteristic for  $N_{sto} = 2^{10}$ . Although increasing the bit length improves the accuracy to some extent, the conventional method still suffers from degradation and the resultant stopband attenuation is not satisfactory. It is conjectured that such severe degradation in the conventional method is due to the scaled adders: in 15th-order FIR filters the data of the convolution sum is scaled down by 16, leading to more severe degradation than the 3rd-order filters. On the other hand, our previous method and the proposed method are free from this scaling problem and hence the computational accuracy is improved. In particular, it should be noted that the proposed method gives a 20dB improvement for  $N_{sto} = 2^{10}$  and a 10dB improvement for  $N_{sto} = 2^{14}$  in minimum stopband attenuation, compared with the conventional method.

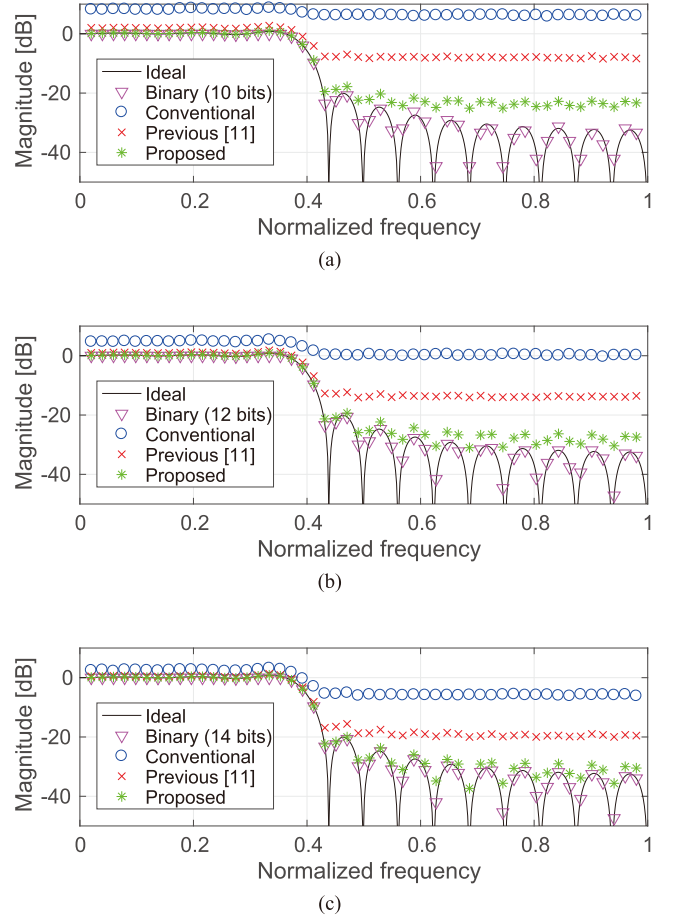


**Fig. 11** Magnitude responses of 15th-order stochastic FIR filters: (a)  $N_{\text{sto}} = 2^{10}$ , (b)  $N_{\text{sto}} = 2^{12}$ , and (c)  $N_{\text{sto}} = 2^{14}$ .

This improvement is due to the use of the coefficient-scaling technique together with the stochastic/binary hybrid computation.

Finally, much higher-order filters are evaluated because many practical applications use very high-order FIR filters that can realize sharp cutoff characteristics. Figures 12 and 13 respectively show the evaluation results for 31st-order FIR filters and 127th-order FIR filters, where the cutoff frequency and the window method are the same as in the previous evaluations. These results show that the use of higher-order filters results in more severe accuracy degradation than the previous results in all stochastic realizations<sup>†</sup>. Nevertheless, it is also clear from these results that the proposed method attains higher accuracy than any other stochastic realization, and that the proposed method keeps the sharp cutoff characteristics even in the higher-order cases. In addition, the proposed method also keeps the desired stopband attenuation (i.e. the maximum value of the stopband ripple) in both the 31st-order and the 127th-order filters. In the case of 127th-order filters with  $2^{10}$  bits, it should be noted that the proposed method achieves 40dB improvement in mini-

<sup>†</sup>Although the purely binary design still shows accurate results, in the 127th-order case we can see that the magnitude responses for the purely binary design slightly deviate from the ideal response.



**Fig. 12** Magnitude responses of 31st-order stochastic FIR filters: (a)  $N_{\text{sto}} = 2^{10}$ , (b)  $N_{\text{sto}} = 2^{12}$ , and (c)  $N_{\text{sto}} = 2^{14}$ .

imum stopband attenuation compared with the conventional method, and 20dB improvement compared with the previous method. Therefore we conclude that the utility of the proposed method is shown in practical high-order FIR filters as well as low-order filters.

## 5.2 Hardware Evaluation

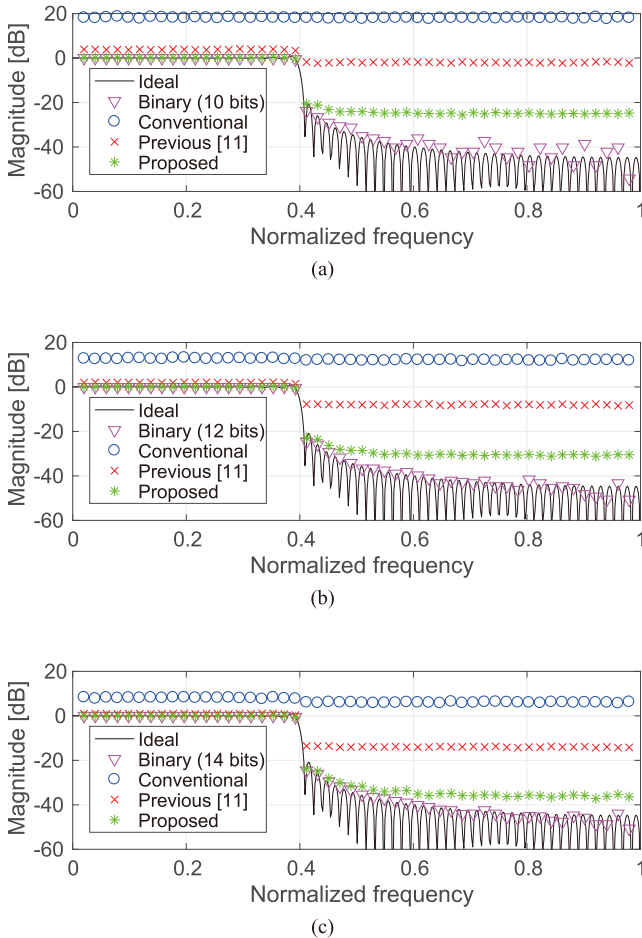
Table 4 shows performance comparisons of the 3rd-order fixed-point and stochastic FIR filters using TSMC 65nm CMOS technology, where the performance of the 15th-order filters is also summarized in Table 5. The stochastic filters are designed using  $N_{\text{sto}} = 2^{10}$  and the fixed-point filter exploits 10-bit signed fixed-point numbers. The FIR filters are designed using Verilog HDL, where the input and the output binary signals are represented by 10-bit signed fixed-point numbers. In the stochastic design, the binary signals are converted to stochastic bit streams using ‘B2S’ that includes a linear-feedback shift register (LFSR) and a digital comparator [5]. In this paper the bit length in an LFSR is set to be 11. The reason for this is that the stochastic signals in the FIR filters are of  $2^{10}$ -bit random numbers, which require at least 11 bits in an LFSR: note that a 10-bit LFSR can generate random sequences with the maximum possi-

**Table 4** Performance comparisons of 3rd-order FIR filters using TSMC 65nm CMOS technology.

	Conventional	Previous [11]	Proposed		Binary
Computation	stochastic		stochastic/binary		fixed-point
Feature	-	-	transposed structure	transposed structure + coefficient scaling	10-bit fixed point
Delay [ns]	1.2	1.15	1.11	1.16	2.16
Area [ $\mu\text{m}^2$ ]	2,206	3,468	2,918	2,838	4,825
Total power@100MHz [mW]	0.223	0.295	0.245	0.234	0.410
(Dynamic power)	0.213	0.278	0.231	0.220	0.381
(Static power)	0.010	0.017	0.014	0.014	0.029

**Table 5** Performance comparisons of 15th-order FIR filters using TSMC 65nm CMOS technology.

	Conventional	Previous [11]	Proposed		Binary
Computation	stochastic		stochastic/binary		fixed-point
Feature	-	-	transposed structure	transposed structure + coefficient scaling	10-bit fixed point
Delay [ns]	1.26	1.24	1.04	1.13	2.64
Area [ $\mu\text{m}^2$ ]	5,783	9,424	7,601	7,141	14,447
Total power@100MHz [mW]	0.621	0.835	0.638	0.636	1.596
(Dynamic power)	0.596	0.794	0.604	0.602	1.511
(Static power)	0.025	0.041	0.034	0.034	0.085

**Fig. 13** Magnitude responses of 127th-order stochastic FIR filters: (a)  $N_{\text{sto}} = 2^{10}$ , (b)  $N_{\text{sto}} = 2^{12}$ , and (c)  $N_{\text{sto}} = 2^{14}$ .

ble length of  $2^{10} - 1$ , which is not enough for generation of  $2^{10}$ -bit stochastic signals. The ‘S2B’ unit, which converts a stochastic bit-stream into the data in binary format,

is simply realized by a counter [5]. For performance evaluation, the FIR filters are synthesized using Synopsys Design Compiler to evaluate the worst-case delay time and area. The synthesized gate-level netlists are exploited to evaluate the power dissipation using Synopsys Power Compiler. The worst-case delay time is set to 5 ns in the logic synthesis phase. All the stochastic FIR filters operate at the similar delay time of around 1.2 ns, while the fixed-point filter operates at the worst-case delay of around 2.2 ns. Note that the stochastic filters take  $(N_{\text{sto}} + 1)$  cycles for a one-cycle operation of the fixed-point filter: the  $N_{\text{sto}}$  cycles are used for FIR filtering, and the last one cycle is used to reset the counter of ‘S2B’.

In terms of area, the conventional stochastic filter is the smallest as it exploits only stochastic computation. The previous filter increases the area by 54.9% and 63.0% in the 3rd-order and the 15th-order design, respectively, in comparison with the conventional filter because the additions are realized using binary logic. In contrast, the proposed filter based on the transposed structure reduces the area by 15.5% and 19.3% in the 3rd-order and the 15th-order design, respectively, compared to the previous filters thanks to the reduction of the number of ‘B2S’s. In addition, the coefficient scaling improves the computational accuracy with the similar hardware complexity. Compared with the fixed-point filters, the proposed stochastic filters achieve 43.8% and 50.6% reductions in area in the 3rd-order and 15th-order design, respectively.

In terms of power dissipation, the previous filter increases the power dissipation by 37.0% and 34.5% in the 3rd-order and the 15th-order design, respectively, in comparison with the conventional filter because of the binary adders. Using the transposed structure, the power dissipations are reduced by 18.3% and 23.6% in the 3rd-order and the 15th-order design, respectively, in comparison with the previous design. Compared with the fixed-point filters, the



proposed stochastic filters achieve 44.9% and 60.2% reductions in power dissipation in the 3rd-order and 15th-order design, respectively.

As a result, in case of the 15th-order design, the proposed filter achieves a 20dB improvement in computational accuracy with the area overhead of 23.4% and the power overhead of 2.4% in comparison with the conventional filter. It also achieves a 10dB improvement in computational accuracy, a 24.2% area reduction and a 23.8% power reduction in comparison with the previous design.

### 5.3 Energy Comparisons with Fixed-Point Design

The energy dissipations of the 10-bit fixed-point and the proposed stochastic filters with  $N_{\text{sto}} = 2^{10}$  are compared in case of the sampling frequency of 16 kHz. The sampling frequency of 16 kHz is often used for voice applications, such as VoIP [20]. As the stochastic filter takes  $(N_{\text{sto}} + 1)$  cycles for a one-cycle operation of the fixed-point filter, the operation frequency of the stochastic filter is 16,400 ( $= 16 \times 1,025$ ) kHz. For the energy evaluation, the rates of activity ( $R_{\text{act}}$ ) of the filters are defined as follows:

$$R_{\text{act}} = \frac{T_{\text{act}}}{T_{\text{act}} + T_{\text{idle}}}, \quad (8)$$

where  $T_{\text{act}}$  and  $T_{\text{idle}}$  are the active and idle time, respectively. Suppose that  $T_{\text{act}}$  is set to 1 for normalization. Considering  $R_{\text{act}}$ , the total energy dissipations of the filters are defined as follows:

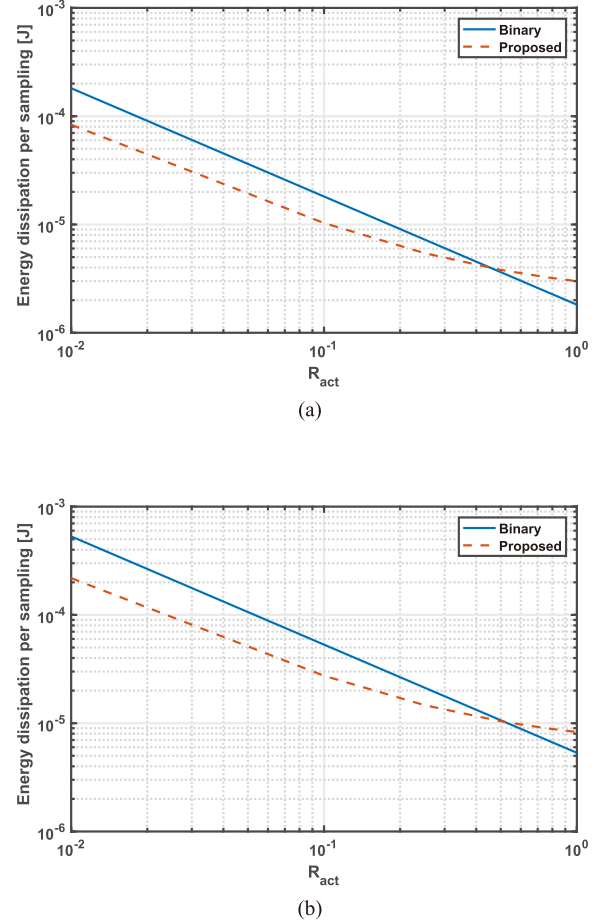
$$\begin{aligned} E_T &= E_{\text{act}} + T_{\text{idle}} \cdot E_{\text{idle}} \\ &= E_{\text{act}} + \left(\frac{1}{R_{\text{act}}} - 1\right) \cdot E_{\text{idle}}, \end{aligned} \quad (9)$$

where  $E_{\text{act}}$  and  $E_{\text{idle}}$  are the energy dissipations at the active and idle modes, respectively.

Figure 14 shows the energy dissipation per sampling vs.  $R_{\text{act}}$  in the 3rd-order and 15th-order FIR filters. In case of high  $R_{\text{act}}$  (e.g.  $R_{\text{act}} > 0.5$ ), the energy dissipation of the fixed-point filter is smaller than that of the proposed stochastic filter as  $E_{\text{act}}$  dominates  $E_T$ . In contrast, in case of low  $R_{\text{act}}$  that  $E_{\text{idle}}$  dominates  $E_T$ , the proposed stochastic filters achieve lower energy dissipations than the fixed-point filters.  $E_{\text{idle}}$  of the proposed filter are smaller than that of the fixed-point filters because of the smaller area.

## 6. Conclusion

This paper has presented FIR digital filters based on stochastic/binary hybrid computation. In our method, data additions in the filtering operation are performed based on binary computation instead of using the conventional scaled adders. This approach brings improvement of the computational accuracy of stochastic FIR digital filters. In addition, the transposed structure has been applied in order to reduce the hardware complexity caused by the stochastic/binary hybrid computation. Moreover, a coefficient-scaling technique has been presented to further improve the computational



**Fig. 14** Energy dissipation per sampling vs. rate of activity ( $R_{\text{act}}$ ) at the sampling frequency of 16 kHz: (a) 3rd-order and (b) 15th-order FIR filters.

accuracy in the FIR filters based on stochastic/binary hybrid computation. In the performance evaluation by MATLAB calculations, we have demonstrated that our proposed method successfully improves the performance of stochastic FIR filters, where our proposed method achieves at most 40dB improvement in minimum stopband attenuation compared with the conventional pure stochastic design.

Future works include modification of the realization of stochastic computation in FIR digital filters and development of other filter structures in order to achieve further improvement of computational accuracy. In addition, we plan to theoretically analyze the computational accuracy of stochastic digital filters.

## Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 15K06049. This work is supported by VLSI Design and Education Center (VDEC), The University of Tokyo with the collaboration with Synopsys Corporation.

## References

- [1] A.V. Oppenheim and R.W. Schaffer, *Discrete-Time Signal Processing*, 3rd Ed., Prentice-Hall, 2010.
- [2] K.K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*, John Wiley & Sons, 1999.
- [3] L.B. Jackson, *Digital Filters and Signal Processing*, 3rd edition, Kluwer Academic Publishers, Norwell, MA, 1996.
- [4] B.R. Gaines, "Stochastic computing systems," *Adv. Inf. Syst. Sci. Plenum*, vol.2, no.2, pp.37–172, 1969.
- [5] A. Alaghi and J.P. Hayes, "Survey of stochastic computing," *ACM Trans. Embed. Comput. Syst.*, vol.12, no.2s, Article 92, May 2013.
- [6] Y.-N. Chang and K.K. Parhi, "Architectures for digital filters using stochastic computing," *IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp.2097–2701, May 2013.
- [7] K.K. Parhi and Y. Liu, "Architectures for IIR digital filters using stochastic computing," *IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp.373–376, June 2014.
- [8] N. Saraf, K. Bazargan, D. Lilja, and M.D. Riedel, "IIR filters using stochastic arithmetic," 2014 Design, Automation and Test in Europe Conference and Exhibition (DATE), pp.1–6, March 2014.
- [9] N. Onizawa, S. Koshita, and T. Hanyu, "Scaled IIR filter based on stochastic computation," *IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp.297–300, Aug. 2015.
- [10] N. Onizawa, S. Koshita, S. Sakamoto, M. Abe, M. Kawamata, and T. Hanyu, "Gammatone filter based on stochastic computation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.1036–1040, March 2016.
- [11] S. Koshita, N. Onizawa, M. Abe, T. Hanyu, and M. Kawamata, "Realization of FIR digital filters based on stochastic/binary hybrid computation," *IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*, pp.223–228, May 2016.
- [12] Y. Liu and K.K. Parhi, "Architectures for recursive digital filters using stochastic computing," *IEEE Trans. Signal Process.*, vol.64, no.14, pp.3705–3718, July 2016.
- [13] B.D. Brown and H.C. Card, "Stochastic neural computation. I. Computational elements," *IEEE Trans. Comput.*, vol.50, no.9, pp.891–905, Sept. 2001.
- [14] V.C. Gaudet and A.C. Rapley, "Iterative decoding using stochastic computation," *Electron. Lett.*, vol.39, no.3, pp.299–301, Feb. 2003.
- [15] S.S. Tehrani, W.J. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol.10, no.10, pp.716–718, Oct. 2006.
- [16] N. Onizawa, W.J. Gross, T. Hanyu, and V.C. Gaudet, "Clockless stochastic decoding of low-density parity-check codes: Architecture and simulation model," *J. Signal Process. Syst.*, vol.76, no.2, pp.185–194, Aug. 2014.
- [17] P. Li, D.J. Lilja, W. Qian, K. Bazargan, and M.D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. VLSI Syst.*, vol.22, no.3, pp.449–462, March 2014.
- [18] N. Onizawa, D. Katagiri, K. Matsumiya, W.J. Gross, and T. Hanyu, "Gabor filter based on stochastic computation," *IEEE Signal Process. Lett.*, vol.22, no.9, pp.1224–1228, Sept. 2015.
- [19] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," *53rd ACM/EDAC/IEEE Annual Design Automation Conference (DAC)*, Article 124, June 2016.
- [20] S. Nagireddi, *VoIP Voice and Fax Signal Processing*, John Wiley & Sons, 2008.



**Shunsuke Koshita** received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 2001, 2003, and 2006, respectively. During the autumn of 2005, he was a Visiting Student at the Technical University of Sofia, Bulgaria. He is currently an Assistant Professor in the Graduate School of Engineering at Tohoku University. His research interests include linear system theory, control theory, circuit theory, and signal processing theory. Dr. Koshita received the Student Award Best Paper Prize from IEEE Sendai Section in 2004, the Best Ph.D. Award from the Graduate School of Engineering, Tohoku University in 2006, the Best Presentation Award of the SICE Tohoku Chapter Workshop in 2006, M. Ishida Foundation Research Encouraging Award in 2006, the Young Excellent Author Award of the 19th IEICE Workshop on Circuits and Systems in Karuizawa in 2007, and the IEICE Young Researcher's Award in 2008. He is a senior member of the IEEE and member of the Society of Instrument and Control Engineers of Japan.



**Naoya Onizawa** received the B.E., M.E. and D.E. degrees in Electrical and Communication Engineering from Tohoku University, Japan, in 2004, 2006 and 2009, respectively. He is currently an Assistant Professor in Frontier Research Institute for Interdisciplinary Sciences at Tohoku University, Japan. He was a postdoctoral fellow at University of Waterloo, Canada in 2011 and at McGill University, Canada from 2011 to 2013. In 2015, he was a Visiting Associate Professor at University of Southern Brittany, France. His main interests and activities are in the energy-efficient VLSI design based on asynchronous circuits and probabilistic computation, and their applications, such as associative memories and brain-like computers. He received the Best Paper Award in 2010 IEEE ISVLSI, the Best Paper Finalist 2014 IEEE ASYNC, 20th Research Promotion Award, Aoba Foundation for the Promotion of Engineering in 2014, and Kenneth C. Smith Early Career Award for Microelectronics Research in 2016 IEEE ISMVL. Dr. Onizawa is a Member of the IEEE.



**Masahide Abe** received the Bachelor of Engineering, Master of Information Sciences, and Doctor of Engineering degrees from Tohoku University, Sendai, Japan in 1994, 1996, and 1999, respectively. In 1999, he joined the Graduate School of Engineering at Tohoku University, Sendai, Japan, where he is currently an Associate Professor. His research interests include digital signal processing, image processing, adaptive digital filtering and evolutionary computation. He received the Young Engineer Award from the IEICE in 1997, the Young Excellent Author Award of the 13th IEICE Workshop on Circuits and Systems in Karuizawa in 2001. He is a member of the IEEE and the Society of Instrument, Control Engineers of Japan, and the Research Institute of Signal Processing, Japan. He is an IEEE Senior Member. He is an IEICE Senior Member.



**Takahiro Hanyu** received the B.E., M.E. and D.E. degrees in Electronic Engineering from Tohoku University, Sendai, Japan, in 1984, 1986 and 1989, respectively. He is currently a Professor in the Research Institute of Electrical Communication, Tohoku University. His general research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or PVT-variation-free VLSI processors, and multiple-valued current-mode circuit and its application to power-aware asyn-

chronous Network-on-Chip systems. He received the Sakai Memorial Award from the Information Processing Society of Japan in 2000, the Judge's Special Award at the 9th LSI Design of the Year from the Semiconductor Industry News of Japan in 2002, the Special Feature Award at the University LSI Design Contest from ASP-DAC in 2007, the APEX Paper Award of Japan Society of Applied Physics in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, Ichikawa Academic Award in 2010, the Best Paper Award of IEEE ISVLSI 2010, the Paper Award of SSDM 2012, and the Best Paper Finalist of IEEE ASYNC 2014. Dr. Hanyu is a Senior Member of the IEEE.



**Masayuki Kawamata** received the B.E., M.E., and D.E. degrees in electronic engineering from Tohoku University, Sendai, Japan, in 1977, 1979, and 1982, respectively. He was an Associate Professor in the Graduate School of Information Sciences at Tohoku University and is currently a Professor in the Graduate School of Engineering at Tohoku University. His research interests include 1-D and multidimensional digital signal processing, intelligent signal processing, control theory, and linear system

theory. He received the Outstanding Transaction Award from the Society of Instrument and Control Engineers of Japan in 1984 (with T. Higuchi), the Outstanding Literary Work Award from the Society of Instrument and Control Engineers of Japan in 1996 (with T. Higuchi), and the 11th IBM-Japan Scientific Award in Electronics in 1997. He is a member of the IEEE, the Society of Instrument and Control Engineers of Japan, the Information Processing Society of Japan, and the Institute of Image Information and Television Engineers of Japan. He is an IEEE Senior Member. He is an IEICE Fellow.