Multi-level Stochastic Processing Circuits

Adão A. Souza Jr[#1], Pietro S. Konzgen[*], William Marques[#]

[#]*Curso de Engenharia Elétrica, Instituto Federal Sul-rio-grandense*
Pelotas, Brasil

[1]adaojr@pelotas.ifsul.edu.br

Abstract— **In this work, a weighted multi-level stochastic representation and its operators are introduced to increase alternatives for design space exploration. Multi-level stochastic circuit architectures are presented and area tradeoffs discussed. Stochastic arithmetic circuits allow better fault tolerance by encoding signals in pseudorandom pulse streams. However, this comes at the expense of higher latencies and a worst dynamic behaviour. Also, in stochastic circuits, signal bandwidth, pseudo-random (PN) sequence length and variance are related to maximum number of stochastic operations one can perform before signal regeneration an that creates a limit to their complexity.**

Keywords*— *Probabilistic computation, stochastic computing, stochastic logic

I. 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*. 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 the expected 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 remaining of this work is organized as follows: section II introduces classical unipolar and bipolar stochastic number representation, its main operators and variance characteristics. Section III introduces multilevel stochastic quantization and its operators. Area tradeoffs are discussed on section IV. A final discussion and future work are presented in section V.

II. STOCHASTIC ARITHMETIC

Assuming a signal *x(t)* such as its dynamic range in confined to the interval $[X_{min}, X_{max}]$, and it is sampled with frequency Fs=1/Ts. A binary series $p_x(t)$ with symbols $\{\wp_0, \wp_1\}$ is defined such as at any given point $to = no \cdot Ts$, one can find an interval ΔT such as for $t \in [t_o-\Delta T/2, t_o+\Delta T/2]$, the expected value of p_x is given by (1). Where x_N is the normalized value on $x(t_o)$.

$$E\{p_x\} = x_N \tag{1}$$

Combining different ways to normalize the input and attribute values to the binary stream alphabet one ends up with

four SC domains (Table 1) [2]. 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 I
STOCHASTIC ARITHMETIC DOMAINS. THIS TABLE EXPANDS [12].

Normalization	Alphabet	SC domain
$x_N = \frac{x - X_{min}}{X_{max} - X_{min}}$	$\{\wp_0=0, \wp_1=1\}$	Unipolar (UP)
$x_N = \frac{X_{min} - x}{X_{max} - X_{min}}$	$\{\wp_0=0, \wp_1=0\}$	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)

Stochastic number generation can be seen as a process where a random series of numbers is compared with a constant value. This process is analogous to one bit quantization with uniform dither addition. Figure 1 shows a signal *x(t)* compared to random uniform sequence *r*. This can be modeled as a signal $c(t) = x(t) - r$. Output probability distribution of p (represented as f_p) can be found using quantization theory [11]. Distribution f_c will be given by the convolution between f_x and the random distribution of the reference *r*. Observe that in a digital implementation *x(t)* and *r* will have discrete distributions.

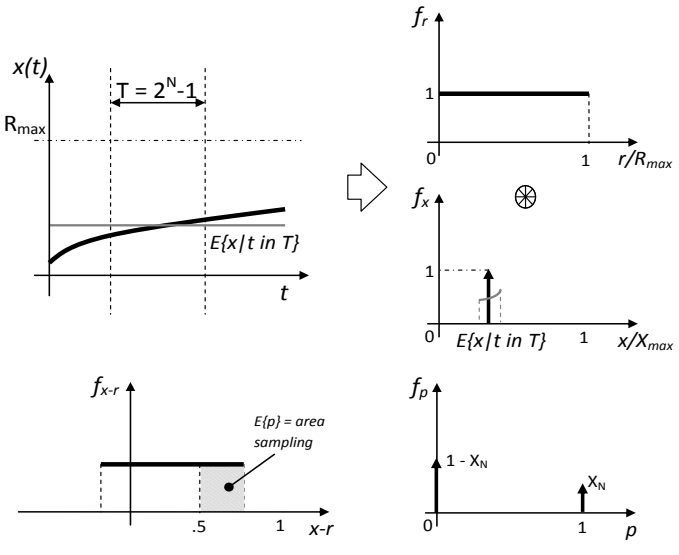

Figure 1. Expected value of p_x in the output of a comparator will be an approximate of x assuming its value is a constant in the interval. For N bits LFSR, random sequence length 2N-1 must be long enough to allow this.

The resulting signal is passed by a quantization function with threshold in 0.5. Probability of the symbol $\wp_1$ will be calculated area sampling f_c. If *x(t)* is approximately constant in the interval *T*, the expected value of the resulting series *p* will be a good approximation to the value of *x(t)*.

Although in some circuits data can be acquired from the analog domain using statistical samplers that will already give stochastic number series as outputs [12], in most applications signals must be first converted from a radix representation to stochastic numbers, processed in one of the stochastic arithmetic domains and them converted back to radix form. Stochastic number generation uses a pseudorandom number generator either implemented as a LFSR or cellular automata [13]. Conversion from stochastic to radix form (S2R) is performed by an accumulator and can be viewed as a low pass process. In some cases, stochastic number regeneration may be necessary to minimize variance degradation.

A. Resolution and Convergence

Given that $M(p_x, K)$ is a K-points estimator of the average of p_x given by $M(p_x, K) = \frac{1}{K}\sum_{k=1}^{K} p_x[k] \rightarrow E\{p_x\}$, its variance will be inversely proportional to K (2). Assuming $x(t_0)$ is a constant value in the averaging interval, resolution of the signal in the UP domain will be limited by the standard deviation of $M(p_x, K)$. Also in the limiting case for a resolution r one will have a minimal value for K given by equation (3).

$$K > 2^{2r-2} \tag{2}$$

Assuming a Bernoulli series also allows us to estimate the variance of p_x over the dynamic range by equation (3).

$$\sigma_M^2 = Var(\, M(p_x, K)|_{x(t0)=X}) = \frac{x_N \cdot (1 - x_N)}{K} \tag{3}$$

Maximum value for variance will occur in the center of the dynamic range. As we will see in the following sections that is not the case for the output of stochastic operators.

B. Stochastic Operators

Let p_x and p_y be binary pulse streams representing respectively two values $x(t_0)$ and $y(t_0)$. Product $z(t_0)=x(t_0)\cdot y(t_0)$,. in the UP domain can be implemented by a single AND gate. It can also be shown that for a bipolar representation (BP or IBP) the product will be implemented by an EXOR gate. Also, since for any values of x_N and $\mathbf{y_N}$, 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 2 shows the main stochastic operators.

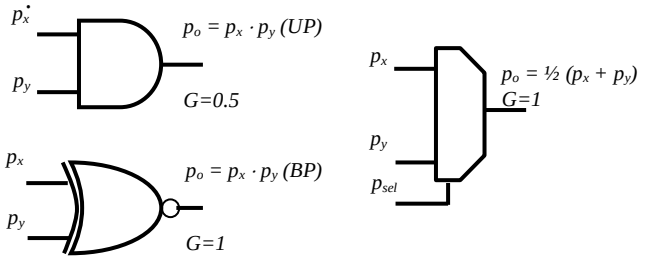

Figure 2. Stochastic arithmetic operators and gains for the unipolar (UP) and the bipolar (BP) stochastic arithmetic domains. Weighted addition is performed with a MUX with an auxiliary stochastic variable (p_{sel}). Product is either a AND (UP) or a XOR (BP) gate. Inverse domains (IUP and IBP) will have inverted product operator outputs.

Although equation (2) gives a good estimate for the variance in the output of the SNG, output of stochastic operators has a different behavior [9]. An analysis of the relationship between the stochastic arithmetic critical path and the circuit behavior in the frequency domain is due but beyond the scope of this paper.

III. Parallel and Multilevel Data Representations

Since in most SC applications *K* is chosen such as it guarantees a given resolution, and assuming totally serial encoding one will have the maximum input frequency restricted by $Fs = 1/K{\cdot}Ts$. Latency and bandwidth are thus important limits for the application of stochastic computing. Parallel data representation is a way to minimize this issue.

The main concept is that each variable can be represented by *J* parallel stochastic numbers generated with uncorrelated random sequences. As the series are uncorrelated one only needs to change the S2R circuit to allow the summation of parallel pulse streams. This will allow a higher resolution and require smaller values of *K* generating smaller latencies. Resolution for parallel stochastic coded (PSC) will thus be defined by the product of *K* and *J*.

In multilevel stochastic coding (MSC), dynamic range of the values in the SNG is split in L parts and each part is separately encoded in a pulse stream. Resolution (r), dynamic range subsection (L), redundant parallelism (J) and averaging depth (K) are related by equation (4)

$$K > \frac{2^{2r-2}}{J \cdot L^2} \tag{4}$$

Figure 3 shows the proposed multilevel stochastic number generator for L=2.

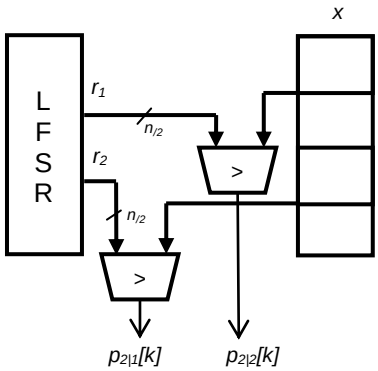

Figure 3. Multilevel Stochastic Number Generation. A single LFSR can be used to generate several uncorrelated sequences. For a 2-section MSC generator LFSR will have n/2 bits.

As a convention for 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.

MSC with two sections can use the same stochastic adders of figure 4. To understand this, one must that any operation all of the sections but one will be saturated either with a value of one or zero. If L=2 the outcome will be two pulse streams with a correct MSC value. If L is a power of two any sum operation can be performed.

A 2-sections MSC multiplier uses more gates than conventional stochastic multipliers to take into account cross products between different sections of each variable (Figure 4). In order to generate the correctly MSC output streams probability space must be sampled in a way that correctly divides it in the right number of sections. As in the case of the adder, the 2-section multipliers can be combined to generate larger ones for cases with a higher number of sections.

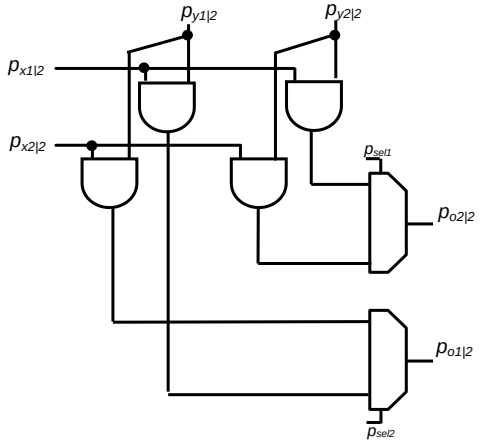

Figure 4. Multilevel stochastic multiplier for a data representation with two subsections (L=2).

IV. Variance and Area Tradeoffs

Since variance is related to effective resolution it is important to MSC and PSC variance characteristics. Figure 5 shows theoretical and simulated variance for a fixed value of K. As the number of parallel stochastic circuits increase variance slowly decreases. Mean standard deviation data for each configuration with K=1024 is summarized in table 3.

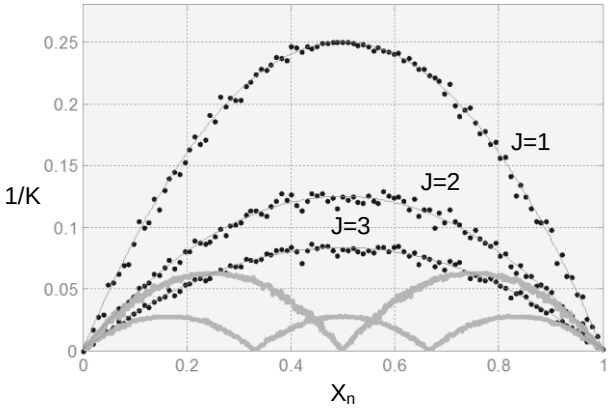

Figure 5. Comparing variance PSC (black dotted) and MSC (gray continuous) stochastic circuits. Using two parallel sections has twice the impact of parallelism without range sub-division. Variance is normalized by 1/K, where K is the average estimator depth. SNG using 10 bits LFSR.

Doubling the number of parallel data paths (J) will only increase resolution by less than half bit. Total area cost will be exactly two times the single path alternative. It is still an interesting alternative to reduce total latency, since it allows a

shorter averaging depth (K). On the other hand multi-level stochastic coding ads one bit at each doubling of L. Table III shows that for the same deviation of using L=2, even a J=3 was not enough. That means that for the same resolution MSC is more area effective than PSC. However, as L increases multiplier area increases with L^2 making it less attractive. The same is not true for adders Figure 6 shows the reconstruction of a MSC signal for different averaging depths (K).

TABLE III
MEAN STANDARD DEVIATION OBSERVED: ALL CASES USE S2R RECONSTRUCTION WITH AVERAGING DEPTH K=1023

Data Representation	Parallel Datapaths		
	1	2	3
PSC	0.0122	0.0086	0.0071
MSC	-	0.0061	0.0041

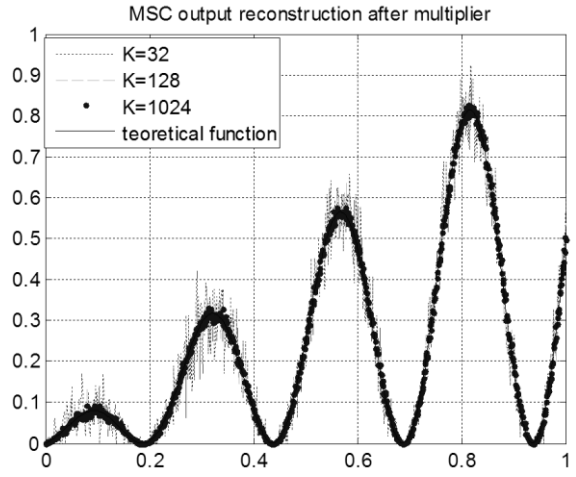

Figure 6. (a) Output for multilevel signal reconstruction with different values of K, J=1 and 2 sub-sections (L=2). Operation performed is $y = t \cdot (0.5 + 0.5 \cdot sin(8\pi t))$ with normalized time t∈[0,1].

To test the core concepts a prototype circuit was implemented using a EP2C35F672C6 FPGA board. The circuit implemented two sinusoid signals with frequencies f_1 = 30 kHz and f_2 = 60kHz both were converted to PSC and MSC (L=2) with parallel statistical sampling and multi-level stochastic. Pulse streams where acquired using a NI ELVIS II prototyping board and processed using MATLAB. Table 4 shows synthesis results.

TABLE IV
SYNTHESIS RESULTS FOR A SINGLE STOCHASTIC MULTIPLIER

Block	Logic Cells	Register bits	%LCs (input+SNG)/Total
PSC	329	103	68,99%
MSC	326	92	69,63%
sin 30kHz	140	8	-
sin 60kHz	87	8	-

V. FINAL REMARKS AND FUTURE WORK

Preliminary 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. Although increasing L at the beginning of the stochastic circuit data path seems an good design alternative one must keep in mind that this will probably impact the inherent fault tolerance of the system. The exact limits of this tradeoff are an issue that remains to be addressed in future research.

Work will be conducted on two main lines: fault tolerance and automatic synthesis. We intend to compare multilevel and parallel stochastic circuits for multiple fault scenarios and evaluate its robustness. Our focus is to understand how the way signals are spread in time and over circuit area will affects its inherent fault tolerance. We will also pursue a methodology to generate arbitrary functions and operators on multi-level stochastic circuits. The ultimate goal is the automatic design of arithmetic stochastic units for a given set of target reliability parameters.

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; Weikang Qian; 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, 2012 IEEE 30th International Conference on Computer Design (ICCD), 2012.

[10] Gupta, P. K.; Kumaresan, R. "Binary Multiplication with PN Sequences" IEEE Transactions on Acoustics Speech and Signal Processing, vol. 36, n. 4, April, 1988.

[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.

[12] Souza Jr. A.; Carro, L. "Highly Digital, Low-Cost Design of Statistic Signal Acquisition in SoCs." Design Automation and Test in Europe – DATE'04, pp. 10-15, France, 2004.

Wolfram, S. "Random Sequences Generation by Celular Automata" Advances in Applied Mathematics. v. 7, pp123-169, 1986.