Optimization of a Stochastic Number Generator Using D Flip Flop

N. Saraswathi[1], Dr.J. Selvakumar[2]

[1]Department of ECE, SRMIST, Chennai, India.
[2]Department of ECE, SRMIST, Chennai, India.
[1]saraswan@srmist.edu.in,[2] selvakuj@srmist.edu.in

Article History Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Escalating development in science and technology with exponential demand for gadgets and devices towards smart, intelligent and effective implementation for sustainable livelihood is the need of the society. The demand has fueled for improvising hardware and its design for enhanced computational speed, less latency, less power consumption, low cost involved in fabricating these hardware components. One such approach is stochastic computing, although developed long back, still the field is at naïve and rudimentary stage mandatory incorporation of latest circuit techniques and methodologies. The objective of our study is to incorporate D-Flip Flops (DFFs) within the stochastic number generator (SNG) architecture and assess the performance of stochastic circuit to break haven the cost involved in hardware fabrication, power consumption etc. Our study compared the positioning of DFFs within and outside SNG and compare the power consumption and area occupancy relating with efficiency. Experimental investigations revealed that DFFs within the SNG architecture was more optimal, effective and efficient, catering the needs and objectives of our study. Since our investigations is at preliminary stage, further warranting and validation is sought to take over to next stage of implementation.

Keywords: Stochastic Computing, Stochastic Number Generator, Random Number Source, D Flip Flop, Comparator.

1. Preamble

The demand for on compactness, energy saving and high reliability had made a pragmatic shift, looking new hardware designs for better contemporary computations. Traditional method of computing and scaling down the dimensions of hardware is paving an avenue for increased unreliability and more power consumption. To overcome the aforesaid technical snag, unconventional deterministic strategies were replaced by probabilistic methodology such as stochastic computing is reviewed and assessed. This stochastic approach integrates complex data in form of binary bits (0s and 1s) and implements using standard logic gates to stochastic form. Stochastic computational designs are developed based on stochastic approach, and all these stochastic computational elements influence better accuracy, reduced hardware consumption, error-tolerance and cost-effectiveness, fueled with increased computational speed and time [6].

2. Background of Stochastic Computing

Stochastic computing is probabilistic based method relied on two digits (0s and 1s), causing arithmetic operations cumbersome and inconvenient. This operational issue can be overcome by implementing scaled operations to meet the arithmetic demand. For better understanding, at any condition, a bit-stream (S) containing 75% 1s and 25% 0s is presented as p = 0.75, representing of occurrence of probability 1 at an arbitrary bit position is p, with no restrictions on length and structure of bit stream (S). In stochastic computing, multiplication can be performed by using a circuit consisting of AND gate, with two binary bit streams acting as input variables (p1 and p2) and output as product of chance (S1×S2) with an assumption that two input streams are unrelated with each other. Figure 01(a), represents the basic multiplication of two numbers using stochastic operator AND gate operation, with two input variables (S1 = 4/8) and (S2 =6/8) respectively. Literally, it will have one output (4/8×6/8 =3/8) (S3). Figure 01(b) represents the similar multiplication, with different output (S3) (4/8×6/8 =2/8). Figure 01(b) output is considered as approximation to the precise output represented in Figure 01(a), as the circuits convert binary form to stochastic numbers and stochastic to binary number converter using stochastic multiplier.

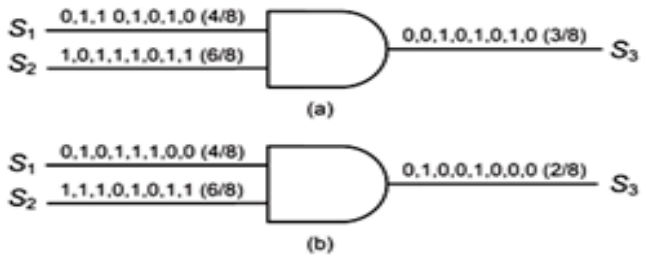

Figure 1. Stochastic Multiplier with AND Gate

Stochastic computing is a complex robust process, were stochastic number generators maps random binary bit streams (numbers) to stochastic bit streams, accounting for effective computation. Figure 02 represents binary-to-stochastic conversion circuit. The input binary number (m-bit sequence) is generated arbitrarily by pseudorandom number generator in every clock cycle. A comparator then scrutinizes the values and checks the input from the random generator number(x). If the number (x) is smaller than the binary range (y), then the comparator generators a bit parity value of "01" as output of the comparator at every clock and generates stochastic number. It produces the value "0", if binary range is greater than generator number, thereby assuming that the random numbers are uniformly distributed over the interval (0,1). The stochastic number's worth p is carried by the amount of 1s in its bit-stream type.

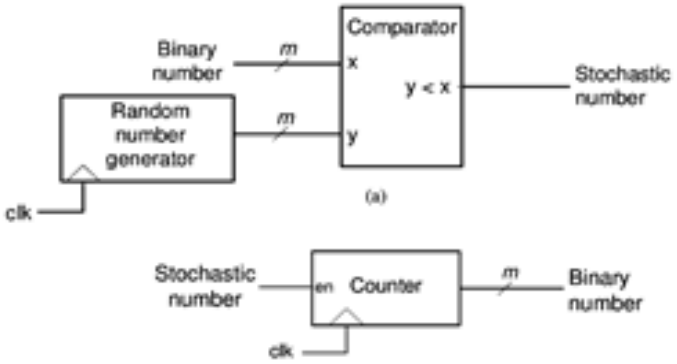

Figure 2. Number Conversion Circuits
(a) Binary to Stochastic and b) Stochastic to Binary)

Stochastic circuits operate based on stochastic bit streams comprises input (*x)*, generated as sequence of random bits based on probability (either one or zero), where each probability *x* is considered as being one and probability 1-*x* of being zero with an equal weight for every bit. Unipolar representation/encoding is non-negative value where, *x* is considered as the number to be encoded and *p* as probability of a stochastic sequence. The simplest mapping is representing a real number within the range (*x≤0≤1).* Bipolar mapping takes a linear transformation of unipolar and it is represented within the range (-1≤*0≤1).* Basically, the stochastic representation of a given number is not unique. In order to overcome such issues, stochastic computing uses redundant number representation system towards representing "n" numbers, where the stochastic number may be a binary sequence of length n with n_1 ('01's) and 1- n_1 ('0's). There are few limitations in stochastic computing such as; correlations inaccuracies during number combinations, errors arising during altering numbers, random fluctuations during representation. Stochastic number inaccuracies can be reduced by employed stochastic number generators, which turn our random and unrelated numbers efficiently by incorporating Linear Feedback Shift Registers (LFSRs). LFSRs have m flip-flops with each cycle passes through distinct state ($n=2^m-1$), excluding all zero states. The generated binary sequence is deterministic, pseudorandom, equal number proportion (0's =1's) etc., by nature.

3. Optimization of SNG's Using D Flip Flops

Stochastic computing has wide applications spanning from reliability analysis, polynomial arithmetic, basic arithmetic calculations, analog and hybrid computational transformations, neural networks, control systems, image processing etc. Despite varied applications, it has its own challenges and limitations such as cost of hardware and excessive power consumption with leading to longer computational dormancy respectively. Figure 2(a) represents the stochastic number generator architecture consists of a random number source and a comparator. The primary role of random number generator is to generate a random uniformly distributed binary number (R) at every clock cycle. LFSR and CMP works coherently back-to-back and CMP compares binary number (R) with a constant binary number (X) and the output is represented as "1", if R < X, and a "0" otherwise.

The comparator compares and 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, advocating to the fact that number of number generators must be number of quantity of inputs of the circuit. All these above factors, result larger architecture of circuit mandating more power consumption and reduced power potency of computing core. An architecture redesigning was carried out to use D flip-flops (DFFs), to reduce circuit delay, computational latency and bit stream errors. One simple solution to reduce the hardware price cost is to optimal sharing of RNS with all SNGs in the architecture. In Figure 3(a), three DFFs were located outside stochastic number generator after comparator (CMP) as per recommendations. In figure 3 (b), represents an improved architecture, minimum of two DFFs was inserted inside the stochastic number generator (where SNGs is equivalent

to inserting DFFs after the comparator). DFFs must be used both inside and outside the SNGs and larger design space with reduced number of DFF.

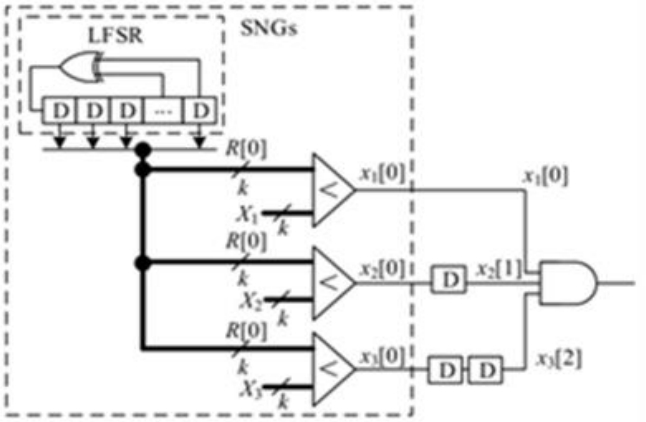

Figure 3(a). D Flip Flops Outside SNG

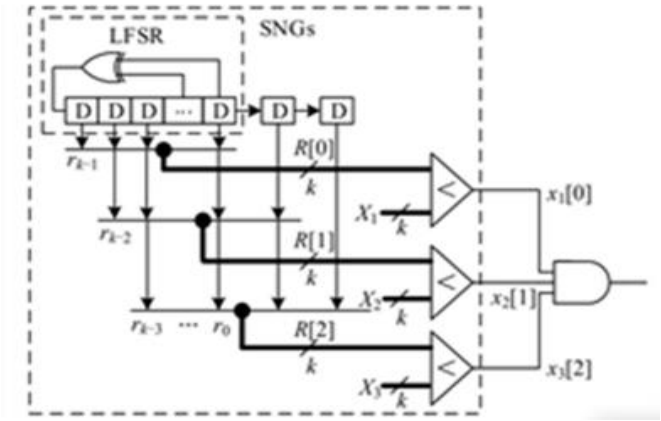

Figure 3(b). D Flip Flops Inside SNG

The experimental set up was designed to study the influence of insertion of DFFs into the stochastic circuit architecture and study the impact of its insertion on the power consumption and area occupancy. An 8-bite LFSR was selected as random source, we compared the efficiency of insertion of DFFs within and outside SNGs in terms of power consumption (µW) and area occupancy (µm^2). Experimental investigation reveals that insertion of DFFs within the SNG, the stochastic multiplier consumes less power and less space. Inference from the Table 01 and based on scientific literatures have reported that a single DFF consumes 4.3% power and 3.9% of area in the circuit architecture. Incorporation of many DFFs in the architecture, would be added advantage in terms of power consumption and area occupancy. But still, it has its own limitations as, DFF incorporation directly influences computation accuracy of the circuit, which have to be validated by RMSE.

Table 1. Comparison of SNGs

Component	**DFF outside SNG**		**DFF inside SNG**	
	Power (µW)	**Area (µm^2)**	**Power (µW)**	**Area (µm^2)**
SNG	50.2	163.9	33.7	110.1
SC core	0.6	1.1	2.4	5.9

4. Conclusion

Stochastic computing is futuristic computing methodology with lower power consumption, limited architecture, error tolerant with high density with wide varied applications in modern electronic world. The stochastic architecture hardware was redesigned by incorporating DFFs into the circuit, to study the efficacy and efficiency of power consumption, computational power and latency. The presence of DFFs inside and outside SNG was analysed and it was found the DFFs within the SNG proved to be reduce power consumption, less area occupancy with less computational time compared with DFFs outside SNG.

References

1. Alaghi A, Hayes JP. Survey of stochastic computing. *ACM Transactions on Embedded Computing Systems,* Vol. 12, No.2, 2013: 1-19.
2. Kaining Han, Junchao Wang, Warren J.Gross, Jianhao Hu, Stochastic Bit Wise Iterative Decoding of Polar Codes", *IEEE Transactions On Signal Processing,* Vol. 67, No. 5, March 2019
3. Zhijing Li, Zhao Chen, Yili Zhang, Zixin Huang and Weeikang Qian, "Simultaneous Area and Latency Optimization for Stochastic Circuits by D Flip Flop Insertion", *IEEE Transactions On Computer Aided Design of Integrated Circuits and Systems,* Vol 38, No.7, July 2019.
4. Han, Jianhao Hu, Jienan Chen, Zhengbing Zhang and Hao Lu, "Fast Converging Normalization Unit for Stochastic Computing", *IEEE Transactions on Circuits and Systems,* Vol.65, No:4, April 2018.
5. Armin Alaghi, Weikang Qian, John P. Hayes, "The promise and challenge of Stochastic Computing", *IEEE Transactions On Computer Aided Design of Integrated Circuits and Systems,* Vol 37, No.8, August 2018.
6. Budhwani R.K., Ragavan R, Sentieys O, "Taking advantage of correlation in stochastic computing," *2017 IEEE International Symposium on Circuits and Systems (ISCAS),* Baltimore, MD, USA, 2017, pp. 1-4. doi: 10.1109/ISCAS.2017.8050807.