iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** VI **Month of publication:** June 2024

DOI: https://doi.org/10.22214/ijraset.2024.63345

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue VI June 2024- Available at www.ijraset.com

Simultaneous Area and Latency Optimisation for Stochastic Circuit by D Flip-Flop Insertion

K. Deepa Rao[1], K. Sri Harsha[2], V. Swathi[3], K. Sai Kiran[4], D. Divakar[5]
[1]*Assistant Professor, Department of ECE, CMR Engineering College, Hyderabad, Telangana, India*
[2, 3, 4, 5]*Department of ECE, CMR College of Engineering & Technology, Hyderabad, Telangana, India,*

Abstract: ***Stochastic circuits, leveraging the inherent randomness in computation, have garnered substantial attention for their potential in energy-efficient computing across various domains, including machine learning, signal processing, and cryptography. However, harnessing the full potential of stochastic circuits necessitates addressing the intertwined challenges of area and latency optimization, which are crucial metrics for the practical deployment of such circuits in real-world applications. This paper presents a comprehensive exploration of simultaneous area and latency optimization for stochastic circuits, focusing on the strategic insertion of D flip-flops.***

The proposed approach capitalizes on the unique characteristics of stochastic circuits, where computation is inherently probabilistic, to devise a systematic method for identifying critical paths and judiciously inserting D flip-flops to enhance both area efficiency and latency performance. By leveraging the stochastic behavior inherent in these circuits, our methodology identifies opportunities for optimizing the circuit's structure to minimize critical path delays and reduce area overhead simultaneously.

Keywords: stochastic computing; stochastic number generator; decorrelation; D flip-flop insertion;

I. INTRODUCTION

In the pursuit of efficient and high-performance computing, stochastic circuits have emerged as a promising avenue, offering unique advantages in energy efficiency and fault tolerance. These circuits, which harness randomness to perform computations, hold immense potential across various domains, including machine learning, signal processing, and cryptography. However, optimizing stochastic circuits for both area and latency remains a significant challenge, given their inherently probabilistic nature.

The optimization of stochastic circuits involves a delicate balancing act between minimizing the physical footprint (area) of the circuit and reducing the time required to perform computations (latency). Traditionally, these optimization objectives have been addressed independently, leading to suboptimal solutions that compromise either area efficiency or latency performance. Consequently, there is a growing demand for methodologies that can simultaneously optimize both area and latency in stochastic circuits, enabling their practical deployment in real-world applications.

In this context, this paper introduces a novel approach for simultaneous area and latency optimization in stochastic circuits through the strategic insertion of D flip-flops. D flip-flops, widely used in digital circuit design, offer a means to introduce synchronization and control the timing of signals within the circuit. Leveraging the inherent stochasticity of these circuits, our methodology identifies critical paths and strategically inserts D flip-flops to mitigate delays and minimize area overhead.

The integration of D flip-flops into stochastic circuits presents a unique opportunity to exploit the interplay between randomness and synchronization, enabling the optimization of both area and latency metrics. By strategically placing flip-flops along critical paths, we can mitigate the effects of probabilistic delays and improve the overall performance of the circuit without significantly increasing its physical footprint. This approach represents a departure from conventional optimization techniques, which often focus solely on either area or latency, neglecting the potential synergies between the two objectives. Through extensive experimentation and benchmarking against state-of-the-art techniques, we demonstrate the effectiveness of our approach in achieving significant improvements in both area and latency metrics. By simultaneously addressing these optimization objectives, our methodology offers a holistic solution for enhancing the efficiency and performance of stochastic circuits, thereby expanding their applicability in a wide range of computational tasks. In summary, this paper presents a novel methodology for simultaneous area and latency optimization in stochastic circuits through D flip-flop insertion. By leveraging the unique characteristics of stochastic circuits and the versatility of D flip-flops, we demonstrate a practical approach to achieving optimal performance while minimizing resource utilization. This work contributes to advancing the state-of-the-art in stochastic circuit design and lays the groundwork for their widespread adoption in energy-efficient and high-performance computing applications.

II. LITERATURE REVIEW

Ref [1]involves creating a specialized integrated circuit (IC) chip designed to be implanted onto a damaged retina. The purpose of this chip is to either directly sense incoming visual information or process images wirelessly transmitted from an external camera.Once this visual information is captured, the IC chip converts it into electrical signals that mimic the neural-style signals normally processed by a healthy retina. These signals are then used to stimulate the remaining healthy retinal cells, generating useful visual sensations for the patient.One critical aspect of this IC design is its power efficiency. It must operate within very strict power limits to prevent excessive heat generation, which could potentially damage the delicate tissues of the eye.Ref [2] The primary challenge associated with using stochastic computing (SC) is the requirement for long bit-streams, which in turn necessitates extended run-times to effectively manage the inherent random fluctuations within the bit-stream patterns. This extended duration is essential to achieve sufficient levels of precision and accuracy in computation.Precision in the context of stochastic numbers (SN) with a length n,n can be quantified as log $2n$ bits. This means that to increase precision by 1 bit, the length of the stochastic number's bit-stream must be doubled. For instance, if you want to enhance precision from k bits to k+1 the length of the bit-stream needs to grow exponentially.Accuracy, on the other hand, refers to how closely the computed stochastic number approximates a desired target value$T$. This closeness is typically expressed in terms of acceptable error bounds. Achieving higher accuracy often requires even longer bit-streams than those needed for precision improvements.

[3]introduce a general combinational design for SC, which only makes an assumption on the necessary input probabilities without any additional assumption on the underlying combinational circuit.[4] to quantify the impact of correlation on the accuracy of stochastic circuits, and to evaluate the major known methods of reducing correlation or, equivalently, maintaining accuracy over multiple computational steps.

III. PROPOSED MODEL

Stochastic computing (SC) has gained attention recently due to its attractive features, such as compact arithmetic units and robustness against bit flip errors. SC operates on stochastic bit streams (SBSs), which are encoded representations of real values.There are two main encoding formats for SBSs: unipolar encoding and bipolar encoding. In unipolar encoding, the real value is represented by the ratio of 1s in the stream. For instance, if a stream has half of its bits set to 1, it encodes the value of 1/2. Unipolar encoding can only represent non-negative values within the interval [0, 1].

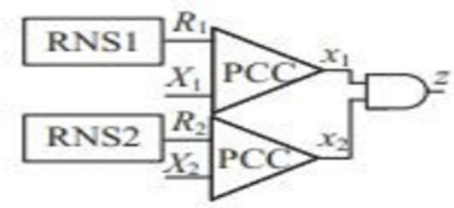
On the other hand, bipolar encoding allows for the representation of values in the interval [-1, 1]. Here, the ratio of 1s in the SBS is used to encode the real value as (2r - 1), where r is the ratio of 1s in the stream.Although the techniques discussed in the paper are applicable to both encoding formats, the discussion primarily focuses on unStochastic computing is a method that relies on probabilities represented by binary digits (0s and 1s). However, performing arithmetic operations directly with these binary digits can be cumbersome. One way to overcome this challenge is by implementing scaled operations to better meet the demands of arithmetic.

In stochastic computing, a bit-stream (a sequence of binary digits) is used to represent probabilities. For example, if a bit-stream contains 75% 1s and 25% 0s, we denote it as p = 0.75, indicating the probability of encountering a 1 at any arbitrary bit position.Multiplication in stochastic computing can be achieved using an AND gate circuit, where two binary bit streams (p1 and p2) act as input variables. The output of this circuit represents the product of the chances (S1 × S2), under the assumption that the two input streams are unrelated to each other.

For instance, let's consider two input bit streams: S1 = 4/8 and S2 = 6/8. When these are multiplied using the AND gate, the resulting output (S3) would be 3/8, as illustrated in Figure 01(a). Similarly, in Figure 01(b), the same multiplication is depicted, but the output (S3) is 2/8. This output is considered an approximation to the precise output represented in Figure 01(a), as the circuits involved convert binary forms to stochastic numbers and vice versa using a stochastic multiplier.

However, the stochastic representation of a given number is not unique, leading to challenges in accuracy and reliability. To overcome these issues, stochastic computing uses a redundant number representation system. In this system, a stochastic number may be represented as a binary sequence of length n with n1 ('1's) and 1-n1 ('0's).

Stochastic computing faces limitations such as inaccuracies due to correlations during number combinations, errors arising during number alterations, and random fluctuations during representation. To reduce inaccuracies, stochastic number generators are employed. These generators efficiently transform random and unrelated numbers by incorporating Linear Feedback Shift Registers (LFSRs). LFSRs consist of m flip-flops, and each cycle passes through a distinct state (n=2^m-1), excluding all-zero states. The generated binary sequence is deterministic, pseudorandom, and has an equal number proportion of zeros and ones.



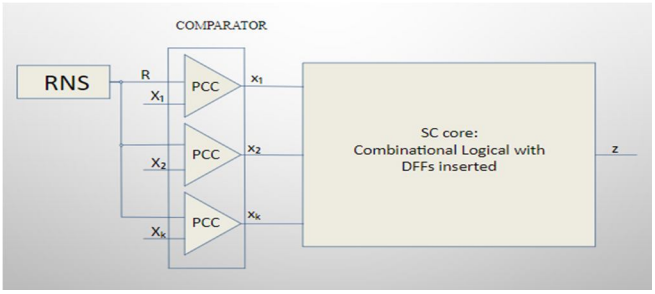
A. Optimization of SNG's Using D Flip Flop

Stochastic computing finds widespread applications across various domains such as reliability analysis, polynomial arithmetic, basic arithmetic calculations, analog and hybrid computational transformations, neural networks, control systems, and image processing. However, despite its versatility, stochastic computing encounters challenges and limitations, including high hardware costs and excessive power consumption, leading to prolonged periods of computational dormancy.

In Figure 2(a), the architecture of a stochastic number generator is depicted, comprising a random number source and a comparator. The primary function of the random number generator is to produce a random uniformly distributed binary number (R) at each clock cycle. The Linear Feedback Shift Register (LFSR) and comparator (CMP) work sequentially, with the CMP comparing the binary number (R) with a constant binary number (X). If R is less than X, the output is represented as "1"; otherwise, it's "0".

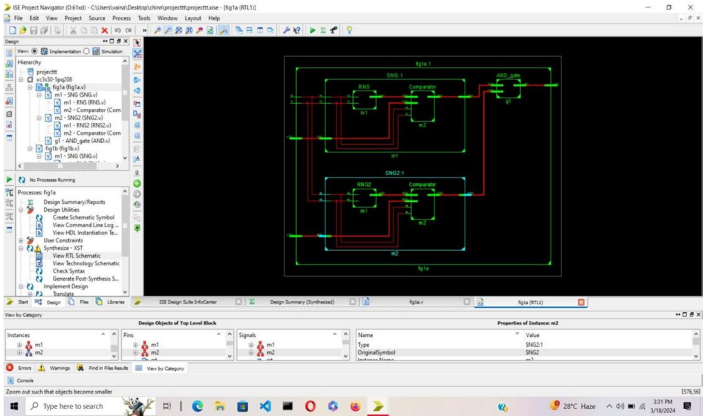
The comparator generates stochastic bit streams based on permutations and combinations equal to the binary number (X1). As stochastic computing relies on all input bit streams to execute efficiently, it's essential that the number of number generators matches the quantity of inputs in the circuit. However, these factors contribute to a larger circuit architecture, resulting in increased power consumption and reduced efficiency of the computing core.

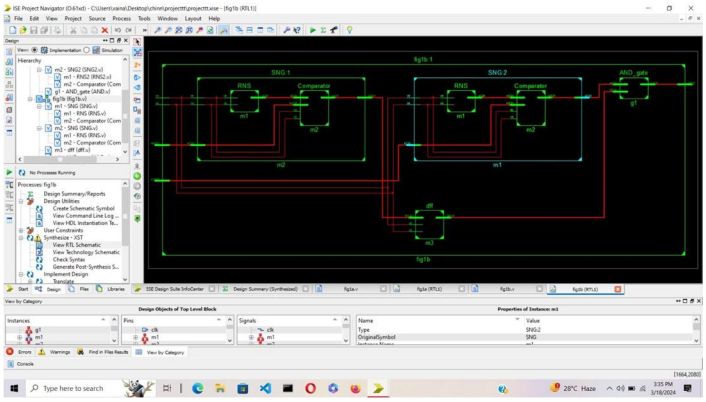
To address these issues, a redesign of the architecture was implemented, utilizing D flip-flops (DFFs) to minimize circuit delay, computational latency, and bit stream errors. An effective strategy to reduce hardware costs involves optimizing the sharing of the random number source with all stochastic number generators (SNGs) in the architecture. This optimization helps streamline the architecture, making it more cost-efficient and power-effective.

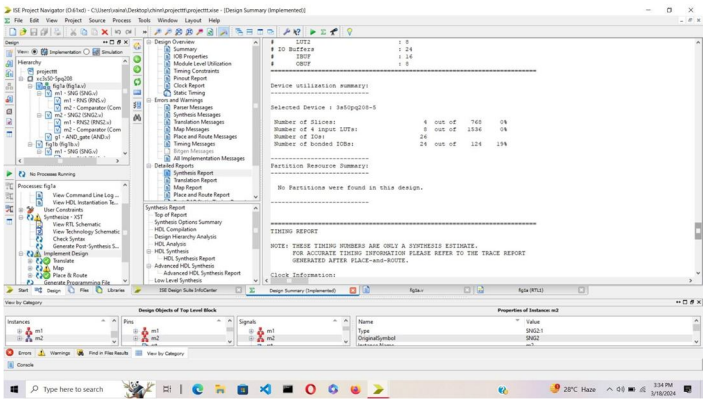


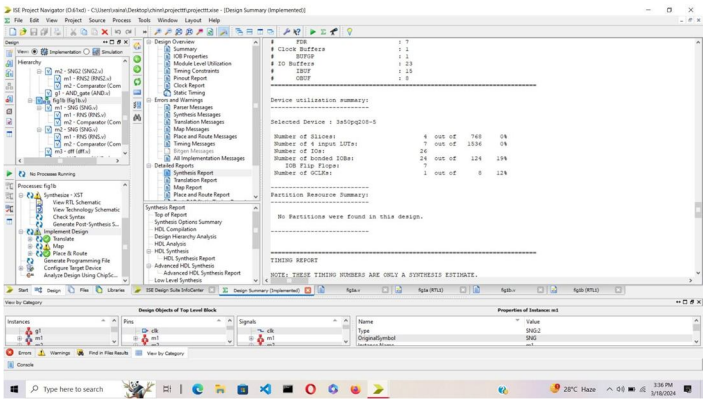
IV. RESULTS & ANALYSIS

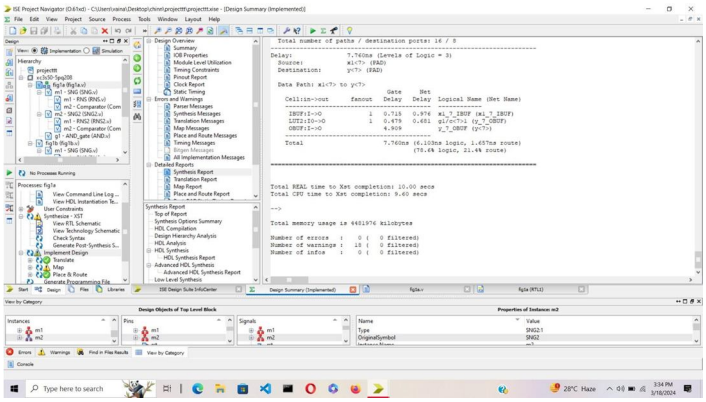
Experimental study to assess the impact of inserting D Flip-Flops (DFFs) into stochastic circuits. They began by examining how DFF insertion affected the power consumption and area of a stochastic multiplier circuit. Two versions of the stochastic multiplier were synthesized using a specific design tool and cell library.











The results showed that SNGs constituted a significant portion of power and area in the stochastic circuit. By simplifying these SNGs through DFF insertion, substantial power (29.3%) and area (28.5%) savings were achieved. Additionally, they observed that while a single DFF contributed relatively small power (4.3%) and area (3.9%) to the overall circuit, the cumulative impact across multiple DFFs could still be significant.

However, DFF insertion affects the computational accuracy of stochastic circuits due to inherent randomness. The output precision of a stochastic circuit was quantified using the root mean square error (RMSE) of the output random variable. Empirical studies compared the RMSE of circuits with DFF insertion against those using separate SNGs. They randomly generated circuits with varying complexity and input probabilities to assess the impact of DFFs on accuracy.

Overall, the study demonstrated that while DFF insertion can lead to substantial power and area savings, it also alters the computational accuracy of stochastic circuits, necessitating a balance between efficiency and precision in circuit design.

V. CONCLUSION

In this paper, the focus was on optimizing stochastic circuits by strategically inserting D Flip-Flops (DFFs) to reduce hardware costs, specifically targeting the Stochastic Number Generators (SNGs) within the circuits. The aim was to overall decrease the expense of implementing stochastic circuits while maintaining their original functionality. Particularly, when the Random Number Source (RNS) was implemented as a Linear Feedback Shift Register (LFSR), the authors introduced a technique to embed DFFs within the SNGs without altering their primary operations.

This extension allowed for the development of a method aimed at minimizing the number of DFFs required within a stochastic circuit. Additionally, the authors proposed an approach to reduce the circuit delay associated with stochastic computing. The combined effect of these optimizations addressed critical challenges inherent in stochastic computing, namely the extended computation latency and the high resource consumption for generating stochastic bit streams.

The experimental results presented in the paper showcased the effectiveness of their proposed optimization techniques when compared to existing state-of-the-art methods involving DFF insertion in stochastic circuits. The authors demonstrated that their approach not only streamlined hardware costs but also enhanced circuit performance by reducing computation delays.

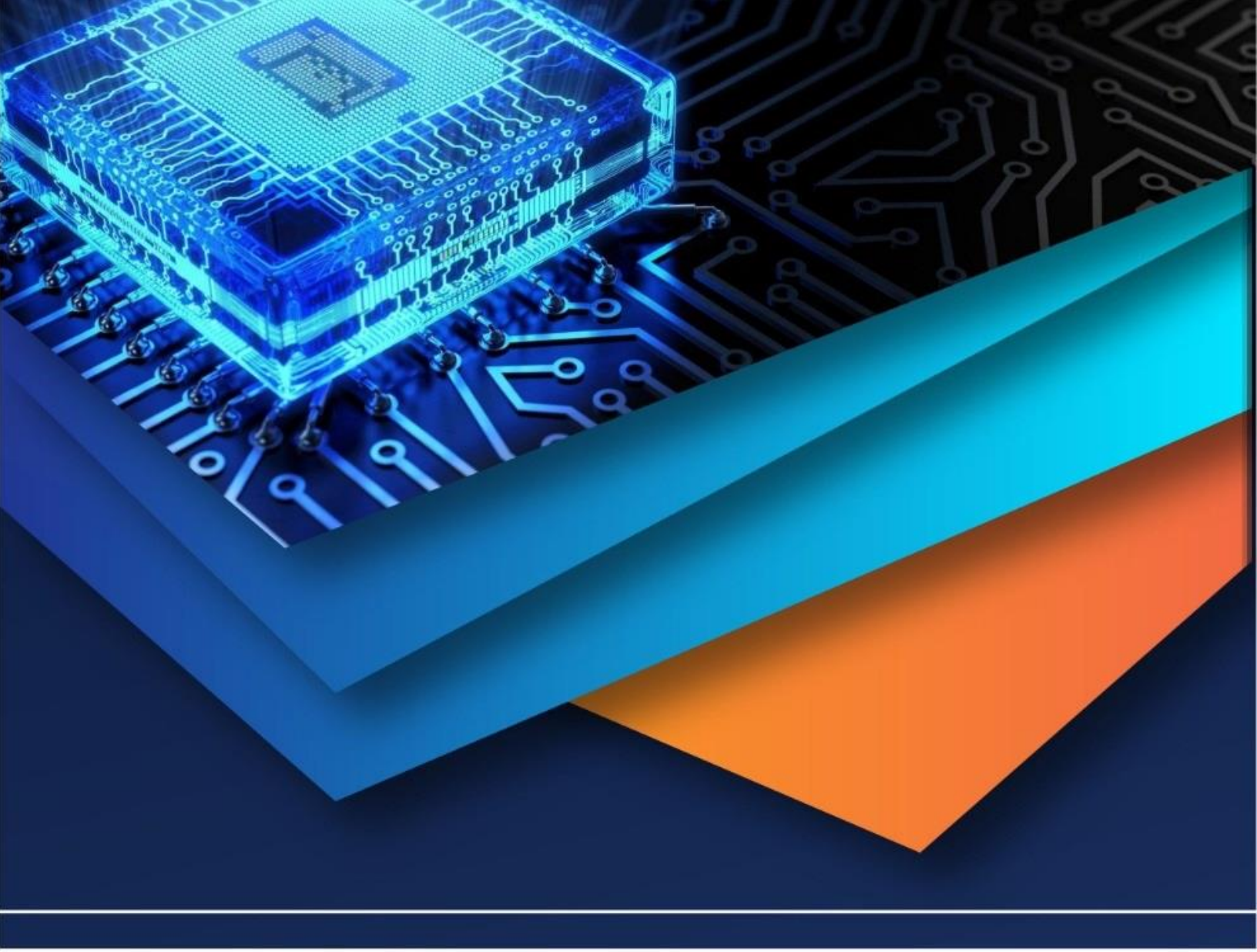
Moreover, the paper introduced a systematic approach for analyzing the stochastic function of circuits incorporating DFFs. This analytical method provided valuable insights into the behavior and characteristics of such stochastic circuits. For instance, it facilitated the establishment of functional equivalence constraints crucial for implementing the proposed optimization strategies effectively.

Overall, the study presents a comprehensive framework for optimizing stochastic circuits through DFF insertion, emphasizing the importance of thoughtful design techniques that balance cost-effectiveness with maintaining circuit functionality and performance. The proposed methodologies and analytical tools offer a valuable contribution to advancing the efficiency and practicality of stochastic computing technologies.

REFERENCES

[1] B. R. Gaines, “Stochastic computing systems,” in Advances in information systems science. Springer, 1969, pp. 37–172.
[2] B. D. Brown and H. C. Card, “Stochastic neural computation I: Computational elements,” IEEE Transactions on Computers, vol. 50, no. 9, pp. 891–905, 2001.
[3] W. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, “An architecture for fault-tolerant computation with stochastic logic,” IEEE Transactions on Computers, vol. 60, no. 1, pp. 93–105, 2011.
[4] H. Ichihara, T. Sugino, S. Ishii, T. Iwagaki, and T. Inoue, “Compact and accurate digital filters based on stochastic computing,” accepted by IEEE Transactions on Emerging Topics in Computing, 2016.
[5] Y. Liu and K. K. Parhi, “Architectures for recursive digital filters using stochastic computing,” IEEE Transactions on Signal Processing, vol. 64, no. 14, pp. 3705–3718, 2016.
[6] A. Alaghi, C. Li, and J. P. Hayes, “Stochastic circuits for realtime image-processing applications,” in Design Automation Conference, 2013, pp. 136:1–136:6.
[7] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, “Computation on stochastic bit streams digital image processing case studies,” IEEE Transactions on Very Large Scale Integration Systems, vol. 22, no. 3, pp. 449–462, 2014.
[8] B. Li, M. H. Najafi, and D. J. Lilja, “Using stochastic computing to reduce the hardware requirements for a restricted Boltzmann machine classifier,” in International Symposium on FPGA, 2016, pp. 36–41.
[9] K. Kim, J. Kim, J. Yu, J. Seo, J. Lee, and K. Choi, “Dynamic energyaccuracy trade-off using stochastic computing in deep neural networks,” in Design Automation Conferernce, 2016, pp. 124:1–124:6.
[10] S. S. Tehrani, S. Mannor, and W. J. Gross, “Fully parallel stochastic LDPC decoders,” IEEE Transactions on Signal Processing, vol. 56, no. 11, pp. 5692–5703, 2008.
[11] S. S. Tehrani, A. Naderi, G.-A. Kamendje, S. Hemati, S. Mannor, and W. J. Gross, “Majority-based tracking forecast memories for stochastic LDPC decoding,” IEEE Transactions on Signal Processing, vol. 58, no. 9, pp. 4883–4896, 2010.

[12] A. Alaghi, W. Qian, and J. P. Hayes, "The promise and challenge of stochastic computing," accepted by IEEE Transactions on ComputerAided Design of Integrated Circuits and Systems, 2018.
[13] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," ACM Transactions on Embedded Computing Systems, vol. 12, no. 2s, p. 92, 2013.
[14] S. W. Golomb, Shift Register Sequences, Revised Ed. Aegean Park Press, 1981.
[15] P.-S. Ting and J. P. Hayes, "Isolation-based decorrelation of stochastic circuits," in International Conference on Computer Design, 2016, pp. 88–95.
[16] A. Alaghi and J. P. Hayes, "Fast and accurate computation using stochastic circuits," in Design, Automation & Test in Europe, 2014, pp. 76:1–76:4.
[17] S. Liu and J. Han, "Energy efficient stochastic computing with Sobol sequences," in Design, Automation & Test in Europe, 2017, pp. 650–653.
[18] P. K. Gupta and R. Kumaresan, "Binary multiplication with PN sequences," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 36, no. 4, pp. 603–606, 1988.
[19] K. Kim, J. Lee, and K. Choi, "An energy-efficient random number generator for stochastic circuits," in Asia and South Pacific Design Automation Conference, 2016, pp. 256–261.
[20] D. Jenson and M. Riedel, "A deterministic approach to stochastic computation," in International Conference on Computer-Aided Design, 2016, pp. 102:1–102:8.
[21] Synopsys, Inc., http://www.synopsys.com/.
[22] NanGate, Inc., http://www.nangate.com/.
[23] T.-H. Chen and J. P. Hayes, "Analyzing and controlling accuracy in stochastic circuits," in International Conference on Computer Design, 2014, pp. 367–373.
[24] K. P. Parker and E. J. McCluskey, "Probabilistic treatment of general combinational networks," IEEE Transactions on Computers, vol. 24, no. 6, pp. 668–670, 1975.
[25] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com
[26] K. Parhi and Y. Liu, "Computing arithmetic functions using stochastic logic by series expansion," IEEE Transactions on Emerging Topics in Computing, 2016.
[27] N. Onizawa, D. Katagiri, W. J. Gross, and T. Hanyu, "Gabor filter based on stochastic computation," IEEE Signal Processing Letters, vol. 22, no. 9, pp. 1224–1228, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY