Low-Complexity Winograd Convolution Architecture Based on Stochastic Computing

Huizheng Wang[1,2,3,4], Zaichen Zhang[2,4], Xiaohu You[2], Chuan Zhang[1,2,4,*]
[1]Lab of Efficient Architectures for Digital-communication and Signal-processing (LEADS)
[2]National Mobile Communications Research Laboratory, Southeast University
[3]Chien-Shiung Wu College, Southeast University
[4]Quantum Information Center of Southeast University, Nanjing, China
Email: {hzwang, chzhang}@seu.edu.cn

***Abstract*—Deep convolution neural networks (CNNs) usually require a large number of iterative convolution operations, which would consume significant amounts of hardware resources. In this paper, we propose an efficient convolution architecture based on Winograd algorithm for convolutional neural networks (CNNs), by employing stochastic computing (SC). For the first step, a fast convolution algorithm, Winograd fast convolution algorithm (WFCA), which can lower the complexity by reducing multiplications is proposed. Although stochastic computing (SC) can achieve significant reduction in hardware complexity compared with the deterministic design, its straightforward application to fast convolution is not well-suited due to the precision loss. Therefore, based on two-line SC, this paper proposes a non-scaled stochastic adder which has higher computation accuracy than the conventional stochastic adder. Numerical results have proved the advantages of the proposed design in both complexity and precision. Although preliminary, it is expected that this design can be the first step of combining neural network and stochastic computing, which are both analog, belief-based and fault-tolerant, thereby unlocking the potentiality for the widespread application of stochastic Winograd algorithm in neural network systems.**

***Keywords*—Stochastic computing (SC), Winograd algorithm, two-line SC, fast convolution, convolutional neural network**

I. Introduction

It has been proved that convolutional neural network (CNN) is powerful in various applications like image processing, pattern recognition, and so on [1]. CNNs can achieve sufficient accuracy as long as the network is deep enough. However, for the convolution plays a vital role in CNN, complicated operations and numerous parameters are required for realization, which hinders its efficient hardware implementations and widespread applications[2].

Fast convolution is an implementation of convolution using fewer multipliers at the expense of more adders, since an adder consumes less area and computation time than a multiplier. However, this reduction in computation complexity is insufficient regarding the massive scale of neural networks. Power efficiency and computation performance are still expected to advance further. One approach is software implementation in GPUs for accuracy consideration. However, this contradicts the requirements of being real-time and mobile. Another approach is to implemented with ASIC, which is however highly sensitive to noise and variation [3].

To this end, SC is considered for fast convolution implementation. The most appealing feature of SC logic compared to deterministic logic is that it makes complicated arithmetic simpler and more elegant[4]. However, it is consensus that the value must be scaled down to stay within an appropriate interval $[0, 1]$, which would lead to precision loss[5].

In this paper, to deal with the dilemma, a high-accuracy SC adder based on two-line SC representation is proposed. With this high-accuracy SC adder, a fast convolution computing unit is constructed. It is demonstrated that the new SC architecture realizes a compromise between precision and complexity compared to traditional SC. Here, Winograd algorithm is taken as an example. Other fast convolution algorithms' [6] designs can be carried out in a similar fashion. This convolution design does not only employs the Winograd algorithm to reduce the complexity, but also makes use of stochastic computing to reduce the complexity on the premise of accuracy.

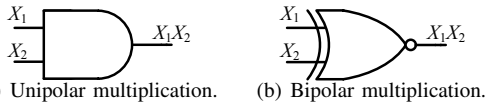
The reminder of this paper is organized as follows. Section II gives a brief review of the stochastic computing and Winograd algorithm. Section III shows the hardware realization and performance of a high-accuracy non-scaled stochastic adder. Section IV presents the Winograd convolution architecture. Numerical results, FPGA results, and relevant analyses are given in Section V. The conclusions are drawn in Section VI.

II. Preliminaries

A. Traditional Stochastic Computing

In this subsection, two kinds of traditional SC are discussed. Inherent drawbacks of them are introduced.

1) Unipolar SC: For *unipolar* SC [7], a number within range $[0, 1]$ can be represented by a length-N bit-stream containing X bits "1", with $x = X/N$. Fig. 1(a) shows the corresponding multiplier with this unipolar representation. Here, one AND gate is needed, thereby significantly reducing the complexity compared to the deterministic multiplier. However, positive-only limitation is a shortcoming for unipolar SC.



(a) Unipolar multiplication. (b) Bipolar multiplication.

Fig. 1. Traditional SC approaches.

2) Bipolar SC: To deal with the drawback of unipolar SC, bipolar representation was proposed, which includes the negative number into the framework of stochastic computing [8]. More specifically, for a number x within range $[-1, 1]$, its bipolar representation is $x = 2(X/N) - 1$, respectively. It should be noted that since the underlying representation has changed, the multiplier using bipolar representation (see Fig. 1(b)) is different from that using unipolar representation.

However, it is worth mentioning that both unipolar SC and bipolar SC employ MUX as a adder. As is depicted in Fig. 2, since probability has its own legitimate range, an L-input MUX scales its output down L times, which causes severe precision loss. As a result, the streams can only achieve the accuracy as low as $1/L$, which is not sufficient to represent the output when L is large. Random fluctuations, correlations, and other physical errors are also the sources of inaccuracy [8].

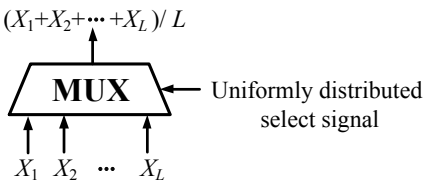


Fig. 2. Scaled addition of unipolar SC and bipolar SC.

B. Winograd Algorithm

The Winograd convolution algorithm is based on the *Chinese Remainder Theorem* of integer rings [6]. To be specific, choose a polynomial $m(p)$, whose ranks equal to that of $s(p)$, and decompose $m(p)$ into $k+1$ coprime polynomials $m^{(i)}(p)$.

$$m(p) = m^{(0)}(p)\ m^{(1)}(p)...m^{(k)}(p). \tag{1}$$

$$N^{(i)}(p)M^{(i)}(p) + n^{(i)}(p)m^{(i)}(p) = \gcd(M^{(i)}(p), m^{(i)}(p)) = 1. \tag{2}$$

Define $M^{(i)}(p) = \frac{m(p)}{m^{(i)}(p)}$, according to the Eq. (2), we can get $N^{(i)}(p)$. Then we can calculate Eq. (3):

$$h^{(i)}(p) = h(p) \mod m^{(i)}(p), \tag{3a}$$

$$x^{(i)}(p) = x(p) \mod m^{(i)}(p), \tag{3b}$$

where $i = 0, 1, \ldots, k$. In term of the $h^{(i)}(p)$ and $x^{(i)}(p)$, we can calculate:

$$s'^{(i)}(p) = h^{(i)}(p)x^{(i)}(p) \mod m^{(i)}(p). \tag{4}$$

By using Eq. (5), we can calculate $s'(p)$:

$$s'(p) = \sum_{i=0}^{k} s^{(i)}(p)\ N^{(i)}(p)\ M^{(i)}(p) \mod m(p). \tag{5}$$

Final, the convolution result $s(p)$ can be calculated as:

$$s(p) = s'(p) + h_{N-1}\ x_{N-1}\ m(p). \tag{6}$$

The coefficients of $s(p)$ are the convolution of x and h.

III. Proposed High-Accuracy SC Adder

Different from down-scaled stochastic adder, the proposed high-accuracy stochastic is based on a two-line SC representation [9]. Here the two-line representation is introduced first.

A. Two-Line SC Representation

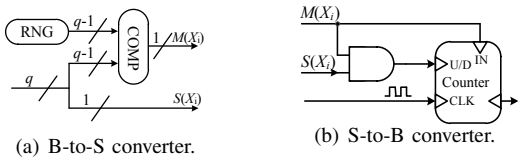
In two-line SC [10], x is represented by magnitude stream $M(X_i)$ and sign stream $S(X_i)$. In general, if the stream length is 2^L, these two bit-streams jointly represent x as Eq. (7).

$$x = \frac{1}{2^L} \sum_{i=0}^{2^L-1} (1 - 2S(X_i))M(X_i). \tag{7}$$

From Eq. (7), we can observe that different from unipolar or bipolar representation, the two-line SC representation introduces the concept of "$+1$" and "-1" for calculating the represented number. As discussed in next subsection, this difference further results in a different architecture of the stochastic arithmetic unit.

B. Binary to Stochastic Converter

In order to convert binary numbers to stochastic domain, the two-line-based SC needs a binary-to-stochastic (B-to-S) conversion unit. The overall architecture for the B-to-S unit which contains a random number generator (RNG) and a comparator is showed in Fig. 3(a). Here for each $2'$s complement number x, each bit in its sign stream is its own sign bit. Notice that this phenomenon only exists when x is converted to two-line bit-stream, while in the SC system each bit in the sign stream is not necessarily the same.



(a) B-to-S converter. (b) S-to-B converter.

Fig. 3. Architectures for two-line SC.

C. Two-Line SC Multiplier (TSM)

Note $A \in [-C_1, C_1]$, $B \in [-C_2, C_2]$, where C_1 and C_2 are positive real numbers. Define $C = AB$, then we have [11]:

$$C \longrightarrow P_C = \frac{A\ B}{C_1 \times C_2} = P_A\ P_B \tag{8}$$

For the magnitude sequence, $M(C_i) = M(A_i)M(B_i)$, which can realized by an AND gate. However, in sign sequence, '0' represents positive number while '1' indicates minus number, so in term of the multiplication characteristics, an XOR gate can achieve the calculation. Therefore, the multiplication circuit of two-line can be designed as Fig. 4.

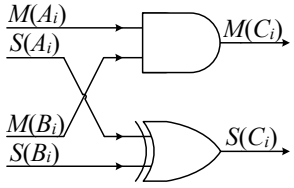


Fig. 4. Two-line stochastic multiplier.

D. High-Accuracy Non-Scaled Stochastic Adder (NSSA)

Based on the two-line SC representation, a high-accuracy non-scaled stochastic adder is proposed. The key thought of the proposed adder is based on the unique representation method of two-line SC. From Eq. (7), we can observe the i-th bits of magnitude stream and sign stream jointly contribute $(1-2S(X_i))M(X_i)$ to x. If we denote $A_i=(1-2S(A_i))M(A_i)$ and $B_i=(1-2S(B_i))M(B_i)$ as the i-th contribution for the two inputs a and b, respectively, the i-th contribution of their sum $c=a+b$, refered as C_i, should be the element of $\{-1,0,1\}$ plus the carry bit. Consequently, if we utilize a three-state counter to store the positive or negative carry bit, then we can get the following truth table for C_i in the Table I.

TABLE I. TRUTH TABLE FOR C_i

A_i+B_i	Current counter	Next counter	C_i
-2	1	0	-1
-2	0	-1	-1
-2	-1	-1	-1
-1	remains	remains	-1
0	1	0	1
0	0	remains	0
0	-1	0	-1
1	remains	remains	1
2	1	1	1
2	0	1	1
2	-1	0	1

TABLE II. GENERATION SCHEME FOR OUTPUT STREAMS

C_i	$S(C_i)$	$M(C_i)$
1	0	1
0	0 or 1	0
-1	1	1

Notice that here since C_i is the contribution of the i-th bits of sign and magnitude bit-streams, we have $C_i=(1-2S(C_i))M(C_i)$. Accordingly, $S(C_i)$ and $M(C_i)$ can be determined as Table II. Based on Tables I and II, the hardware architecture of the proposed non-scaled stochastic adder (NSSA) is shown in Fig. 5. The whole algorithm is concluded in Algorithm 1.

There are two main advantages of the proposed adder in Fig. 5 compared with the conventional stochastic adder in Fig. 2. Firstly, it is a non-scaled adder. Although OR gate can be used as approximated non-scaled adder in some application [7], the OR-gate-based adder suffers from huge error as well as limitation for positive-only addition, thereby causing the OR-gate-based adder to have very severe accuracy loss if taking the negative inputs into consideration (see Fig. 6). Secondly, the proposed stochastic adder has very high accuracy in term of signal-to-noise ratio (SNR), especially for the adder chain case. Although when the L is 2, the MUX-based adder has better computation accuracy than the proposed two-line based design for single adder case as shown in Fig. 6, its performance degrades significantly if the application needs the use of adder chain as depicted in Fig. 7, whereas the accuracy of the two-line-based design still remains at a relatively high level.

E. Stochastic to Binary Converter

Besides B-to-S conversion unit, the realization for Winograd fast convolution algorithm (WFCA) also needs S-to-B conversion unit. Fig. 3(b) shows its hardware circuit, which is generally the same as the conventional counterpart.

Algorithm 1 Non-scaled stochastic adder

Input:
Sequence A referred as $S(A_i)$ and $M(A_i)$.
Sequence B referred as $S(B_i)$ and $M(B_i)$.
Output:
Sequence C referred as $S(C_i)$ and $M(C_i)$.

for $i=1:1:N$ **do**
  $A_i=(1-2S(A_i))M(A_i)$
  $B_i=(1-2S(B_i))M(B_i)$
  C_i=A_i+B_i
  if $(A_i+B_i+Counter(i)>1)$ **then**
    $C_i=1$
    $Counter(i+1)=1$
  else if $(A_i+B_i+Counter(i)<-1)$ **then**
    $C_i=-1$
    $Counter(i+1)=-1$
  else if $(A_i+B_i+Counter(i)=1)$ **then**
    $C_i=1$
    $Counter(i+1)=0$
  else if $(A_i+B_i+Counter(i)=-1)$ **then**
    $C_i=0$
    $Counter(i+1)=1$
  else if $(A_i+B_i+Counter(i)=0)$ **then**
    $C_i=0$
    $Counter(i+1)=0$
  end if
end for
The sum of A and B is represented in C_i.

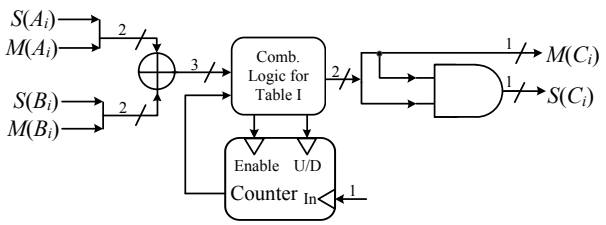


Fig. 5. Hardware architecture of the proposed stochastic adder.

IV. PROPOSED WINOGRAD CONVOLUTION ARCHITECTURE

A. Fast Convolution Unit

As discussed in Section II, a 2×3 Winograd fast convolution unit (WFCU) can be constructed. Consider two sequences, $h=h_0,h_1$ and $x=x_0,x_1,x_2$, they can be expressed as:

$$\begin{cases} h(p)=h_1p+h_0, \\ x(p)=x_2p^2+x_1p+x_0. \end{cases} \tag{9}$$

The result of convolution can be written as a polynomial product:

$$\begin{aligned} s(p)&=h(p)\cdot x(p) \\ &=s_3p^3+s_2p^2+s_1p+s_0. \end{aligned} \tag{10}$$

Choose $m(p)=p(p-1)(p+1)$, then $m^{(i)}(p)$ is

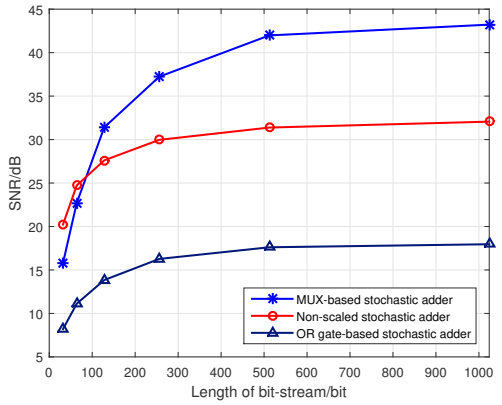


Fig. 6. SNR of different stochastic adders for single adder case.

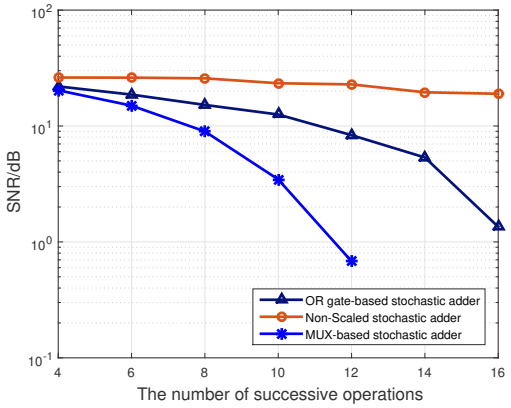


Fig. 7. SNR of different stochastic adders for addition chain. Here the length of bit-stream is 1024.

$$\begin{cases} m^{(0)}(p) = p, \\ m^{(1)}(p) = p-1, \\ m^{(2)}(p) = p+1. \end{cases} \tag{11}$$

Moreover, due to $M_{(i)}(p) = \frac{m(p)}{m^{(i)}(p)}$, $i = 0, 1, 2$, and Eq. (2), Table III can be obtained:

TABLE III. CALCULATION PARAMETER OF 2×3 WFCU

i	$m^{(i)}(p)$	$M^{(i)}(p)$	$n^{(i)}(p)$	$N^{(i)}(p)$
0	p	p^2-1	p	-1
1	$p-1$	p^2+p	$-\frac{1}{2}(p+2)$	$\frac{1}{2}$
2	$p+1$	p^2-p	$-\frac{1}{2}(p-2)$	$\frac{1}{2}$

Thus, the remainder can be calculated by Eq. (3) as follows:

$$\begin{cases} h^{(0)}(p) = h_0, & x^{(0)}(p) = x_0; \\ h^{(1)}(p) = h_0 + h_1, & x^{(1)}(p) = x_0 + x_1 + x_2; \\ h^{(2)}(p) = h_0 - h_1, & x^{(2)}(p) = x_0 - x_1 + x_2. \end{cases} \tag{12}$$

With $h^{(i)}(p)$, $x^{(i)}(p)$, and Eq. (4), $s'^{(i)}(p)$ can be obtained:

$$\begin{cases} s'^{(0)}(p) = h_0 x_0, \\ s'^{(1)}(p) = (h_0 + h_1)(x_0 + x_1 + x_2), \\ s'^{(2)}(p) = (h_0 - h_1)(x_0 - x_1 + x_2). \end{cases} \tag{13}$$

Therefore, the result of convolution can be seen as follows:

$$\begin{cases} s_0 = h_0 x_0, \\ s_1 = h_0 x_1 + h_1 x_0, \\ s_2 = h_0 x_2 + h_1 x_1, \\ s_3 = h_1 x_2. \end{cases} \tag{14}$$

It is worth mentioning that the 2×3 WFCU can be extended to $L \times N$ [6].

B. Efficient Hardware Architecture

Based on the proposed High-accuracy adder and two-line SC multiplier, the whole architecture of 2×3 WFCU is shown as Fig. 8.

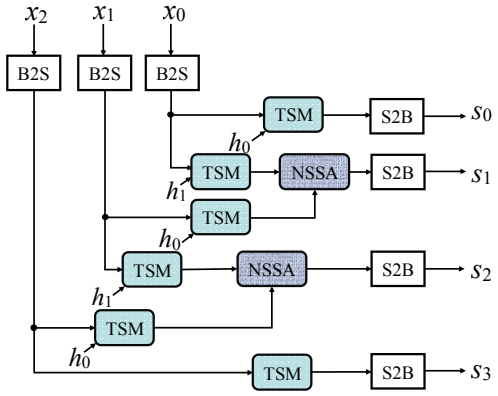


Fig. 8. The proposed architecture of two-line SC-based 2×3 Winograd.

V. PERFORMANCE AND ANALYSIS

A. Computation Accuracy

Since for a 2×3 WFCU can be extended to $L \times N$ WFCU [6]. We apply traditional SC [7] and Two-line SC into 2×3-point WFCU to compare their performance. The mean squared error (MSE) is used to compare the error degree of output results regarding the binary computing results as the standard value. As it is seen from Fig. 9, with the increasing of sequence length, the computing precision of all the SC design improves. However, in the case of the same sequence length, the traditional SC including unipolar SC and bipolar SC performs worst, while two-line SC design can achieve higher accuracy comparatively. To conclude, compared to the traditional stochastic SC-based WFCA designs, the proposed two-line-based designs shows significant advantages on computation accuracy.

B. Latency Time Analysis

Table V gives a comparison on latency between traditional SC and two-line SC. It is measured by the clock cycles in a whole multiplication or addition. N_1 and N_2 denote the stream length of traditional SC and two-line SC respectively. Apparently, with the same length of stream, traditional SC delays less time than two-line SC. However, as it depicted in Fig. 9, to achieve equal MSE, the length of traditional SC should be four times longer than that of two-line SC at least.

TABLE IV. HARDWARE COMPLEXITY COMPARISON

Convolution scales	Methods	Multiplications (LUT)	Additions (LUT)	Total consumption (LUT)
2×3	DCUs with binary computing	432	30	462
	WFCUs with binary computing	288	60	348
	WFCUs with traditional SC	4	12	16
	WFCUs with two-line SC	8	24	32
$L \times N$	DCUs with binary computing	$72 \times L \times N$	$10 \times \max\{L, N\}$	$72LN + 10\max\{L, N\}$
	WFCUs with binary computing	$72 \times (L + N - 1)$	$10 \times L \times N$	$72(L + N - 1) + 10LN$
	WFCUs with traditional SC	$L + N - 1$	$2 \times L \times N$	$(L + N - 1) + 2LN$
	WFCUs with two-line SC	$2 \times (L + N - 1)$	$4 \times L \times N$	$2(L + N - 1) + 4LN$

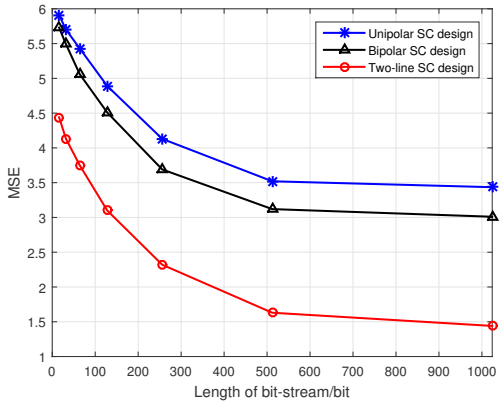


Fig. 9. The output accuracy of proposed fast convolution architecture using different stochastic computing designs.

TABLE V. CLOCK CYCLES FOR DIFFERENT OPERATIONS

Methods	Addition latency	Multiplication latency
Traditional SC	N_1	N_1
Two-line SC	$2N_2$	N_2

C. Computation Hardware Complexity Analysis

After successful FPGA verification, we re-generate the HDL code to evaluate the detailed hardware resources in term of equivalent LUTs. For this evaluation, both the binary computing multiplier and adder are 8 bits and the test is implemented on Xilinx Spartan-3E FPGA platform with XC3S250E chip. The comparison of the hardware complexity using different convolution methods with different sizes is shown in Table IV. Here DCU denotes direct convolution unit.

From Table IV, we can see that in the same conditions, WFCUs reduce the number of multiplications from LN to $L + N - 1$. Though the number of addition increases, the reduction in multiplication still contributes to significant savings for an adder consumes less LUTs than a multiplier in FPGA. More importantly, based on stochastic computing, the hardware consumption is around merely 9.1% of traditional binary computing. It worth mentioning that although traditional SC [7] has lower hardware complexity than proposed two-line SC, as discussed in Section V-A, it usually introduces unbearable precision loss. Thus, the proposed SC design can achieve a balance between precision and hardware consumption.

VI. CONCLUSION

In this paper, we focus on an efficient convolution architecture combining the fast convolution algorithm with stochastic computing. In the first place, we choose the Winograd algorithm to improve that convolution efficiency. Then, due to the precision loss of traditional SC, we employ two-line SC design. Implementation results have shown the design's advantages in low complexity and good accuracy.

ACKNOWLEDGEMENT

This work is supported in part by NSFC under grants 61871115 and 61501116, Jiangsu Provincial NSF for Excellent Young Scholars, Huawei HIRP Flagship under grant YB201504, the Fundamental Research Funds for the Central Universities, the SRTP of Southeast University, State Key Laboratory of ASIC & System under grant 2016KF007, ICRI for MNC, and the Project Sponsored by the SRF for the Returned Overseas Chinese Scholars of MoE.

REFERENCES

[1] Z. Zeng and J. Wang, *Advances in Neural Network Research and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2010.
[2] W. Xu, Z. Wang, X. You, and C. Zhang, "Efficient fast convolution architectures for convolutional neural network." *2017 IEEE 12th International Conference on ASIC (ASICON)*, p. 904, 2017.
[3] P. Li and D. J. Lilja, "Using stochastic computing to implement digital image processing algorithms," in *Proc. IEEE International Conference on Computer Design (ICCD)*, 2011, pp. 154 – 161.
[4] B. Yuan, C. Zhang, and Z. Wang, "Design space exploration for hardware-efficient stochastic computing: A case study on discrete cosine transformation." *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 6555, 2016.
[5] R. Xu, B. Yuan, X. You, and C. Zhang, "Efficient fast convolution architecture based on stochastic computing." *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, p. 1, 2017.
[6] H. J. Nussbaumer, *Fast Fourier transform and convolution algorithms*. Springer Science & Business Media, 2012, vol. 2.
[7] B. R. Gaines *et al.*, "Stochastic computing systems," *Advances in information systems science*, vol. 2, no. 2, pp. 37–172, 1969.
[8] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Trans. Embed. Comput. Syst. (TECS)*, vol. 12, no. 2s, pp. 92:1–92:19, 2013.
[9] J. Yang, C. Zhang, S. Xu, and X. You, "Efficient stochastic detector for large-scale MIMO," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6550–6554.
[10] B. Yuan and Y. Wang, "High-accuracy FIR filter design using stochastic computing," in *Proc. IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2016, pp. 128–133.
[11] P.-S. Ting and J. P. Hayes, "Stochastic logic realization of matrix operations," in *Proc. Euromicro Conference on Digital System Design (DSD)*, 2014, pp. 356–364.