

# Tolerating Technology Scaling Challenges using Stochastic Computation

Sara Belichki

College of Electrical Engineering & Computer Science  
University of Central Florida  
Orlando, USA

**Abstract**—As technology continues to scale down, the probability for hardware error to occur becomes a more prevalent issue. These issues can be attributed to manufacturing imperfections, voltage and temperature variations, and aging. Despite attempts to circumvent such issues, a different approach yields to the fact that hardware error exists and cannot be avoided. Stochastic computation (SC) is an approach that exploits such errors through describing their error-prone behavior with regards to statistical quantities and utilizing these terms toward computational means. Many varying implementations fall under the term stochastic computation ranging from logic gate computations, reliability evaluation, to signal processing. Each differ in how they represent SC but share the same aspect that the probability of error occurring in the hardware is utilized in such a way that functionality of the system can still occur. Such implementations will be discussed and explored within this survey in order to detail the current advancements and limitations that stochastic computation has undergone since its inception in the 1960s. It is with this introduction into SC, that other modern reliability approaches are compared to in order to evaluate the efficacy and feasibility of SC.

**Keywords**—*stochastic computation; probabilistic circuit; reliability analysis; estimation cost; fault-tolerant*

## I. INTRODUCTION

Stochastic computing (SC) was initially meant to be a low-cost alternative to the conventional binary computing that is still used today[1]. SC was defined as using deterministic means to represent and process probabilistic values[2]. The advantages that SC brought along was low design complexity for arithmetic units and the ability to tolerate errors within the system. The main idea developed during the 1960s, involved bit-streams that are represented as probabilities for circuits. Thus, both inputs and outputs are represented in a randomized set of bit streams. The original SC approach resulted in a few SC-based computers being built. However, long computation times, inaccuracies, and the success coupled with cost-effectiveness of binary circuits caused SC to fall out of favor overtime.

The direction that modern technology has undertaken over the past few decades involves the exponential down-scaling of CMOS logic. However, as we begin approaching the physical boundaries that Moore's Law can attain, errors within the circuit design of CMOS technology have become more prevalent. Manufacturing imperfections and aging degradations have become more prominent with this scaling,

leading to hardware faults that inevitably lead to functionality errors residing within the system. Scaled CMOS technology at the thermal limit of low operating voltages, makes it highly susceptible to discrepancies caused by noise[11]. This paper explores the original SC implementation alongside with modern SC approaches that seek to use some method of probabilistic aim based on circuit design to provide low-power and reliability. An example being, probabilistic circuit design that describes the resultant signal by the probability distribution[2]. Thus, it is here that we define SC in general as some means of using probability as a measure of confronting and quantifying errors within a probabilistic design as a way of maintaining system functionality. Once all approaches have been well delineated, some of them will be assessed and evaluated against other modern computing approaches that also highlight reliability from error-prone systems. This assessment will determine the value and future developments that come from SC.

## II. STOCHASTIC COMPUTATION

### A. SC Overview

As previously mentioned, the original SC approach represents bit streams in the form of probabilities based on the number of 1's found within the stream. The probabilities are computed by the ratio of the number of 1's found in the entire bit stream to the bit stream length. For example, a bit stream of 01101010 is represented by a stochastic number value of 4/8. The stochastic number is independent of the positions where the 1's may take place. This number is a representation of the probability that an input or output has a 4/8 probability of being a 1 during computation. With the probabilities computed per stream, it is the probabilities that are computed as opposed to the binary bits themselves. Figure 1 shows an example of using an AND gate as a multiplier for the stochastic numbers generated for the inputs S1 and S2. The resultant output S3 is simply the product of the stochastic numbers from S1 and S2. Notice that the AND operation of the binary bit streams follow the logic AND operation as well, thus verifying that the output stochastic number adheres to the binary operation.

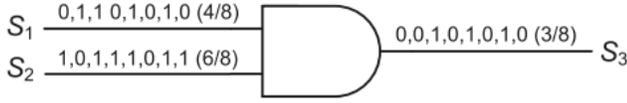


Fig. 1. AND gate as a stochastic multiplier [1]

However, this exactness in accurate stochastic numbers is not always the case. Discrepancies between the stochastic number and the resultant bit stream can occur. If we flip a bit in S1 using the prior example, the stochastic number generated is not the actual product of S1 and S2's stochastic numbers as showcased in Figure 2.



Fig. 2. Incorrect yet approximate result for stochastic multiplier [1]

This inexactness is tolerable as it is close enough to the true 3/8 result which showcases one of the most outstanding benefits that SC has to offer, error tolerance. If a bit flip in the stream barely affects the results in the overall computation, we are able to tolerate such errors without producing erroneous results. To better improve the precision of stochastic numbers, it is required that all inputs are uncorrelated and independent from one another. Furthermore, an exponentially longer bit stream will yield better precision but at the cost of longer computational times, one of the disadvantages to the original SC approach. A longer bit stream is more tolerant toward bit flips that may occur so that an approximate stochastic number can still be processed.

Stochastic number generators (SNGs) are circuit designs used to generate these randomized bit streams given a binary value. By taking in the binary fractional equivalent of a stochastic number, a pseudo-random number generator relays a random value per clock cycle that is the same bit length as the input binary value. A 1 or 0 is output for that clock cycle depending on if it is less than or greater than the input binary value respectively.

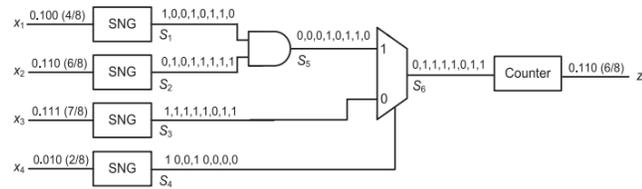


Fig. 3. Stochastic circuit implementing arithmetic design [1]

Figure 3 demonstrates a stochastic circuit that fulfills the arithmetic design  $z = x_1x_2x_4 + x_3(1 - x_4)$ . Notice that inputs  $x_1, x_2, x_3$ , and  $x_4$  are initially represented in their binary fractional equivalent of their respective stochastic numbers. After being processed through the SNGs, their randomized bit streams with respect to their stochastic numbers have been generated accordingly. As demonstrated in the Figure 1 example, the AND gate acts as a multiplier for S1 and S2's stochastic values. The multiplexer that results in S6 computes

based on the probability of that S4 is a 1, with which the probability of S5 is chosen. If the probability of S4 is a 0 (represented as  $(1-p(S4))$ ), then we consider the probability that S3 is selected at that instance. Thus, the probability of S6 is computed through the resulting equation [1] which forms its resulting stochastic value:

$$p(S6) = p(S4 \wedge S5) + p(S4 \wedge S3) - p(S4)p(S1)p(S2) + (1-p(S4))p(S3) \quad (1)$$

### B. SC Design Obstacles

Despite the advantages that SC can bring forth, it holds its own shortcomings. Unintended correlations can still occur between input signals. This however, can be corrected through the use of better SNG designs than the basic one mentioned in this survey. Improved designs have found that by providing "pseudo-random" stochastic values than ones that are purely random, i.e. using to an extent determinism to generate the random binary streams, they provide better accuracy for SC. For example, a linear feedback shift register

Other means of providing accuracy in SC results invoke the utilization of correlation between inputs. By establishing a measure of correlation between stochastic numbers, a probabilistic transfer matrix (PTM) for a logic circuit provides details into whether inputs are correlated or uncorrelated. The PTM details the probability distributions for each input [2]. Therefore, if inputs X and Y have a probability of 0.5 if they are both 1 or both 0 but hold 0 probability if together, they hold opposite values, then one can declare that these inputs are highly correlated [3].

## III. STOCHASTIC COMPUTATION FOR RELIABILITY ANALYSIS

### A. Probabilistic Gate Models

Using the concept of stochastic numbers, reliability analysis of probabilistic circuits branched off from the original concept. Instead of carrying out computations from the resulting stochastic numbers, these probabilities are used to model and understand how reliable a VLSI circuit design is. Probabilistic gate models (PGMs) seek to relate the probabilities of an output to its input and error probabilities according to the function and malfunction of the gate itself [4]. Unlike SC, the inputs are considered deterministic and fault-free with no probability taken into account. An error rate,  $\epsilon$ , is used to represent von Neumann errors, also known as soft errors, occurring at the gate's output [5]. Thus a gate's output is based on whether the output of a gate was 1, expressed as  $p$ , caused by no error or that the output of a gate was 1 but due to error. This output probability is expressed in equation 2 below [4]:

$$Z = p(1-\epsilon) + (1-p)\epsilon \quad (2)$$

Equation 2 is in effect an XOR logic operation, where only one outcome can occur. PGMs are simplified to overcome the issue of reconvergent fanouts, which in this case occur in the form of reconvergent fanouts as exemplified in Figure 4. Using PGMs with dependent signals will result in an output probability that is an approximation of its actual value [10]. Therefore, to maintain accuracy, reconvergent fanouts are

decomposed. With the example provided in Figure 4, gates G1 and G2 are both dependent on input B, thus making the inputs for G1 and G2 dependent. To prevent statistical correlation between gates, B is split into two separate deterministic signals. A conditional probability for each of the inputs for fanout B is computed for both cases 0 and 1[5]. As a result, the probability of the output at G3 is based on the gate probabilities of G1 and G2 with no dependence between them.

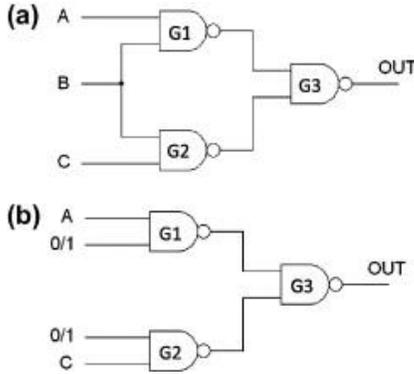


Fig. 4. a) Reconvergent fanout example b) Same circuit with reconvergent fanout decomposed for PGM [5]

As a PGM goes through a vast circuit design, all reconvergent fanouts are decomposed before their output reliabilities are calculated by partitioning the circuit. A flowchart of the PGM algorithm is displayed in Figure 4.

### B. Stochastic Computational Models

PGMs can be computationally expensive in reliability analysis, however. The decomposition time of reconvergent fanouts in a circuit design doubles per each fanout found[4]. To offset complex, time intensive reliability analysis, stochastic computational models (SCMs) use SC along with some aspects of PGMs to model circuit reliability in a simplified, tractable manner. Equation 2's modeling of probabilistic error at the output of a gate may be used in conjunction with stochastic circuits in the form of adding an XOR gate to a stochastic circuit's output with the error rate  $\epsilon$  as the secondary input. SCMs do not have to perform reconvergent fanout decomposition due to inputs now being bit streams caused by stochastic numbers as opposed to single, deterministic bits at the inputs. SCMs utilize the fact that SC maintains correlations at the inputs to the output while inputs that are exact, yet independent maintain their independence at the next level of signal propagation. The basic overview that the SCM approach takes to model reliability is by at first adding XOR gates at the output of each gate within a circuit as shown in Figure 6's example with the C17 circuit benchmark.

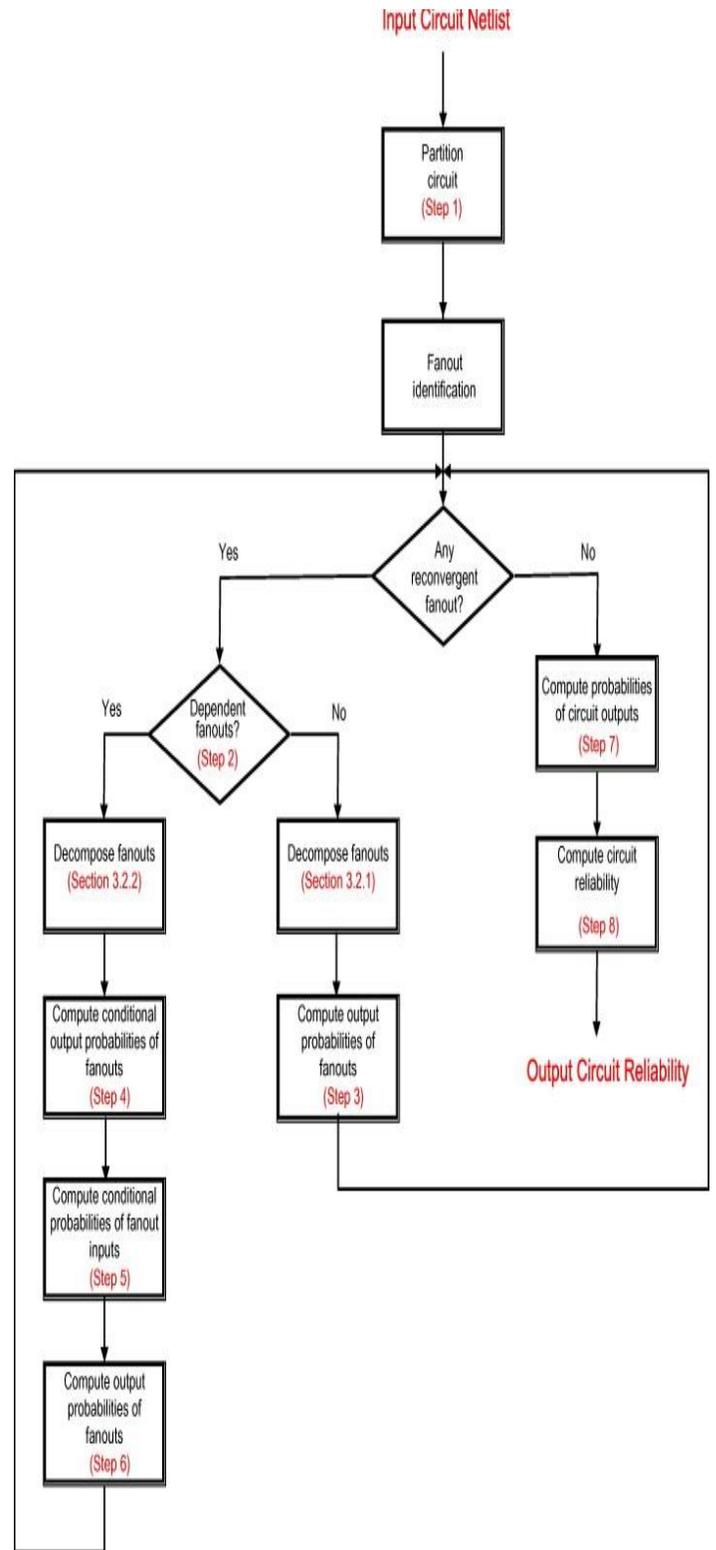


Fig. 5. Flowchart of PGM algorithm [5]

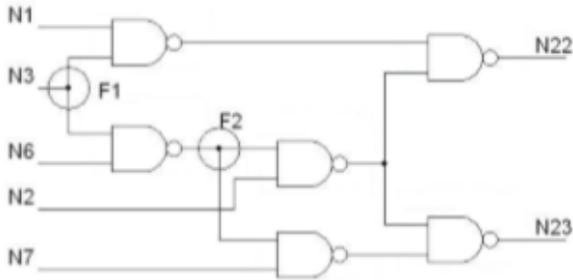


Fig. 6.a. C17 circuit[4]

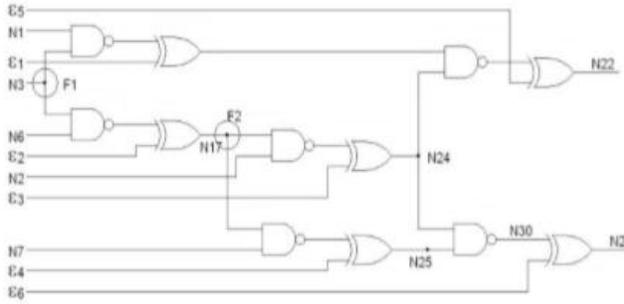


Fig. 6.b. C17 circuit as a SCM network with XOR gates implemented[4]

Upon the addition of XOR gates within the circuit, randomized bit streams based on the stochastic values for the inputs and error rates are propagated through the entire circuit. The inputs and error rates generated at the beginning of the circuit in Figure 6.b are generated through established error rates and randomly chosen stochastic input values. At each intermediate output, new bit streams are generated based on the resultant output probability. At the final outputs, in C17's case, outputs N22 and N23, the output probabilities detail the reliability of the entire circuit. The SCM approach is found to be more computationally efficient as it grows linearly with each gate included while offering slightly more accurate reliability measures than PGM[4].

However, the drawbacks to standard SC transfer over to the SCM approach. Computational time is extensive for the generations of these random bits and precision error is still a cause for concern. Stochastic numbers that are not exact to a bit flip in one of the inputs is still an issue for SCMs. However, precision can be attained by increasing the sequence length of random bits for a stochastic number as it determines the resolution of the reliability results[4]. For example, a sequence length of  $n$  produces a resolution of  $1/n$ , which dictates that probabilities less than  $1/n$  are not accurately represented. Thus, sequences of greater length prove beneficial toward representing a better range of probabilities as they provide more randomizing to occur.

#### IV. STOCHASTIC COMPUTATION FOR POWER-RELIABILITY

##### A. Error Statistics

Development in CMOS technology has resulted in a greater knowledge regarding low-power design and fault-tolerant computing. However, these metrics are treated

independently, while the idea of jointly optimizing power and reliability is a new area under development [7]. Statistical signal processing is one such application under this field. This approach follows a technique titled algorithmic noise tolerance (ANT) in which signal and error statistics are exploited in order to detect and correct intermittent errors in a nominal-case design while maintaining energy efficiency and fault-tolerance within a system [7]. Estimation theory using the notion of cost functions is taken into account to estimate the best signal out of various processing elements (PEs). The ANT technique takes a PE and an estimator PE, which is a lower-complexity block of the main PE that generates a statistical estimate of the main PE's output[8], and compares the output of the two PEs with hardware error and estimation error taken into account for the two blocks respectively. The ANT voter determines the final output based on a decision rule that has been adjusted for application parameters. Thus if the difference between the output of the main PE and the estimator PE are below some threshold (application specific), the output of the main PE is deemed reliable as the final output. Otherwise, the output of the estimator block is taken as a nominal-case approximation. ANT has been shown to maintain energy efficiency savings by up to 3 times, making it advantageous for certain statistical signal processing applications[6].

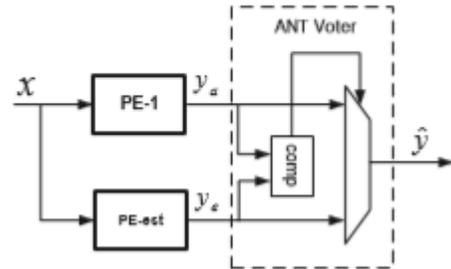


Fig. 7. ANT System [8]

##### B. Cost Functions used for Computation

A system cost function may be used to be modeled under two parameters,  $\theta$  and  $\theta(\bar{Y})$  represent the desired result of computation and the estimator  $\theta$  that operates on the input  $\bar{Y}$  respectively. The vector  $Y$ , may be modeled as the PEs which are the noisy measurements of the output,  $\theta$  [10]. Cost functions may vary depending on what design constraints may be taken into account, such as voltage levels or performance. These functions may also change to fit the application they seek to estimate. For example, for common general-purpose computations, a 0-1 cost function is best for the tolerating of errors at a certain threshold point as detailed in equation 3 [10]. Another cost function commonly used for signal processing applications is the squared-error cost function.

$$C[\hat{\theta}(\bar{Y}^*), \theta] = \begin{cases} 1, & |\hat{\theta}(\bar{Y}^*) - \theta| \geq \Delta \\ 0, & |\hat{\theta}(\bar{Y}^*) - \theta| < \Delta \end{cases}$$

Stochastic networked computation (SNC) decomposes an input signal into statistically similar PEs, called sensors, where each sensor performs its own sub-computation. After all the sub-computations of cost functions have been done for each of the PEs, a component labeled as the fusion block produces the final estimate of all the computations by "fusing" together all the components. A key drawback to SNC is that some applications where the input signal cannot be decomposed into statistically similar components that can generate similar outputs, thus making this approach rather limited. The unifying characteristic of this processing system is that robustness is achieved by means of post-processing the outputs of the PEs[10]. The fusion block architecture employs the sub-computations (with sensor and hardware error taken into account per sensor) into calculating a robust estimate via scale and location estimates. Further details into how the fusion block determines the final computation extends beyond the scope of this survey and may be found in [7]. Figure 7 details this approach.

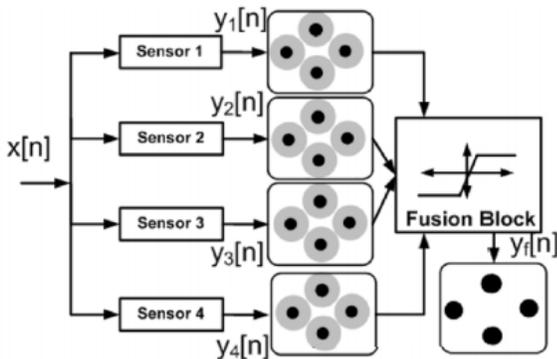


Fig. 8. SNC with centralized fusion block [7]

### C. Soft NMR

N-modular redundancy is a well-known fault-tolerant technique that employs design redundancy to deterministically detect which redundant system is processing errors by comparing it to its counterparts through a majority voter[8]. Using error statistics, a "soft" NMR approach can be taken by utilizing some aspects of the ANT technique described in Section IV, A. The general framework for soft NMR is detailed in Figure 9. The benefit that soft NMR offers is that the algorithms for the soft voter are dependent on the three factors outlined in Figure 9: metrics derived from the application, error and data statistics derived from the hardware itself and the detection techniques used to minimize the cost in making the decision for the PEs. This differs from NMR's voting element in which the voter is a general purpose voter that simply selects the majority rule on what the output should be based on the population of PEs. Due to this flexibility in how the soft voter algorithm may be outlined based on application constraints, only a general overview in what the soft voter achieves is discussed within this survey.

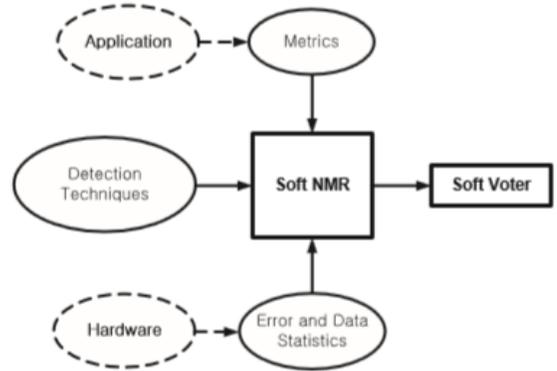


Fig.9. Soft NMR framework [8]

The voter element for soft NMR relies on two factors: an error model of the individual PEs and the performance metrics of the entire soft NMR system. The error model is based on data and error statistics that are obtained through behavioral and RTL simulations of the PEs. This leads to the disadvantage that the storing of statistical information can create hardware overhead. The use of a cost function for soft NMR takes into consideration the cost of choosing an element, with the possibility of it being the incorrect element taken into consideration. Therefore, the cost function is the sole element that determines the functionality of the soft voter and is closely tied to the application performance metric[8] by using Bayes decision rule as a means of averaging the cost over the random variables that a decision may undertake. Thus, the voter must determine the output that on average, is the best suited out of the PEs based on the metrics defined by the application while minimizing the Bayes risk. The soft voter contains a predetermined hypothesis set that consists of all decision rules that may be found from all the PEs in the set so that the final output is selected as a result from this set. This set is constrained to be limited and may either consist of the observation space being the outputs of all individual PEs in the system or it may be a value within a neighborhood. Architectures exist for soft NMR, with greater detail discussed in [8] regarding the error models for both NMR and soft NMR that detail how cost is minimized for metrics given in an application. Soft NMR provides an improvement in the fault-tolerance over that of NMR while being power-efficient due to its reduced footprint estimator. Soft NMR is advantageous due to the matter that it exploits error statistics based on the application while NMR is never adapted toward unique systems.

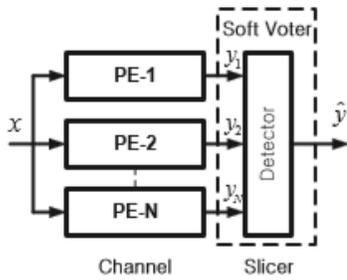


Fig. 10. Soft NMR system[11]

## V. TAXONOMY OF STOCHASTIC COMPUTATION

The approaches discussed within this survey all served to exploit some probabilistic aspect. SC may be viewed in two overall aspects of exploitation. Either, the low-level behavior of the circuit may be exploited through probabilistic means or the metrics outlined by an application may best be exploited probabilistically. Figure 11 details what may be seen as a scale of how SC can be seen and applied in varying degrees from high-level system design to low-level error-prone logic gates.

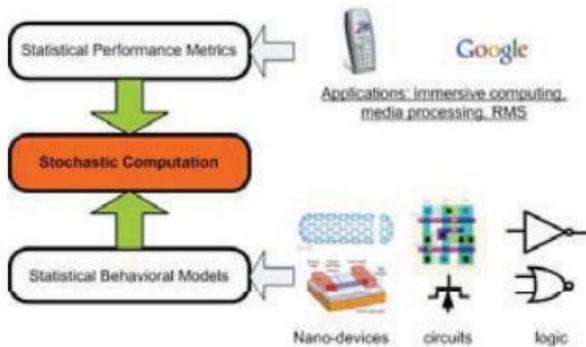


Fig.11. SC scale of applicability [11]

This survey juxtaposed three areas of SC: stochastic numbers, reliability design and statistical signal processing. Reliability design and statistical signal processing both share an emphasis with fault-tolerance. However, reliability design seeks to analyze the reliability of the stochastic system whereas statistical signal processing maintains fault-tolerance through the exploitation of statistical error models. Furthermore, all statistical signal processing approaches mentioned within this discussion sought to maintain power-

efficiency while PGMs and SCMs did not emphasize the power metric at all.

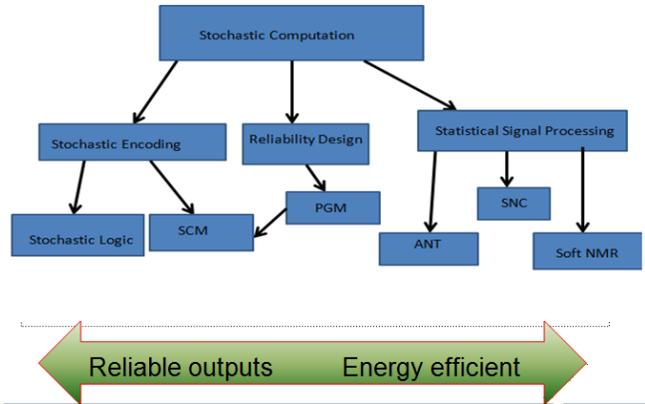


Fig. 12. Taxonomy of SC approaches

Table 1 compares and contrasts the approaches surveyed while providing examples of applications in which the approaches have been used. Common pitfalls continuously appear for the approaches listed, meaning they are still areas to overcome. For the approaches falling under the stochastic number taxonomy, they face the computational time versus accuracy tradeoff that was discussed intensively in the survey. In order to provide accurate results within these stochastic computation methods, the sequence lengths of the bits must be exponentially increased so that they are less susceptible to soft errors occurring in the bit stream. This leads to greater time of computation. Whereas many of the issues falling under the stochastic signal processing realm, involve being application limited, being resource hungry on the chip area or experiencing infeasibility in the decomposition of the sensors. However, statistical signal processing currently offers promising results with vastly improved robustness and energy-efficiency. Research must be ongoing in the overall SC field in order to improve these areas while discovering improved implementations.

Approach	Technique	Pitfalls	Benefits	Applicable Areas
<b>Stochastic Logic</b>	Uses stochastic numbers for logic functions	Computationally intensive	High degree of error tolerance	Image processing, LDPC[1]
<b>SCM</b>	Uses stochastic numbers with PGMS & XOR logic to compute output probabilities	Computationally intensive at design-time	Accurate modeling of reliability	Reliability evaluation of large-scale VLSI design[4]
<b>PGM</b>	Computes probabilities of outputs in circuit design	Computational time greatly increases due to reconvergent fanouts	Accurate modeling of reliability	Reliability evaluation of large-scale VLSI design[5]
<b>ANT</b>	Uses estimator and main PE to determine best output given error estimation	Application limited to areas where errors are detected and corrected to application specification [11]	Energy-efficient, sufficient error detection	FIR filters, Viterbi decoders[11]
<b>SNC</b>	PEs compute with hardware error and fuse outputs to determine final corrected output	Takes up large amount of area on computational fabric. Decomposition of signal into PEs where outputs are statistically similar can be difficult	Energy efficient. 800 times improvement in error detection [11]	CDMA PN-code acquisition system[7]
<b>Soft NMR</b>	Uses Bayesian decision to minimize cost in selecting functional component	Needs improvement in storing statistical information for large scale application	Robust and energy efficient	DCT Image coding[8]

Table 1-Approaches and their attributes

## VI. ASSESSMENT OF SC METHODS

After surveying the various notions that exist under the term of SC, a taxonomy was able to be established by allowing SC to fall under three terms as visualized in Figure 12. To evaluate modern-day feasibility in applying any of the approaches discussed, an assessment must be made of them by comparison to other non-SC approaches that seek to provide a reliable system from the same issue of error-prone CMOS logic. Of the SC taxonomy, we assess and evaluate stochastic encoding and stochastic signal processing against other viable alternatives in order to deem their effectiveness as a potential solution toward solving the technology scaling issue.

### A. Inexact Arithmetic and Stochastic Encoding

Approximate computing (AC) relies on probabilistic computing, meaning that it exploits the probabilistic behavior of the circuit design in order to determine how to tolerate error from the hardware. This differs from SC in that it does not involve assumptions on the stochastic nature of any underlying processes implementing the system (such as stochastic encoding) but instead utilizes statistical properties of data and algorithms to trade quality for energy reduction [13]. Therefore AC procures deterministic designs that allow for imprecision within their resultant outputs. AC is similar to stochastic encoding in that approximate computations may be processed within a tolerated boundary of imprecision. Furthermore, like SC, various AC techniques exist from high-level (programming languages) to low-level (transistor/RTL).

One such AC technique is the implementation of approximate arithmetic circuits that alter some aspect of the circuit with the goal of allowing imprecision kept in mind. This may be done so through creating adders that ignore lesser insignificant bits, or carries in parts of the adder, in addition to transistor truncation. AC performs best under specific applications that SC is best suited for as well such as image and signal processing. Both inexact arithmetic units and stochastic encoding may be applied toward DCT applications for image processing [1],[13]. Currently, there exists no experimental results comparing SC implementations with AC in order to determine which approach best offers power efficiency and reliability. However, due to AC's ability to alter and minimize resources, it may offer the advantage of lesser area and energy overhead for some cases when compared to stochastic circuits for some cases. An opposing case would be the multiplier stochastic circuit example in Figure 1 in comparison with an approximate multiplier circuit that ignores bits in the partial products [13].

### B. hCED and Soft NMR

Concurrent error detection (CED) is similar to the NMR approach in that error is deterministically found by comparing redundant systems as shown in Figure 13.a. Functional elements (FEs) are essentially redundant functionally identical implementations. The same inputs are delivered to the FEs with their respective outputs compared to one another through the discrepancy detector to yield a measure of fitness between the two FEs. The drawbacks to standard CED however is that power consumption runs high in addition to the hardware overhead cost in implementing multiple instances of the same FE. As a means of promoting power efficiency with less hardware overhead, heterogeneous CED (hCED) is a proposed method of detecting errors between an instance of an FE and a secondary "checker" unit. The checker is not an exact replica of the FE but rather a reduced version of the FE with some of its capabilities marginalized such as reduced throughput[12]. This is done to offer the prospect of error detection with conservation of on board resources and energy. The checker is essentially an inexact logic function which acts to predict the next output. hCED, overall, is utilizes a consensus between the

two diverse yet functionally similar circuits. The inclusion of a checker along with a main FE to compare against is reminiscent of the ANT technique and soft NMR previously discussed in Sections IV,A and C respectively.

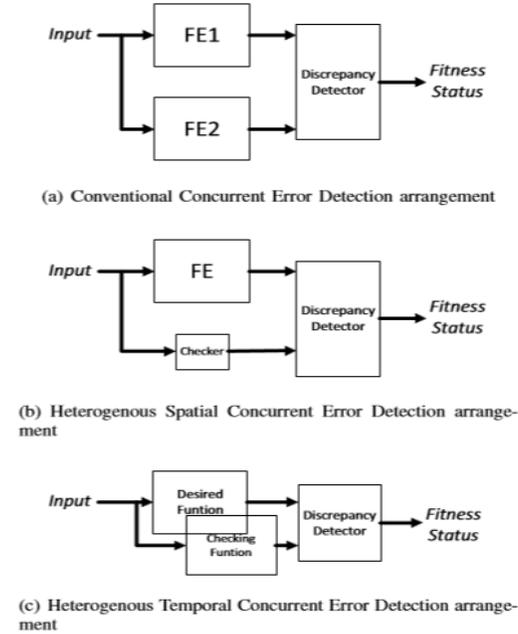


Fig. 8. CED implementations [12]

There are two approaches to hCED, spatial and temporal. In spatial hCED, both primary and checker circuits are statically-loaded with no reconfiguration occurring during runtime. hCED is configured to evaluate error detection for any linear transformation that is pipelineable such that the checker estimates what the current output is based on past evaluations by monitoring the difference values. If the output function changes too abruptly as observed by the checker, an error is deemed to have been found. Temporal hCED interchanges the checker and FE in resources of the hardware fabric while following the same estimation methodology as proposed in the spatial approach.

When used for linear functions, hCED has been shown to consume less energy than the general solutions supplied by ANT or soft NMR as it is designed for such applications. hCED may be seen as an alternative to such stochastic signal processing techniques due to the fact that it follows a linear scheme of past values to determine how to estimate future, upcoming values. Thus, while ANT and soft NMR offer general statistical solutions based on metrics found from the application, a simpler solution for linear transformations will always procure the more energy efficient result. The simplicity of the checker function itself can also be factored into possibly being less hardware intensive than an estimator model or voting element that requires storage and analysis of statistical information.

## VII. FUTURE DIRECTION OF SC

As seen from the assessment of certain SC approaches, SC supports the notion that reliable systems can be designed with unreliable components, but that application requirements need to be brought into the design process[11]. It can also provide insights into the probabilistic aspects of other computing technologies, such as conventional circuits subject to random faults, as discussed in stochastic encoding and reliability analysis, or even quantum computing [1]. However, despite being application-dependent, there are a few issues that SC must overcome for all of its approaches. These obstacles include engineering design and synthesis techniques that produce statistical models that ensure graceful increase in error rates, obtaining desired error rates, and ensuring independent, uncorrelated errors [11].

Thus, it is reasonable to conclude that SC, in the near future, may be used in a variety of applications such as embedded systems, signal processing, circuit design and more as offering a reliable, consistent solution to the technology scaling issue. The long computation times found within stochastic encoding coupled with the independent, uncorrelated inputs issue described, leaves this approach to have much to be improved upon before becoming widely used for application specific uses and embedded systems [1]. With regards to reliability analysis, it offers at design time, a system with the probabilities of error occurring already modeled. However, it does not overcome erroneous behavior during runtime, making this the unlikely approach to become widely used in the future. Of the approaches discussed, it seems that statistical signal processing techniques may offer the best direction toward first implementing SC within industry. This prediction is due to the successful experimental results relayed by ANT, SNC, and soft NMR [7],[8] where energy efficiency and system robustness are attained to successful levels of implementation.

## VIII. CONCLUSION

SC, once being cast aside after the exponential growth of binary circuits, is now experiencing a revival in CMOS research. This is greatly due to the fact that the scaling of CMOS technology is beginning to experience unreliability in its design from manufacturing, temperature and voltage variations. Rather than finding methods of avoiding error, SC embraces the fact that errors can occur to its benefit. By exploiting the probability of error, SC can perform computations, analyze the reliability of unreliable circuit designs, and perform signal processing from error-prone signals. Each of the approaches mentioned, perform best with regards to reliability and energy efficiency when under application specific implementations. Overall, we can conclude that this promising field currently offers many implementations that have offered results that demonstrate system functionality with aberrant circuit design and logic.

## REFERENCES

- [1] Armin Alaghi and John P. Hayes. 2013. Survey of Stochastic Computing. *ACM Trans. Embed. Comput. Syst.* 12, 2s, Article 92 (May 2013), 19 pages. DOI=10.1145/2465787.2465794
- [2] A. Paler, A. Alaghi, I. Polian, and J. Hayes. 2011. Tomographic Testing and Validation of Probabilistic Circuits. *European Test. Symposium 16th IEEE*. pp.63-68. DOI=10.1109/ETS.2011.43
- [3] A. Alaghi and J. P. Hayes. 2013. Exploiting Correlation in Stochastic Circuit Design. 2013 IEEE 31st International Conference on Computer Design. pp.39-46. DOI=10.1109/ICCD.2013.6657023
- [4] H. Chen and J. Han. 2010. Stochastic Computational Models for Accurate Reliability Evaluation of Logic Circuits. *Great Lakes Symposium on VLSI*. pp.61-66. DOI=10.1145/1785481.1785497
- [5] J. Han, H. Chen, E. Boykin, and J. Fortes. 2010. Reliability evaluation of logic circuits using probabilistic gate models. *Microelectronics Reliability*. pp.486-476. DOI=10.1016/j.icorel.2010.07.154
- [6] N. Shanbhag. Stochastic Computing for Embedded Platforms. 2012. Available:<http://ucoweb01.intel-research.net/wp-content/uploads/2012/06/ISTC-seminar-05-31-12.pdf>
- [7] Varatkar, G.V.; Narayanan, S.; Shanbhag, N.R.; Jones, D.L., "Stochastic Networked Computation," *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on , vol.18, no.10, pp.1421,1432, Oct. 2010 DOI= 10.1109/TVLSI.2009.202467
- [8] E. Kim and N. R. Shanbhag, 2012. Soft N-modular redundancy. *IEEE Transactions on Computers*, vol. 61, no. 3. pp. 323-336. DOI=10.1109/TC.2010.253
- [9] E. Taylor and J. Fortes. 2007. Device Variability Impact on Logic Gate Failure Rates. *Government Microcircuit Application and Critical Technology Conference (GOMACTech-07)*. Available: <https://www.acis.ufl.edu/tags/device-variability-impact-logic-gate-failure-rates>
- [10] S. Narayanan, G. Varatkar, D. L. Jones, and N. R. Shanbhag, 2010. "Computation as estimation: A general framework for robustness and energy-efficiency in SoCs," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4416-4421.
- [11] Naresh R. Shanbhag, Rami A. Abdallah, Rakesh Kumar, and Douglas L. Jones. 2010. Stochastic Computation. 47th ACM/IEEE Design Automation Conference. pp.859-864.
- [12] N. Imran, R. F. Demara, "Heterogeneous Concurrent Error Detection (hCED) Based on Output Anticipation", in proceedings of the *International Conference on Reconfigurable Computing and FPGAs (ReConFig 2011)*, pp.61-66, Nov. 30 2011-Dec. 2 2011.
- [13] Jie Han and Michael Orshansky. 2013. Approximate Computing: An Emerging Paradigm for Energy-Efficient Design. *Proceedings of the 18th IEEE European Test Symposium*, Avignon, France. pp.1-6.