Low-Cost and Highly-Efficient Bit-Stream Generator for Stochastic Computing Division

Mehran Shoushtari Moghadam, *Graduate Student Member, IEEE*, Sercan Aygun, *Member, IEEE*, Sina Asadi, and M. Hassan Najafi, *Senior Member, IEEE*

***Abstract*—Stochastic computing (SC) division circuits have gained importance in recent years compared to other arithmetic circuits due to their low complexity as a result of an accuracy tradeoff. Designing a division circuit is already complex in conventional binary-based hardware systems. Developing an accurate and efficient SC division circuit is an open research problem. Prior work proposed different SC division circuits by using multiplexers and JK-flip-flop units, which may require correlated or uncorrelated input bit-streams. This study is primarily centered on exploring a cost-effective and highly efficient bit-stream generator specifically designed for SC division circuits. In conjunction with this objective, we assess the performance of multiple bit-stream generators and analyze the impact of correlation on SC division. We compare different designs in terms of accuracy and hardware cost. Moreover, we discuss a low-cost and energy-efficient bit-stream generator via powers-of-2 Van der Corput (VDC) sequences. Among the tested sequence generators, our best results were achieved with VDC sequences. Our evaluation results demonstrate that the novel VDC-based design yields promising outputs, resulting in a 15.5% reduction in the area-delay product and an 18.05% saving in energy consumption for the same accuracy level compared to conventional bit-stream generators. Significantly, our investigation reveals that employing the proposed generator improves the precision compared to the state-of-the-art. We validate the proposed architecture with an image processing case study, achieving high PSNR and structural similarity values.**

***Index Terms*—Division, image processing, low-discrepancy sequences, random number generation, stochastic computing.**

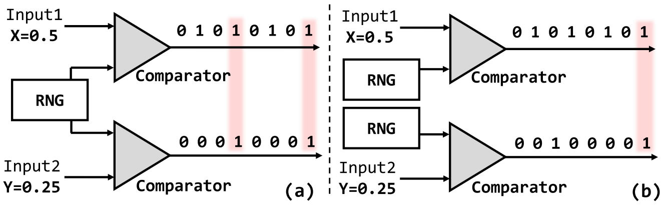


Fig. 1. Generating bit-streams, (a) by sharing a common RNG (correlated case), and (b) with different RNGs (uncorrelated case).

I. Introduction

Stochastic Computing (SC) has emerged as an alternative computing paradigm, drawing attention to low-cost and noise-tolerant hardware designs for complex arithmetic operations [1]. In SC, numbers are represented using uniform random bit-streams. Unlike traditional binary computing, which operates on positional inputs, SC processes bit-streams with no significant digit. In this unconventional representation, the ratio of the number of 1s to the length of the bit-stream determines the data value. By harnessing the information conveyed by `logic-1` and `logic-0` and employing unipolar or bipolar encoding [2], complex arithmetic operations can be realized with simple logic circuits.

Bit-stream generation is an essential step in SC, which directly affects the accuracy of the computations. To generate stochastic bit-streams, a *random number generator* (RNG) alongside a binary comparator is utilized. *Correlated* bit-streams, with high overlap in the position of 1s, can be generated by sharing a common RNG between different inputs, as depicted in Fig. 1(a). Using a different RNG for each input can produce *uncorrelated* bit-streams as shown in Fig. 1(b). Correlation or independence directly impacts the accuracy and functionality of stochastic circuits [3]. For instance, an AND gate is used as an SC multiplier when independent bit-streams are connected to its inputs. The same gate acts as a minimum operator when there is a high (positive) correlation (i.e., maximal overlap) between input bit-streams [4].

The initial development of SC-based dividers is attributed to Gaines's adaptive digital element (ADDIE) design [1]. An alternative approach involves the use of a simple JK flip-flop [5], [6]. Chen and Hayes [7] explored a low-cost stochastic division architecture known as *Correlated Division* (CORDIV), which takes advantage of correlation. Chu [8] proposed a *Saturating Subtractor Division* (SSDIV) design incorporating a saturating subtractor circuit alongside a JK flip-flop. Wang et al. [9] introduced another stochastic division design that evaluates $\frac{\min(X,Y)}{\max(X,Y)}$ as the output bit-stream where X and Y are stochastic dividend and divisor bit-streams, respectively. We call this Min-Max-based SC division or *MMDIV*.

The focus of this work, distinct from prior work, is to develop an effective bit-stream generator that can work for all state-of-the-art (SOTA) SC division circuits. Previous designs

Manuscript received 16 October 2023; revised 18 December 2023; accepted 16 January 2024. Date of publication 26 January 2024; date of current version 7 March 2024. This work was supported in part by the National Science Foundation (NSF) under Grant 2339701 and Grant 2019511, in part by the Louisiana Board of Regents Support Fund under Grant LEQSF(2020-23)-RD-A-26, and in part by Generous Gifts from Cisco, Xilinx, and Nvidia. The review of this paper was arranged by guest editors of the Special Issue for NanoArch2022. *(Corresponding author: Sercan Aygun.)*

The authors are with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA 70503 USA (e-mail: m.moghadam@louisiana.edu; sercan.aygun@louisiana.edu; sina.asadi1@louisiana.edu; najafi@louisiana.edu).

Digital Object Identifier 10.1109/TNANO.2024.3358395

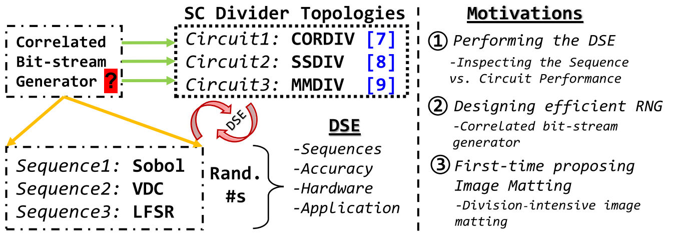


Fig. 2. Proposed exploration and the motivations.

TABLE I
CHRONOLOGICAL DEVELOPMENT OF SC DIVIDERS (FF: FLIP-FLOP)

SC Divider	Design Details	Key Aspect
ADDIE [1] *1969*	1×Counter + 3×RNG + 1×AND	Dynamic unipolar division
Ananth's [26] *2003*	1×Counter + 3×Weighted binary generators + 1×AND	Reduced-error SC number conversion
CORDIV [7] *2016*	1×RNG + 1×MUX + 1×D-FF	Correlated division
SSDIV [8] *2020*	1×RNG + 1×NOT + 1×AND + 1×JK-FF	Saturating subtractor for JK-FF fine-tuning
MMDIV [9] *2022*	1×RNG + 1×JK-FF + 1×AND + 1×XOR + delay element(s)	Delay insertion for decorrelation

Unipolar bit-stream representation is referenced for all of the designs.

used two comparators and a shared RNG (see Fig. 1(a)) to generate correlated bit-streams. In contrast, our novel design incorporates a more accurate division operation employing an efficient bit-stream generator comprising a counter, an RNG, and a comparator. We validate the designed circuit with an image matting case study. To the best of our knowledge, this is the first time image matting is implemented in SC literature. Image matting, based on the problem of separating foreground and background sections in composite images, is a method used to determine the blending ratio of pixel values (*alpha* values) at the intersection points of two composite images [10]. This image processing task requires many division operations. We explore three exemplary topologies in the divider circuits: CORDIV, SSDIV, and MMDIV as SOTA designs that share the common characteristic of being fed with correlated bit-streams.

The majority of prior SC division circuits work with correlated bit-streams, offering both low-cost and highly accurate results. However, some studies, like [11], also employ uncorrelated bit-streams. We explore the performance of different SOTA circuits depicted in Fig. 2. We examine different types of bit-streams and bit-stream generators from linear-feedback shift registers (LFSRs) [12] to Finite-State Machine (FSM)-based [13] generators. We employ high-quality random source generators such as Sobol [14] and Van der Corput (VDC) [15] that provide low discrepancy (LD) quasi-orthogonal bit-streams. LD sequences are known in the literature as random sources that achieve high accuracy for SC-based arithmetic [16]. There are prior hardware designs for efficient generation of LD sequences [13], [17], [18], [19]. Overall, our solution achieves a lower arithmetic error of up to 88% improvement when utilizing the simplest Sobol sequence with the SOTA division circuits. In summary, the main contributions of this study are as follows:

❶ We explore an adaptive and versatile bit-stream generator (sequence generator [20] + correlator [21]) that conforms to the well-known SOTA SC division circuits.

❷ The presented bit-stream generator is compatible with different random sequences. We provide a comparative benchmark with different types of sequences and SOTA circuits.

❸ For the first time, we implement the image matting problem, a challenging application that requires extensive division operations, in the SC domain.

The remaining sections are organized as follows: Section II provides an overview of the fundamentals of SC and the division operation. Section III introduces the architecture of the proposed bit-stream generator. Section IV presents a comprehensive design space exploration (DSE) by combining the bit-stream generation step with different quasi-random (Sobol and VDC) and pseudo-random (LFSR-based) numbers and the SOTA SC dividers. Section V evaluates the performance of the overall division architecture using the SC-based image matting case study. Section VI presents the findings from a hardware perspective. Finally, Section VII concludes the study.

II. BACKGROUND

SC has emerged as a promising model of computation, offering remarkable advantages such as resilience to noise, high parallelism, and power efficiency. SC has found applications in various domains, including image processing [22], sorting [4], and machine learning [23], among others [2]. The key strength of SC lies in its ability to perform complex arithmetic operations using simple logic gates, leading to significant cost savings in hardware implementation. A crucial step in SC systems is the conversion of real numbers into bit-streams, wherein each bit position holds equal significance, distinguishing it from conventional binary representation. SC operates on data within the unit interval $[0, 1]$. For instance, a data value of 0.5 is represented by a bit-stream with 50% of its bits set to 1. This encoding format is known as unipolar encoding (UPE). In UPE, the probability of observing a '1' in the bit-stream X, denoted as $P(X = 1)$, equals the input value x. Typically, generating a bit-stream of length N involves comparing N random numbers $(R_1 \ldots R_N)$ with the input value over N cycles. If the input value is greater than the random number, a logic 1 is produced; otherwise, a logic 0 is produced. The occurrence of logic 1s in the resulting bit-stream depends on the sequence of random numbers. In scenarios involving signed values (x within the range $-1 \leq x \leq 1$), bipolar encoding (BPE) is employed [2].

Multiplication of bit-streams in UPE is achieved through bitwise AND operation [3]. To ensure accurate multiplication, the input bit-streams must be uncorrelated. Performing bitwise AND on correlated bit-streams, i.e., bit-streams with the maximum overlap in the positions of 1s, yields the input bit-stream with minimum value [24]. Scaled addition in SC is realized using a multiplexer (MUX) unit for both encoding formats [25]. While the main inputs of the MUX can be correlated, they should be uncorrelated with the select input bit-stream.

SC supports a wide range of arithmetic operations, including division. Table I summarizes the important prior SC division

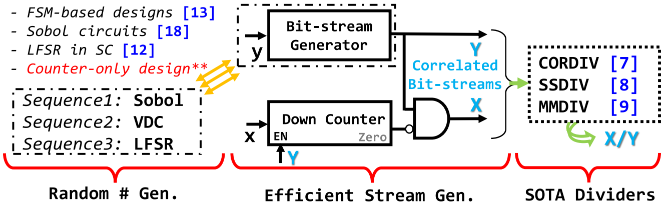


Fig. 3. Proposed approach incorporating a *bit-stream generator* (including random sequence sources -Sobol, VDC, LFSR- [20]) and *correlator* with a down counter to generate the correlated second bit-stream [21]. It is noted that $X < Y$ for the SC division operation ($\frac{X}{Y}$). **Among various options, the VDC stands out as a promising candidate for counter-based-only hardware design targeting random number generation.

circuits. Gaines's pioneering ADDIE design [1] stands as the first SC-based divider. Ananth's subsequent iteration [26] builds upon Gaines's ADDIE, aiming for improved stochastic number conversion with reduced error. Expanding on this foundation, Chen and Hayes [7] introduced CORDIV, leveraging correlation for stochastic division. CORDIV optimizes area cost through a shared RNG, one MUX, and a single D flip-flop. Chu [8] refines division with SSDIV. He demonstrates a correlation-based approach by exploiting saturating subtractors to fine-tune approximations from JK flip-flops. The MMDIV design [9] enhances accuracy by incorporating delay elements (DEs), effectively decorrelating inputs of JK flip-flop through a concatenation of D flip-flops.

In addition to more complex designs, simple yet effective division circuits have been investigated in the SC literature. For instance, a JK flip-flop can serve as an approximate divider [5], [6]. By applying the input bit-streams $X1$ and $X2$ to the J and K ports of the JK flip-flop, respectively, and setting the probabilities P_J and P_K, the output bit-stream Y is obtained from the Q port, resulting in a probability $P_Y = P_J/(P_J + P_K)$ [6], [27]. Although this provides an approximate division, recent efforts have focused on improving the accuracy of the JK flip-flop divider by converging toward $P_Y = P_J/P_K$ [8], [9], [11]. These prior endeavors highlight the ongoing research and development aimed at advancing division circuits in SC. By harnessing the inherent properties of SC, such as correlation, and utilizing various components, such as RNGs, comparators, MUX, and flip-flops, researchers are continuously striving to enhance the accuracy and efficiency of SC division.

III. Proposed Approach

The conventional approach for generating correlated bit-streams (Fig. 1(a)) uses a shared RNG with two comparators. This work employs a design with 1×*RNG* + 1×*Comparator* (= *Bit-stream Generator*) for the first and intermediary stream that affects the second stream. The design is combined with a down counter to generate the second bit-stream with a proper correlation. Fig. 3 illustrates our proposed solution. The first bit-stream generator determines the RNG for any of the *Sequences* in Fig. 2 (Sobol, VDC, or LFSR) and correlates the first generated bit-stream through the down counter. The foundation of this design, established in [21], emphasizes the importance of examining the design and its performance with different types of random source generators and division topologies. Particularly, using new random sources such as VDC is promising due to their counter-only design and offering an accuracy close to the Sobol-based design but with significantly lighter hardware. We differentiate from the prior art at this point and concentrate on the bit-stream generator.

Algorithm 1: Efficient Correlated Bit-Stream Generation [21].

```
Input: Dividend $0 \leq X \leq N$, Divisor $0 < Y \leq N$, Bit-stream size $N$
Output: $X_{\text{stream}}, Y_{\text{stream}}$
Initialize $LD\_seq$ as any low-discrepancy sequence (or LFSR)
for $i = 1$ to $N$ do
    if $(\frac{Y-1}{N} > \text{LD\_seq}(i))$ then
        $Y_{\text{stream}}(i) = 1$
    end if
end for
Set $m = X$
for $j = 1$ to $N$ do
    if $m > 0$ then
        $X_{\text{stream}}(j) = Y_{\text{stream}}(j)$
        if $Y_{\text{stream}}(j) = 1$ then
            $m = m - 1$
        end if
    else
        $X_{\text{stream}}(j) = 0$
    end if
end for
return $X_{\text{stream}}, Y_{\text{stream}}$
```

Algorithm 2: CORDIV [7].

```
Input: $X_{\text{stream}}$ and $Y_{\text{stream}}$
Output: Quotient $Z_{\text{CORDIV}}$
Initialize $Z$ with logic-0s
for $k = 1$ to $N$ do
    $Z(k+1) =$
    $(\neg Y_{\text{stream}}(k) \wedge Z(k)) \vee (Y_{\text{stream}}(k) \wedge X_{\text{stream}}(k))$
end for
return $Z_{\text{CORDIV}} = Z$
```

Algorithm 1, as its foundations for correlated bit-stream generation discussed in [21] (by coupling the generator with FSM-based random source), outlines the steps for suitable correlated bit-stream generation. Once the to-be-divided values X and Y are converted into bit-streams, they are used as inputs to the CORDIV design. The Y bit-stream is initially generated by using any LD sequence (or LFSR). Concurrently, the second bit-stream is generated with proper correlation using the down counter. The proposed correlator circuit receives the data in binary format. The Y bit-stream is produced directly by a bit-stream generator, while the X bit-stream duplicates the Y bit-stream. This procedure goes on until the down counter, starting from x, counts down at each clock cycle to reach zero. This activates the zero port that forces the remaining bits to zero. The enable signal of the down counter is controlled by the Y bit-stream. Here, at the cost of a down counter and an

TABLE II
MAE(%) Comparison of 8-Bit Precision SC Division Operation With the Conventional Correlated Bit-Stream Generators

Seq.	MMDIV Conv.					SSDIV	CORDIV
Gen.	DE=0	DE=1	DE=2	DE=3	DE=4	Conv.	Conv.
Sobol-1	2.79 *(8.50)*	6.12 *(24.61)*	6.92 *(41.67)*	7.50 *(33.62)*	4.11 *(37.73)*	2.79 *(8.50)*	2.68 *(8.50)*
Sobol-2	1.58 *(8.30)*	5.25 *(20.25)*	3.53 *(17.55)*	2.18 *(19.65)*	3.42 *(16.66)*	1.58 *(8.30)*	1.67 *(9.10)*
Sobol-3	1.79 *(8.70)*	6.19 *(16.75)*	3.87 *(16.65)*	2.79 *(13.91)*	1.96 *(13.64)*	1.79 *(8.70)*	1.73 *(9.10)*
Sobol-4	1.58 *(7.10)*	4.88 *(15.25)*	2.87 *(13.65)*	2.24 *(11.74)*	2.96 *(15.27)*	1.58 *(7.10)*	1.54 *(7.50)*
Sobol-5	1.60 *(8.85)*	5.23 *(23.25)*	4.10 *(21.25)*	2.85 *(19.76)*	3.31 *(23.72)*	1.60 *(8.85)*	1.57 *(9.30)*
VDC-2	2.79 *(8.50)*	6.12 *(24.61)*	6.92 *(41.67)*	7.50 *(33.62)*	4.11 *(37.73)*	2.79 *(8.50)*	2.68 *(8.50)*
VDC-4	6.80 *(24.75)*	10.25 *(40.50)*	10.77 *(41.67)*	7.67 *(42.58)*	6.74 *(40.10)*	6.80 *(24.75)*	6.83 *(24.75)*
VDC-8	10.85 *(42.50)*	16.66 *(54.50)*	11.91 *(50.07)*	12.13 *(50.21)*	12.74 *(52.71)*	10.85 *(42.50)*	10.88 *(42.50)*
VDC-16	13.78 *(56.50)*	17.93 *(62.11)*	12.32 *(55.50)*	14.50 *(60.74)*	13.53 *(59.89)*	13.78 *(56.50)*	13.80 *(56.50)*
LFSR	8.32 *(35.12)*	10.38 *(35.50)*	5.04 *(20.02)*	4.67 *(19.93)*	4.69 *(24.12)*	8.32 *(35.12)*	8.50 *(50.50)*

TABLE III
MAE(%) Comparison of 8-Bit Precision SC Division Operation With the Proposed Correlated Bit-Stream Generators

Seq.	MMDIV Prop.					SSDIV	CORDIV
Gen.	DE=0	DE=1	DE=2	DE=3	DE=4	Prop.	Prop.
Sobol-1	0.33 *(6.24)*	0.33 *(6.24)*	0.50 *(8.38)*	0.59 *(8.40)*	0.82 *(12.52)*	0.33 *(6.24)*	0.32 *(6.24)*
Sobol-2	0.89 *(19.93)*	0.89 *(19.93)*	1.19 *(19.99)*	1.36 *(20.02)*	1.58 *(20.03)*	0.89 *(19.93)*	0.89 *(19.93)*
Sobol-3	0.89 *(16.77)*	0.89 *(16.77)*	1.16 *(16.83)*	1.36 *(16.83)*	1.54 *(16.89)*	0.89 *(16.77)*	0.88 *(16.77)*
Sobol-4	0.79 *(13.91)*	0.79 *(13.91)*	1.05 *(14.24)*	1.28 *(14.32)*	1.47 *(14.39)*	0.79 *(13.91)*	0.79 *(13.91)*
Sobol-5	0.79 *(13.37)*	0.79 *(13.37)*	1.05 *(13.51)*	1.27 *(13.59)*	1.47 *(13.91)*	0.79 *(13.37)*	0.79 *(13.37)*
VDC-2	0.33 *(6.24)*	0.33 *(6.24)*	0.50 *(8.38)*	0.59 *(8.40)*	0.82 *(12.52)*	0.33 *(6.24)*	0.32 *(6.24)*
VDC-4	0.35 *(6.28)*	0.35 *(6.28)*	0.47 *(6.30)*	0.45 *(6.23)*	0.68 *(6.30)*	0.35 *(6.28)*	0.35 *(6.28)*
VDC-8	0.71 *(8.32)*	0.71 *(8.32)*	0.58 *(8.33)*	0.60 *(8.33)*	0.69 *(8.34)*	0.71 *(8.32)*	0.71 *(8.32)*
VDC-16	1.26 *(14.56)*	1.26 *(14.56)*	0.86 *(12.57)*	0.88 *(12.57)*	0.92 *(12.60)*	1.26 *(14.56)*	1.25 *(14.56)*
LFSR	2.21 *(17.83)*	2.21 *(17.83)*	2.49 *(22.23)*	2.55 *(22.74)*	2.78 *(22.80)*	2.21 *(17.83)*	2.20 *(16.06)*

AND gate, we generated a bit-stream (divisor) correlated with another bit-stream (dividend). After generating bit-streams with the proposed approach, Algorithm 2 performs the division operation according to the MUX structure of CORDIV. This part can be replaced with any division circuit operating with correlated streams.

IV. Design Space Exploration (DSE)

In this section, we conduct a comprehensive accuracy evaluation of the SOTA SC division designs by incorporating two LD sequence generators, Sobol and VDC. We compare the accuracy of these designs with that of the conventional LFSR-based designs. To assess the accuracy of the SC dividers, we utilize the Maximum and Mean Absolute Error (MAE) metrics for all possible 8-bit precision values within the range of $[0, 255]$. Both the dividend and divisor are assigned 2^8-bit SC bit-streams. Table II presents the MAE comparison of the SOTA SC dividers that incorporate shared RNGs for generating correlated bit-streams. The numbers underneath each value (in parentheses) correspond to the maximum error for that specific configuration. We used the first 5 Sobol sequences from MATLAB for the Sobol-based and the maximal length LFSR with polynomial function $x^8 + x^4 + x^3 + x^2 + 1$ for the LFSR-based designs.

We observe that using LFSR with the MMDIV design improves the accuracy when the number of DEs increases. However, using Sobol sequences instead of LFSR yields even better accuracy without the need for any DEs. Hence, this approach reduces the hardware cost by eliminating DEs. For the VDC sequence, we considered bases 2, 4, 8, and 16 during sequence generation [20]. Notably, the base-2 VDC is similar to the first Sobol sequence (further details can be found in Section VI). The accuracy results for all SC division designs were superior when using VDC with bases 2 and 4 compared to the LFSR case.

Table III presents the accuracy results for the proposed method applied to the SOTA SC division designs. We used the same configuration as in Table II. In the LFSR case, increasing the number of DEs in the MMDIV design led to a degradation in accuracy because by adding more DEs, the required *independence* [9] is diminished due to utilizing the proposed correlated SC bit-streams. This property also affects the hardware footprint of the division circuit. Similarly, for Sobol sequences, the accuracy behavior was consistent. Consequently, there is no need to incorporate DEs in the MMDIV design when using the proposed method. Compared to the results in Table II, the proposed method demonstrated significant improvements in the accuracy of the SC division. As can be seen in Tables II and III, the accuracy improvement regarding the errors for the first Sobol sequence is up to 88%. For the LFSR case, the proposed method yielded an accuracy improvement of over two times. As can be seen, compared to the shared RNG method, all the VDC-based designs are superior to the LFSR counterpart with the proposed bit-stream generator. Notably, the proposed method achieves the same level of accuracy for both the CORDIV and SSDIV designs. In addition to SC-based division, the VDC-based bit-stream generator shows better performance compared to other generators in various computing elements, including the SC multiplier and scaled adder [20], [28].

V. SC Image Processing Involving Division

In this section, we evaluate the performance of the proposed approach in an image processing case study. SC gained popularity for the simple execution of complex tasks such as image processing tasks [29]. Fig. 4 provides an overview of important prior SC case studies in the image processing domain and the bit-stream size used in each case. As can be seen, numerous applications, from edge detection to depth sensing, have been

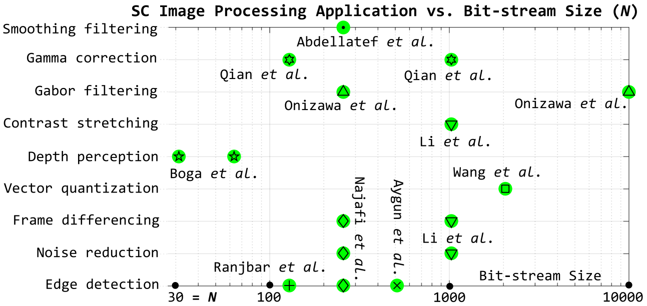
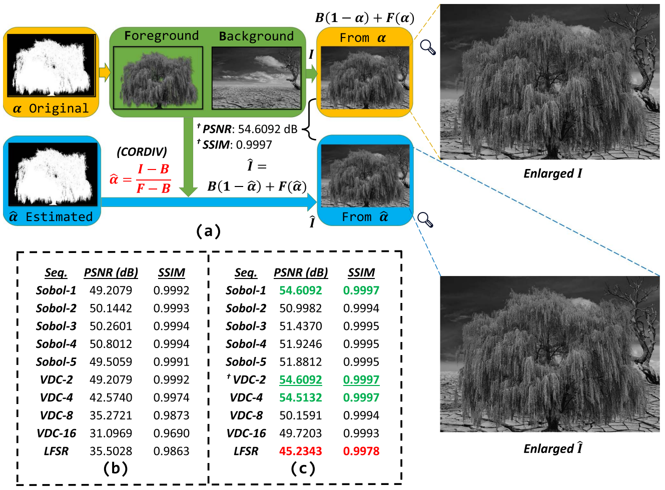


Fig. 4. SC image processing applications [32]. (◇ → [33], ✡ → [29], ⊕ → [34], ▣ → [35], ⊗ → [36], ▽ → [22], △ → [37], [38], ☆ → [39], ⊙ → [40].)

explored. However, despite exploring many applications, there is no definitive solution for separating composite images created through *image blending*. Image blending involves merging *foreground* and *background* images to create a composite image [30]. By reversing the process, *image matting* is a more intricate task that encompasses the extraction of the foreground object from a composite image [31]. Achieving accurate separation requires iterative division operations. Initially, we have three inputs: ⓐ **B** (Background) image, ⓑ **F** (Foreground) image, and ⓒ foreground opacity or matte (the foreground image includes this additional information in the form of an extra image channel, *alpha:* α [41], which contains green screen background information). While opacity ⓒ information provides a value of 0 for entirely background regions, it offers precise values within the range [0,1] specifically for the foreground ⓑ and intersection areas. The image matting is based on *alpha estimation* that is derived from: $I = B \times (1 - \alpha) + F \times \alpha$ (blending formula), where I as an output represents the merged image [42]. This operation involves multiplication, addition, subtraction-from-1 and can be performed in a single iteration; however, estimating the α value from the merged image, along with the background ⓐ and foreground ⓑ information, is a challenging task that requires multiple iterations. The refinement equation can be expressed as $\hat{\alpha} = \frac{I-B}{F-B}$ and is called *alpha estimation* [31]. Alpha estimation requires the following inputs: I (the merged image using the original alpha), B, and F. It is essential to emphasize that alpha estimation involves a division operation. The result of this operation outputs the estimated transparency, $\hat{\alpha}$, particularly affecting the edge pixels of the foreground object, which, in turn, contributes to the overall natural appearance of the image.

He et al. [43] illustrate the potential number of estimated points within the **FB** search space using an example image of size 800 × 600. The estimation process indicates that the number of image pixel couples for estimation can reach up to 10^8 for the given example. The alpha estimation formula ($\hat{\alpha}$) is iteratively employed for each operation. Considering the high cost of employing a binary divider in custom hardware, a low-cost and efficient division circuit can help in the accurate implementation of this task. The $\hat{\alpha}$ estimation transforms into an optimization problem for estimating approximate intersection points, which are not precisely known. Generally, a broader border area, including a third region apart from foreground and background, is defined as the intersection region (*trimap* images [44]), and optimal $\hat{\alpha}$ values are sought for this region. This process requires numerous division operations.



Seq.	PSNR (dB)	SSIM
Sobol-1	49.2079	0.9992
Sobol-2	50.1442	0.9993
Sobol-3	50.2601	0.9994
Sobol-4	50.8012	0.9994
Sobol-5	49.5059	0.9991
VDC-2	49.2079	0.9992
VDC-4	42.5740	0.9974
VDC-8	35.2721	0.9873
VDC-16	31.0969	0.9690
LFSR	35.5028	0.9863

(b)

Seq.	PSNR (dB)	SSIM
Sobol-1	54.6092	0.9997
Sobol-2	50.9982	0.9994
Sobol-3	51.4370	0.9995
Sobol-4	51.9246	0.9995
Sobol-5	51.8812	0.9995
† VDC-2	54.6092	0.9997
VDC-4	54.5132	0.9997
VDC-8	50.1591	0.9994
VDC-16	49.7203	0.9993
LFSR	45.2343	0.9978

(c)

Fig. 5. Image matting performance on alpha estimation using division operation. The proposed bit-stream generator is fed to the CORDIV design. (a) The original alpha- and estimated alpha-related calculations yield blended images to be used as a performance check (I vs. $\hat{I}$). (b) Performance results of different *Sequences* used in the shared RNG method. (c) Performance results of different *Sequences* used in the proposed generator. PSNR: Peak Signal-to-Noise Ratio, SSIM: Structural Similarity. The maximal length LFSR with the polynomial $x^8 + x^4 + x^3 + x^2 + 1$ was used. $N = 2^8$.

TABLE IV
HARDWARE COST OF GENERATING ONE SC BIT-STREAM FOR $\mathrm{N} = 2^8$

Bit-stream Generation	Area (μm^2)	Power (μW)	CPD✦ (ns)	Gen. Time* (ns)	Energy (pJ)
VDC-2, 4, 8, 16	202	6.02	1.03	263.6	1.58
Sobol-2, 3, 4, 5	853	47.3	1.22	312.3	14.7

✦: Critical Path Delay (per bit generation) | *: Generation time (CPD×N)

Fig. 5 illustrates the performance results of alpha estimation for $X = I - B$ and $Y = F - B$ using the proposed bit-stream generator and CORDIV circuit ($\hat{\alpha} = \frac{X}{Y}$). The example in Fig. 5(a) demonstrates the alpha estimation. Subsequently, the obtained $\hat{\alpha}$ reblends the plain F and B images similar to the original alpha (α), and the resulting comparison is made between I and $\hat{I}$. Fig. 5(b) presents the accuracy results of the image matting alpha estimation task with the shared RNG-based design. Fig. 5(c) shows the performance of the same task with the proposed bit-stream generator.

VI. HARDWARE COST COMPARISON

In this section, we conduct a thorough hardware cost evaluation to assess the cost-efficiency of generating correlated bit-streams using both the shared RNG approach and the proposed method. The designs were synthesized using Synopsys Design Compiler v2018.06 using the 45 nm FreePDK gate library. Table IV compares the hardware costs in terms of hardware footprint, power and energy consumption, and critical path delay (CPD) when generating one SC bit-stream (bit-stream Y in Fig. 3) of length $N = 2^8$ with Sobol and VDC sequences. The CPD parameter demonstrates the latency for generating

TABLE V
ISO-ACCURACY HARDWARE COST COMPARISON OF GENERATING TWO LD BIT-STREAMS WITH THE SHARED RNG AND THE PROPOSED METHOD

Bit-stream Generation	Area (μm^2)	Power (μW)	CPD (ns)	# of Clk. Cycles	Gen. Time (ns)	Energy (pJ)
Shared RNG bit-stream generator						
VDC-2, 4, 8, 16	274	8.20	1.03	256	263.6	2.16
Sobol-2, 3, 4, 5	925	49.51	1.22	256	312.3	15.46
Proposed bit-stream generator						
VDC-2	333	9.69	1.03	243	250.2	2.42
VDC-4				220	226.6	2.19
VDC-8				197	202.9	1.96
VDC-16				178	183.3	1.77
Sobol-2, 3	986	50.9	1.22	251	306.2	15.61
Sobol-4, 5				252	307.4	15.67

one SC bit, while the generation time exhibits the total time to generate the whole SC bit-stream. The total hardware cost when generating the two SC bit-streams with the shared RNG method (Fig. 1(a)) and the proposed method (Fig. 3) are reported in Table V. The hardware costs with the proposed bit-stream generator are slightly higher than the shared RNG counterpart due to exploiting a down counter, an AND gate, and a NOT gate to generate the second SC bit-stream (bit-stream X in Fig. 3). This minor overhead is negligible compared to the significant enhancement in the accuracy results achieved by utilizing our proposed bit-stream generation method. For an iso-accuracy comparison, we assessed the number of clock cycles (corresponding to the length of bit-streams) to achieve the same accuracy level with the two approaches. Table V compares energy consumption numbers for both VDC-based and Sobol-based designs. As it can be seen, the VDC-based designs lower the processing time, resulting in lower energy consumption compared to the Sobol-based designs. Finally, we observed that the Sobol-based designs show poor performance when reducing the length of bit-streams, as their accuracy sharply declines when decreasing the length. This was in contrast to the VDC-based bit-streams, which demonstrated greater resilience to reducing the bit-stream length.

The VDC sequence in base-ρ (radix-ρ) can be derived by reversing the integer digits in that base and converting them to a fractional number within the [0,1) interval. Similarly, the first Sobol sequence is constructed by reversing the output bits of a counter. Generating VDC sequences with powers of 2 bases can be accomplished through a straightforward counter implementation, incurring no additional costs [20]. However, for bases other than powers of 2, the hardware design becomes more intricate, leading to higher costs.

By integrating a counter for VDC with bases 2, 4, 8, and 16, the hardware footprint is remarkably small compared to the Sobol cases. In all scenarios, the inclusion of a down counter and an AND gate is necessary to generate the second correlated bit-stream. In terms of hardware efficiency, the VDC design demonstrates an area reduction of 4× and a remarkable energy efficiency improvement of 9× when compared to the Sobol design. Furthermore, the VDC bit-stream generation is 1.18× faster than the Sobol counterpart. Considering the lower accuracy of the LFSR designs, integrating lightweight VDC generators with the proposed bit-stream generator opens up new design possibilities, providing a high accuracy similar to the Sobol sequences.

VII. CONCLUSION

This study provides a novel perspective on stochastic computing (SC)-based division by focusing on bit-stream generation. Prior art offers low-cost and highly efficient solutions using correlated bit-streams; however, this work discusses various options for bit-stream generation. Linear-feedback shift register (LFSR) is the most common random number generator (RNG) used in the SC literature. The state-of-the-art designs, however, use finite-state machine (FSM)-based low-discrepancy (LD) sequence generators. In this study, we explored an efficient correlated bit-stream generation method that employs different LD sequences for high accuracy. Particularly, we showed that counter-based generators such as Van der Corput (VDC) can provide high hardware efficiency compared to prior LFSR and FSM-based Sobol generators. We employed the proposed approach for implementing the division operations in the image matting problem, which is implemented for the first time in the SC literature. It is important to mention that even though the proposed design needs slightly more hardware, the tradeoff in area-delay is favorable. Considering hardware efficiency, the versatile, low-cost, and highly efficient generator offers a solution for all SC dividers requiring correlated bit-streams.

REFERENCES

[1] B. R. Gaines, "Stochastic Computing Systems," in *Advances in Information Systems Science*. Boston, MA, USA: Springer, 1969, pp. 37–172.

[2] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1515–1531, Aug. 2018.

[3] V. T. Lee, A. Alaghi, and L. Ceze, "Correlation manipulating circuits for stochastic computing," in *Proc. Des., Automat. Test Eur. Conf. Exhib.*, 2018, pp. 1417–1422.

[4] M. H. Najafi, D. J. Lilja, M. D. Riedel, and K. Bazargan, "Low-cost sorting network circuits using unary processing," *IEEE Trans. Very Large Scale Integr.*, vol. 26, no. 8, pp. 1471–1480, Aug. 2018.

[5] X. Jiao, "An improved stochastic decoding algorithm of LTE turbo codes," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2012, pp. 301–308.

[6] S. S. Tehrani, S. Mannor, and W. J. Gross, "Survey of stochastic computation on factor graphs," in *Proc. 37th Int. Symp. Mult.-Valued Log.*, 2007, pp. 54–54.

[7] T.-H. Chen and J. P. Hayes, "Design of division circuits for stochastic computing," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2016, pp. 116–121.

[8] S.-I. Chu, "New divider design for stochastic computing," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 67, no. 1, pp. 147–151, Jan. 2020.

[9] S. Wang et al., "Highly accurate division and square root circuits by exploiting signal correlation in stochastic computing," *Int. J. Circuit Theory Appl.*, vol. 50, pp. 1375–1385, 2022.

[10] J. Wang and M. F. Cohen, "Image and video matting: A survey," *Found Trends Comput. Graph. Vis.*, vol. 3, pp. 97–175, 2007.

[11] Y. Zhou, G. Xie, J. Han, and Y. Zhang, "Absolute subtraction and division circuits using uncorrelated random bitstreams in stochastic computing," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures*, 2021, pp. 1–6.

[12] J. H. Anderson, Y. Hara-Azumi, and S. Yamashita, "Effect of LFSR seeding, scrambling and feedback polynomial on stochastic computing accuracy," in *Proc. Des., Automat., Test Europe Conf., Exhib.*, 2016, pp. 1550–1555.

[13] S. Asadi, M. H. Najafi, and M. Imani, "A low-cost FSM-based bit-stream generator for low-discrepancy stochastic computing," in *Proc. Des., Automat., Test Europe Conf., Exhib.*, 2021, pp. 908–913.

[14] I. M. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *USSR Comp. Math. Math. Phys.*, vol. 7, no. 4, pp. 86–112, 1967.

[15] J. V. d. Corput, "Verteilungsfunktionen I," in *Proc. Kon. Ned. Akad. v. Wetensch*, 1935, pp. 813–821.

[16] M. H. Najafi, D. J. Lilja, and M. Riedel, "Deterministic methods for stochastic computing using low-discrepancy sequences," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, 2018, pp. 1–8.

[17] S. Liu and J. Han, "Energy efficient stochastic computing with sobol sequences," in *Proc. Des., Automat., Test Eur. Conf., Exhibit.*, 2017, pp. 650–653.

[18] S. Liu and J. Han, "Toward energy-efficient stochastic circuits using parallel sobol sequences," *IEEE Trans. Very Large Scale Integr.*, vol. 26, no. 7, pp. 1326–1339, Jul. 2018.

[19] M. H. Najafi, D. Jenson, D. J. Lilja, and M. D. Riedel, "Performing stochastic computation deterministically," *IEEE Trans. Very Large Scale Integr.*, vol. 27, no. 12, pp. 2925–2938, Dec. 2019.

[20] M. S. Moghadam et al., "P2LSG: Powers-of-2 low-discrepancy sequence generator for stochastic computing," in *Proc. Asia South Pacific Des. Automat. Conf.*, 2024, pp. 1–8.

[21] S. Asadi, M. H. Najafi, and M. Imani, "CORLD: In-stream correlation manipulation for low-discrepancy stochastic computing," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Des.*, 2021, pp. 1–9.

[22] 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. Very Large Scale Integr.*, vol. 22, no. 3, pp. 449–462, Mar. 2014.

[23] Z. Li et al., "HEIF: Highly efficient stochastic computing-based inference framework for deep neural networks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst*, vol. 38, no. 8, pp. 1543–1556, Aug. 2019.

[24] A. Alaghi and J. P. Hayes, "Exploiting correlation in stochastic circuit design," in *Proc. IEEE 31st Int. Conf. Comput. Des.*, Asheville, NC, USA, 2013, pp. 39–46.

[25] T. J. Baker and J. P. Hayes, "CeMux: Maximizing the accuracy of stochastic MUX adders and an application to filter design," *ACM Trans. Des. Automat. Electron. Syst.*, vol. 27, pp. 1–26, 2022.

[26] R. Ananth, "Programmable supervisory circuit and applications thereof," U.S. Patent 6618711, Sep. 2003.

[27] M. Faix et al., "Cognitive computation: An exact Bayesian inference stochastic machine," *Int. J. Softw. Sci. Comput. Intell.*, vol. 9, no. 3, pp. 37–58, 2017.

[28] M. S. Moghadam et al., "Accurate and energy-efficient stochastic computing with van der Corput sequences," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures*, 2023, pp. 1–6.

[29] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Comput.*, vol. 60, no. 1, pp. 93–105, Jan. 2011.

[30] R. Szeliski, *Computer Vision Algorithms and Applications. Texts in Computer Science*. London, U.K.: Springer, 2011.

[31] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.

[32] S. Aygun, M. H. Najafi, M. Imani, and E. O. Gunes, "Agile simulation of stochastic computing image processing with contingency tables," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 10, pp. 3474–3478, Oct. 2023.

[33] M. H. Najafi and D. J. Lilja, "High-speed stochastic circuits using synchronous analog pulses," in *Proc. 22nd Asia South Pacific Des. Automat. Conf.*, 2017, pp. 481–487.

[34] M. Ranjbar, M. E. Salehi, and M. H. Najafi, "Using stochastic architectures for edge detection algorithms," in *Proc. 23rd Iranian Conf. Elect. Eng.*, 2015, pp. 723–728.

[35] R. Wang, J. Han, B. F. Cockburn, and D. G. Elliott, "Stochastic circuit design and performance evaluation of vector quantization for different error measures," *IEEE Trans. Very Large Scale Integr.*, vol. 24, no. 10, pp. 3169–3183, 2016.

[36] S. Aygün, M. Altun, and E. O. Güneş, "Sobel filter operation in image processing via stochastic arithmetic-logic unit design," in *Proc. 25th Signal Process. Commun. Appl. Conf.*, 2017, pp. 1–4.

[37] 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, Sep. 2015.

[38] N. Onizawa, D. Katagiri, K. Matsumiya, W. J. Gross, and T. Hanyu, "An accuracy/energy-flexible configurable Gabor-filter chip based on stochastic computation with dynamic voltage–frequency–length scaling," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 3, pp. 444–453, Sep. 2018.

[39] K. Boga, N. Onizawa, F. Leduc-Primeau, K. Matsumiya, T. Hanyu, and W. J. Gross, "Stochastic implementation of the disparity energy model for depth perception," in *Proc. IEEE Workshop Signal Process. Syst.*, 2015, pp. 1–6.

[40] H. Abdellatef et al., "Accurate and compact stochastic computations by exploiting correlation," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 27, pp. 547–564, 2019.

[41] S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong, "Soft edge smoothness prior for alpha channel super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[42] J. Wang and M. F. Cohen, "Optimized color sampling for robust matting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[43] K. He, C. Rhemann, C. Rother, X. Tang, and J. Sun, "A global sampling method for alpha matting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2049–2056.

[44] D. Cho, S. Kim, Y. -W. Tai, and I. S. Kweon, "Automatic trimap generation and consistent matting for light-field images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1504–1517, Aug. 2017.

Mehran Shoushtari Moghadam (Graduate Student Member, IEEE) received the B.Sc. degree in computer engineering- hardware and the M.Sc. degree in computer engineering- computer systems architecture from the University of Isfahan, Isfahan, Iran, in 2010 and 2016, respectively. He is currently working toward the Ph.D. degree with the School of Computing and Informatics, Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA, USA. He distinguished as one of the top-ranking students during both his B.Sc. and M.Sc. studies. He has more than ten years of experience as a computer hardware and network specialist in the industry. His research interests involve emerging and unconventional computing paradigms, including energy-efficient stochastic computing, real-time and highly-accurate hyperdimensional computing, and hardware security.

Sercan Aygun (Member, IEEE) received the B.Sc. degree in electrical & electronics engineering and a double major in computer engineering from Eskisehir Osmangazi University, Türkiye, in 2013, the M.Sc. degree in electronics engineering from Istanbul Technical University, Istanbul, Türkiye, in 2015, the second M.Sc. degree in computer engineering from Anadolu University, Eskisehir, Türkiye, in 2016, and the Ph.D. degree in electronics engineering from Istanbul Technical University, in 2022. He is currently a Postdoctoral Researcher with the University of Louisiana at Lafayette, Lafayette, LA, USA. His Ph.D. work has appeared in several Ph.D. Forums of top-tier conferences, such as DAC, DATE, ASP-DAC, and ESWEEK. He works on emerging computing technologies, including stochastic computing in computer vision and machine learning. He was the recipient of the Best Scientific Research Award of the ACM SIGBED Student Research Competition (SRC) ESWEEK 2022 and the Best Paper Award at GLSVLSI'23. Dr. Aygun's Ph.D. work was recognized with the Best Scientific Application Ph.D. Award by the Turkish Electronic Manufacturers Association.

Sina Asadi received the B.Sc. degree in computer engineering from the Amirkabir University of Technology, Tehran, Iran, in 2014, the M.Sc. degree in computer engineering from the Sharif University of Technology, Tehran, Iran, in 2016, and the Ph.D. degree in computer engineering from the University of Louisiana at Lafayette, Lafayette, LA, USA, in 2023. He is currently a Faculty Member of the Electrical and Computer Engineering Department, College of Engineering, University of Louisiana. His research interests include approximate and stochastic computing, fault-tolerant system design, machine learning and intelligent systems, computer architecture, and electronic circuits. He was selected as a DAC Young Fellow in DAC 2020 and the Young Programs Award at the DATE 2021.

M. Hassan Najafi (Senior Member, IEEE) received the B.Sc. degree in computer engineering from the University of Isfahan, Isfahan, Iran, the M.Sc. degree in computer architecture from the University of Tehran, Tehran, Iran, and the Ph.D. degree in electrical engineering from the University of Minnesota, Twin Cities, MN, USA, in 2011, 2014, and 2018, respectively. He is currently an Assistant Professor with the School of Computing and Informatics, University of Louisiana, Lafayette, LA, USA. He has authored/co-authored more than 75 peer-reviewed papers and has been granted five U.S. patents with more pending. His research interests include stochastic and approximate computing, unary processing, in-memory computing, and hyperdimensional computing. In recognition of his research, he was the recipient of the 2018 EDAA Outstanding Dissertation Award, Doctoral Dissertation Fellowship from the University of Minnesota, and Best Paper Award at the ICCD'17 and GLSVLSI'23. Dr. Najafi has been an Editor of IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS.