Implementing Stochastic Multilevel Processing Circuits in FPGA

Könzgen, P. S.[#] pietroserpa@yahoo.com.br
Marques, W. R.[#] williamrodriguesmarques@gmail.com
Souza Jr., A. A.[#] adaojr@gmail.com

[#] Federal Institute of Education, Science and TechnologySul-Rio-Grandense

ABSTRACT
In this work is presented a multilevel stochastic circuit. It is an innovating approach compared to current works. Some of the advantages of using stochastic computing are that it is possible to build processing circuits using little hardware. Furthermore this technique is an attractive solution when fault tolerance is desired. Nevertheless this technique presents some disadvantages: a dynamic performance that slows with the need for resolution and random fluctuations of stochastic bit-streams. Thereby is presented an approach that aims to reduce the stochastic bit-stream variance and increase the dynamic performance through dynamic range subdivision.

Categories and Subject Descriptors
G.3 [**Probability and Statistics**]: *distribution functions,random number generation, statistics computing.*

General Terms
Measurement, Performance, Design, Reliability, Experimentation, Verification.

Keywords
Stochastic computing, oversampling, stochastic logic.

1. INTRODUCTION
Stochastic processing is a well-known technique to design arithmetic circuits by encoding variables as expected values of uncorrelated pulse streams [1]. Changing numeric data representation from binary radix to stochastic streams allows for arithmetic operators that consume a very low amount of area and are well suited to algorithms with massive parallelism of operators [2]. Also, stochastic modulation is one of the data representation techniques that have a natural resistance to soft errors and a tendency to show graceful performance degradation when subjected to multiple failures [3],[4].

Since its first introduction, stochastic circuits (SC) have been used to address many different applications [2], [3], [4]. More recently stochastic arithmetic has been used to LDPC decoding [5] and image processing [6]. Some recent research has focused on systematic design methodologies for stochastic operators using finite state machines [7], [8] and spectral transforms [9].

Main disadvantage of stochastic arithmetic is its demand for relatively high number of cycles to accurately represent variables with a given resolution. Contrary to radix binary numbers where word length increases linearly with the resolution r, in stochastic arithmetic word length is an exponential function of r. Compared with radix binary arithmetic, the stochastic arithmetic presents larger computational error due to stochastic bit-streams fluctuations. Also, although research indicates that time/area product favours SC over binary radix serial (BRI) architectures for resolutions below ten bits [4], SC presents variance degradation along the data path that makes then harder to successfully design [9].

Since variance control in the circuit is fundamental to its precise operation, a thorough variance analysis must be included in the SC design flow [9]. Although variance in the output of the stochastic number generators (SNG) closely resembles theexpected values for a Bernoulli series, subsequent operators will change its distribution. It is, therefore, very important for SC design that to have a tool to estimate variance.

When compared to recent research of in SC, this work presents some important differences. Previous research has focused mainly on single bit representation of stochastic signals [2], [3], [4], [5]. SC is mainly explored as a way to reduce the area taken by the operators on the implementation of massively parallel algorithm. Our work mainly aims to take advantage of SC fault tolerance characteristics; therefore we focus on parallel stochastic data representation as a way to reduce latency issues. A recent paper proposes a parallel stochastic circuit to perform numerical integration [6], it does not, however, explore the dynamic range sub-division to create a multilevel parallel stochastic coding like this paper does.

Finally, our proposal for multilevel stochastic is a technique that involves the weighting, or masking, of the signals in the stochastic number generator (SNG). The idea bears some resemblance to the weighted stochastic series introduced by Gupta and Kumaresan to prove the feasibility of exact stochastic multiplication [10]. Our proposal, however, starts from the full dynamic range of the signal and make a few partitions while previous work operates a bit by bit weighting. This difference implies that our paper must define new stochastic operators to perform both summation and product on multilevel stochastic coded (MSC) signals.

The remainder of this paper is organized as follows: section 2 presents the basic principles of stochastic computation, their resolution and variance characteristics. Section 3 presents the multilevel and parallel data coding techniques. Simulations of the proposed methods are shown in section 4. Section 5 presents resultsand discussions of the practical implementations and section 6 conclusions and future works.

2. STOCHASTIC ARITHMETIC
Stochastic systems make pseudo analog operations using stochastically coded pulse sequences [10]. Stochastic computing is an unconventional technique that processes data in the form of probabilities represented by bit-streams [5].This type of representation allows arithmetic operations are performed through simple circuits.

The information is converted into a synchronous pulse sequence that codifies the information as the probability, at a given clock cycle, of the pulse being at “high level”. Equation 1 defines a stochastic pulse stream.

$$p_x(t) = \sum_{k=-\infty}^{\infty} u_k(t) \cdot \chi_k \quad (1)$$

Where

$$u(t) = \begin{cases} 1\,, k\,T < t < (k+1)T \\ 0\,, else \end{cases} \quad (2)$$

And

$$\chi_k = \begin{cases} P[\,\chi_k = 1\,] = p \\ P[\,\chi_k = 0\,] = 1 - p \end{cases} \quad (3)$$

The process of change the numeric data representation from binary radix to stochastic streamsinvolves the generation of an n-bit random binary number in each clock cycle by a pseudo-random number generator, and comparing it to the n-bit input binary number. The comparator produces 1 if the random number is less than the binary number and a 0 otherwise. Assuming that the random numbers are uniformly distributed over the interval [X_{min},X_{max}], the probability of a 1 appearing at the output of the comparator at each clock cycle is equal to the binary input of the converter interpreted as a fractional number (see figure 1) [2].

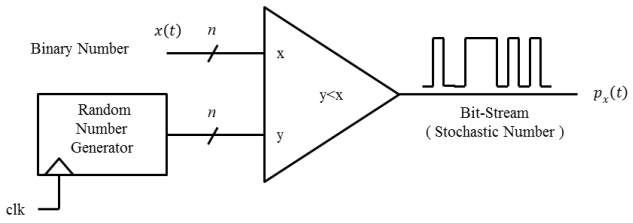

Figure 1– Binary number to stochastic bit-stream converter.

To obtain the original information,that is, to convert the stochastic number to the radix binaryformatit is necessary to remember that the stochastic number value $x(t)$is given by the density of 1's in its bit-stream form, thus it suffices to count these 1's in order to extract p[2], this process can be viewed as a low pass process.

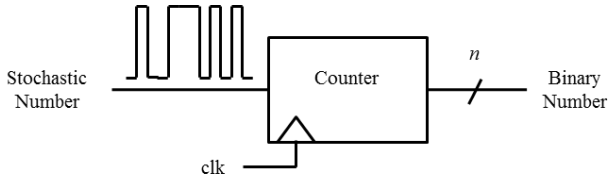

Figure 2 – Stochastic bit-stream to binary radix converter.

Whereas xisa limited binary number within the range [Xmax, Xmin], it can be represented as a bit streamp$_x$ with symbols $\{\wp 0, \wp 1\}$, the expected value of p_x is given by equation 2. Where X_N is the normalized value of x,and its limits are dependent on the chosen domain (symbols {p0, p1}).

$$\mathrm{E}\{p_x\} = \mathrm{x_N} \quad (4)$$

Table 1 shows the normalization to fourSC domains. Throughout this paper, unless told otherwise, we will be working with unipolar representation (UP). It must be noted that results can be generalized for the other domains.

Table 1 – Stochastic circuits domains normalized.

Normalization	Alphabet	SC domain
$x_N = \frac{x - X_{min}}{X_{max} - X_{min}}$	$\{\wp_0=0, \wp_1=1\}$	Unipolar (UP)
$x_N = \frac{X_{min} - x}{X_{max} - X_{min}}$	$\{\wp_0=0, \wp_1=-1\}$	Inverse unipolar (IUP)
$x_N = \frac{x - (X_{max} + X_{min})}{X_{max} - X_{min}}$	$\{\wp_0=-1, \wp_1=+1\}$	Bipolar (BP)
$x_N = \frac{(X_{max} + X_{min}) - x}{X_{max} - X_{min}}$	$\{\wp_0=+1, \wp_1=-1\}$	Inverse Bipolar (IBP)

2.1 Resolution and Convergence

Given that $\mu_{p_x,K}$, where K is the word length of the bit-stream, is a K-points estimator of the average of $p_x$given by $\mu_{p_x,K} = \frac{1}{K}\sum_{k=1}^{K} p_x[k] \rightarrow E\{p_x\}$, its variance will be inversely proportional to K. Assuming $x_N$is a constant value normalized in the UP domain, the resolution of the stochastic number that represent $x_N$will be limited by the standard deviation of $\mu_{p_x,K}$. Or considering a particular resolution r, the minimum value of K that ensures convergence ofexpected value of p_x to x_N is given by equation (3).

$$\mathrm{K} > 2^{2r-2} \quad (5)$$

Assuming p_x as a Bernoulli sequencedefined in equation (1), we can estimate their variance about the dynamic range in equation (4).

$$\sigma_\mu^2 = Var\left(\mu_{p_x,K}|_{x(t_0)=x_N}\right) = \frac{x_N \cdot (1 - x_N)}{K} \quad (6)$$

Maximum value for variance will occur in the center of the dynamic range. This is the variance in the output of the stochastic number generator, in the output of stochastic operators the behavior will be different.

2.2 Stochastic Operators

Let p_x and p_y be binary pulse streams representing respectively two values $x(t_0)$and $y(t_0)$. A single AND port implements the product $z(t_0) = x(t_0) \cdot y(t_0)$in the UP domain. In a bipolar representation (BP or IBP) the product can be implemented by an EXOR gate. Also, since for any values of $x_N$and y_N, results of $z_N = x_N + y_N$ will generate an output with double of the inputs dynamic range, weighted summation is performed sampling the stochastic series using a multiplexer and an additional variable p_{sel} (E{p_{sel}}=0.5). Figure 3 shows the main stochastic operators.

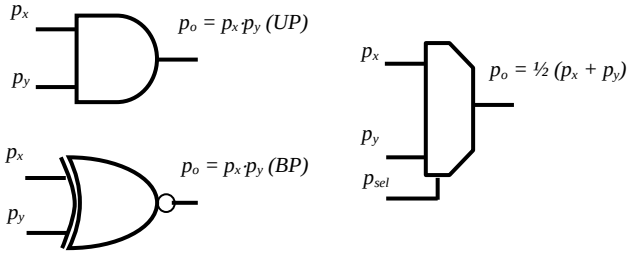

Figure 3 – Stochastic arithmetic operators for the UP and BP stochastic domains.

3. DATA REPRESENTATIONS

In stochastic representation the length K of the bit-stream should be chosen so as to guarantee the correct encoding of the value with a resolution r assuming a fully serial encoding, we will have the maximum input frequency restricted by Fs = 1/K.Ts. The high length of K is the major problem of stochasticcomputing.Different methods have been proposed in order to reduce this latency.In this paper we propose a new method of data representation based on the subdivision of the dynamic range in Lparts. For comparison purposes it is used the parallel codification method.

3.1 Parallel Data Representation

In stochastic parallel data representation each value is represented by J parallel stochastic numbers generated with uncorrelated random sequences (see figure 4). As the series are uncorrelated one only needs to change the S2R circuit to allow the summation of parallel pulse streams. This allows a higher resolution and requires smaller values of K generating smaller latencies. Thus the resolution for parallel stochastic coded (PSC) will thus be defined by the product of K and J. It is easy to see that the variance of the average of the means of the output bit-streams p1(x) and p2(x) will be smaller than the variance of the means of p1(x) and p2(x). In section 4 simulation results shows this obviousness.

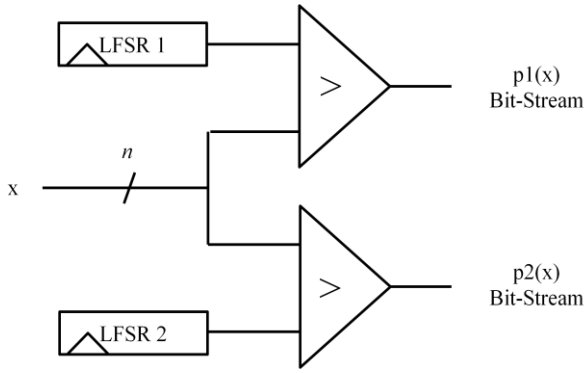

Figure 4 – Parallel Stochastic Circuit (PSC) generating two stochastic parallel bit-streams, J=2.

3.2 Multilevel Data Representation

In this work we propose a new method to coding stochastic number based in the subdivision of the dynamic range in L parts, each part of the division is separately encoded in a stochastic bit-stream.The equation below shows the relationship between length K of bit stream with the number of subdivisions (L) of the dynamic range andthe number of redundant parallel circuits (J).

$$K > \frac{2^{2r-2}}{J \cdot L^2} \qquad (7)$$

In figure 5 a schematic diagram representing the multilevel stochastic circuit subdividing the dynamic range into two parts is presented.

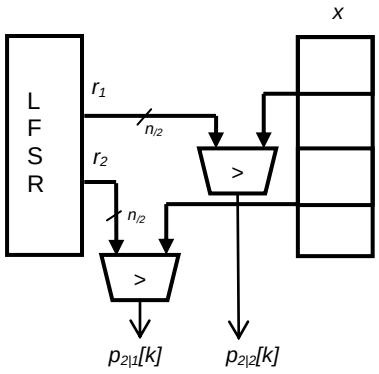

Figure 5 – Multilevel Stochastic Circuit (MSC) generating two stochastic pulse streams with L=2

As a convention for multilevel stochastic coding (MSC) each pulse stream that encodes part of the dynamic range of variable is numbered starting from the most significant section. Thus $p_{x1|2}$ is the pulse that encodes the upper mid-section of the variable x, and $p_{x2|2}$ its bottom section.

4. SIMULATION RESULTS

The quality of the stochastic codification can be measured by their variance, through it we determine the length K required to correct a stochastic representation with a resolution r. Based on that, we simulated the variance for a fixed value of K using theMSC and the PSC. In the figure below we can see that increasing the number of parallel stochastic circuits the variance slowly decreases. On the other hand the multilevel coding variance decreases considerably. Figure 6 presents a comparison between the two techniques: PSC and MSC. This figure shows that using two parallel sections has twice the impact of parallelism without range subdivision.

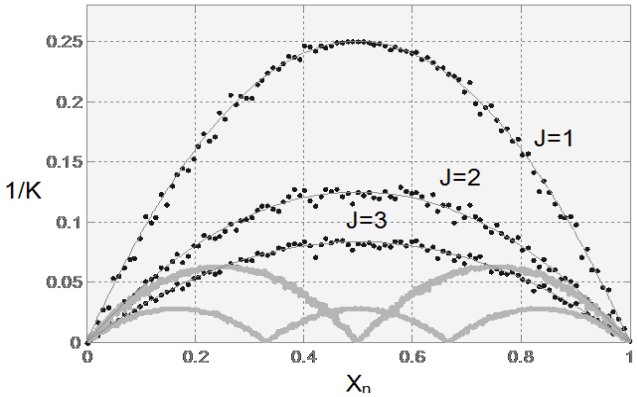

Figure 6 – Comparing variance PSC (black) and MSC (gray) stochastic circuits. Variance is normalized by 1/K, where K is the average estimator depth. SNG using 10 bits LFSR.

5. PRACTICAL RESULTS

To test the key concepts described in this work, we implemented a prototype circuit using an Altera® prototyping board with EP2C35F672C6FPGA.The circuit implemented performs the multiplication of two sine waves with frequencies f1 = 30 kHz andf2 = 60 kHz, both signs were converted to PSC (J= 2) and MSC (L=2) (see figure 7).The output bit-streams were acquired using the NI ELVIS II prototyping board and processed using the MATLAB. Figure 7 shows the circuit PSC implemented infpga

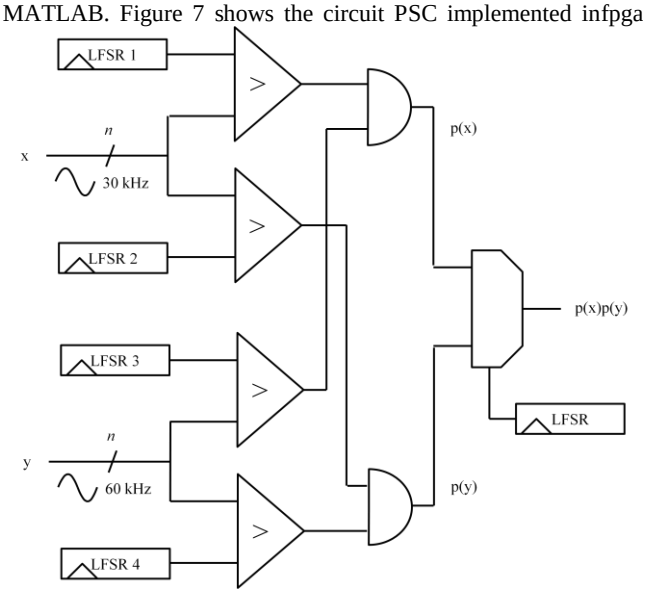

Figure 7 – PSC implemented on the FPGA prototyping board

Figure 8 presents the MSC implemented in FPGA. After acquire the outputs po1/2 and po2/2, the data were added into MATLAB

in order to eliminate the variance interference caused by an stochastic adder circuit implementation.

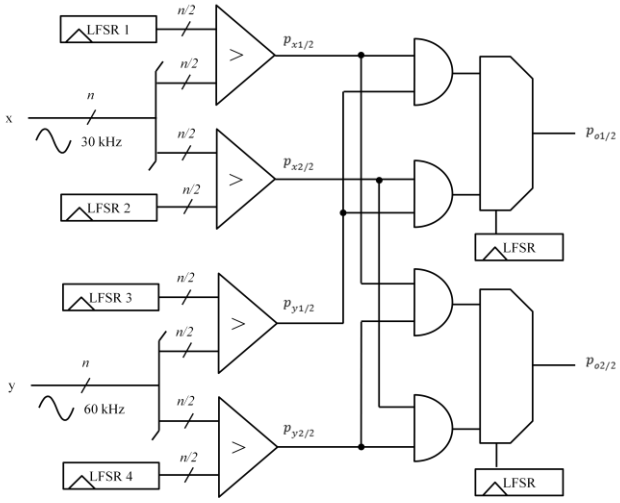

Figure 8 – Multilevel Stochastic Circuit multiplier implemented in FPGA with the subdivision of the range into two parts, L=2.

Figure 9 shows the frequency spectrum of the bit-stream resulting from the multiplication of two sine waves, f1 and f2 using MSC and PSC techniques.The resulting frequency spectrum of two sine waves is the convolution of the frequency spectrum of a sine wave f1 and f2. The amplitude of the main frequency peaks of the MSC is greater than the PSC, this shows that the resulting spectral spread of multiplication was greater in the PSC.Therefore the technique MSC obtained a better performance than PSC.

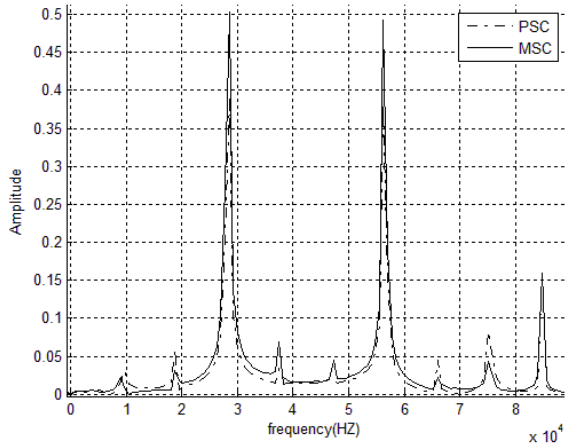

Figure 9 – Frequency Spectrum of the the PSC and the MSC multiplier

Table 2 shows the logical synthesis of circuits implemented, note that the hardware costin MSC is smaller than the PSC due to the smallerLFSR used in MSC. The complexity of arithmetic circuits in MSC increases considerably when L increases, the behavior of the variance with the L increasing still needs to be better studied in practical circuits.

Table 2 – Synthesis results for a single stochastic multiplier

Block	Logic cells	Register bits
PSC	318	66
MSC	301	70
Sin 30kHz	87	8
Sin 60kHz	140	8

6. CONCLUSIONS AND FUTURE WORKS

The results indicate that MSC is a viable alternative to implement stochastic arithmetic systems. It can be useful when combined with direct redundant parallelism and customized for a particular application.

A future work will be focused on the understanding of the variance propagation in different stochastic processing circuits. In addition, it will also be compared multilevel and parallel stochastic circuits for various failure sceneries and evaluate its robustness.

7. REFERENCES

[1] Gaines, B.R., “Stochastic computing,” Proc. AFIPS Spring Joint Computer Conf., pp.149-156, 1967.

[2] Alaghi, A., Hayes, J.P., “Survey of stochastic computing,” ACM Trans. Embedded Computing Systems, 2012.

[3] Peng Li; WeikangQian; Lilja, D.J. “A stochastic reconfigurable architecture for fault-tolerant computation with sequential logic”. Proc. 2012 IEEE 30th International Conference on Computer Design (ICCD), pp. 303-308, 2012.

[4] Brown, B. D. and Card, H. C., “Stochastic neural computation I: Computational elements,” IEEE Transactions on Computers, vol. 50, pp. 891–905, September, 2001.

[5] Alaghi, A.; Hayes, J. P. “A Spectral Transform Approach to Stochastic Circuits”, Proc. of the 2012 IEEE 30th International Conference on Computer Design – ICCD’12. Pp. 315-321, 2012.

[6] Wang, C.; Li, P. Lilja, D. J.; Bazargan, K.; Riedel, M. D. “An efficient implementation of numerical integration using logical computation on stochastic bit streams”, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Pp. 156-162, Nov, 2012.

[7] L., Peng; Qian, W; Riedel, M. ;Bazargan, K.; Lilja, D. J. “The Synthesis of Linear Finite State Machine-Based Stochastic Computational Eleements”. Proceedings of 2012 17th Asia and South Pacific Design Automation Conference - ASP-DAC, pp. 757-762, 2012.

[8] Li, P; Lilja, D. J.; Qian, W.; Bazargan, K.; Riedel, M. ; “The synthesis of complex arithmetic computation on stochastic bit streams using sequential logic”. Proceedings of the International Conference on Computer-Aided Design - ICCAD '12. pp. 480-487, 2012.

[9] Ma, Chengguang; Zhong, Shunan, Dang, Hua. “Understanding Variance Propagation in Stochastic Computing Systrems” 2012 IEEE 30th International Conference on Computer Design (ICCD), pp. 213-218

[10] Toral, S.L.; Quero, J.M.; Franquelo, L.G. “Stochastic Pulse Coded Arithmetic” 2000 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 599-602, 2000.

[11] Widrow, B. and Kollár, I. "Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications," Cambridge University Press, Cambridge, UK, 2008. 778 p.