Energy-Efficiency and Accuracy of Stochastic Computing Circuits in Emerging Technologies

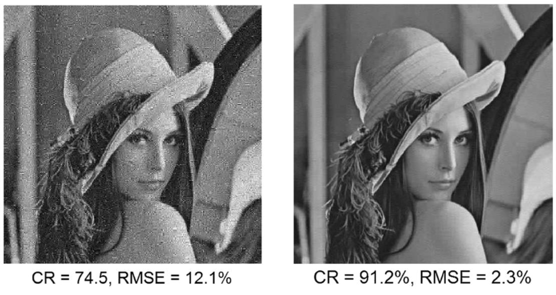
Bert Moons, *Student Member, IEEE*, and Marian Verhelst, *Senior Member, IEEE*

***Abstract*—The continued scaling of feature sizes in integrated circuit technology leads to more uncertainty and unreliability in circuit behavior. Maintaining the paradigm of deterministic Boolean computing therefore becomes increasingly challenging. Stochastic computing (SC) processes digital data in the form of long pseudo-random bit-streams denoting probabilities and is therefore less vulnerable to uncertainty. When transient circuit variations are present, SC greatly outperforms classical binary implementations. Under these circumstances, it is impossible for binary systems to achieve arbitrarily low error rates, while SC can still trade-off precision for energy by using longer bit-streams. This makes the technique a valuable alternative to binary logic in emerging technologies with high inherent transient uncertainty. This paper assesses the feasibility of multi-stage SC and discusses energy and accuracy considerations in SC design. First, the basics of SC-circuit design are discussed. Second, we investigate three different sources of noise or uncertainty and assess their impact on SC accuracy. Third, we propose a methodological design strategy to evaluate the accuracy of general, multi-stage SC systems. The validity of this new approach is illustrated through the design of a 1D-DCT stochastic circuit, as part of a JPEG compression accelerator. Our analysis shows multi-stage stochastic computing requires very long word lengths to achieve high accuracy, resulting in low energy efficiency. Exploiting stochastic computing's transient error tolerance in emerging technologies will thus have a high energy cost.**

***Index Terms*—Accuracy, energy, modelling, multi-stage, stochastic computing (SC).**

I. Introduction

DIGITAL electronics has always relied on error-less circuit operation. Precise Boolean functionality, defined in a deterministic logical layer, is translated into a physical layer that produces voltages. These can be interpreted as the needed exact logic values. This abstraction has been successful, but becomes ever more costly in emerging technologies. All forms of noise and uncertainty in the physical layer have to be compensated for through more complex and energy-hungry designs with large design margins. Recently, new research is focussing on novel ways to handle device uncertainty in a more efficient way. A very promising class of techniques, labeled "Stochastic Computation," exploits probability theory to deal with variations. Shanbhag *et al.* give an overview of different techniques [1]. Stochastic Computing (SC), a computational technique introduced by Gaines [2], [3] processes data in the form of digitized probabilities. Von Neumann [4] also looked into probabilistic logic for unreliable components. SC has three main advantages over conventional computing approaches.



CR = 74.5, RMSE = 12.1%
(a)

CR = 91.2%, RMSE = 2.3%
(b)

Fig. 1. JPEG compression using a (a) binary and (b) stochastic ($L = 2^{16}$) DCT under transient variations with bit-flip rate $p_t = 1e - 3$. CR and RMSE of the ideal picture are used as formal performance measures.

First SC's main advantage is that its probabilistic aspect makes it inherently tolerant to soft transient errors (such as bit-flips and supply voltage ringing) and robust against spatial variations. Due to this error tolerance, the logic type seems a good alternative to binary computing in emerging technologies suffering from high uncertainty. Fig. 1 illustrates robustness to transient circuit variations. For this example, we have implemented a DCT-block as part of JPEG compressor, both in a binary and in a stochastic way. Both systems are subjected to bit-flips at a rate p_t of $1e - 3$. The accuracy degradation of the binary implementation is striking, while SC can still achieve almost perfect results. However, in order to exploit SC's extraordinary transient error tolerance in multi-staged circuits, there will be an energy cost, since long bit-streams are needed to minimize the effect of inherent faults.

Second, SC uses very low complexity building blocks, making it suitable for massively parallel processing.

Third, there is the possibility to create logic with scalable, progressive precision. Shortened computation can already provide an early estimate of a target value. This concept allows trading-off precision for energy at run-time, an advantage that can be well exploited in emerging ultra-low energy applications [5].

Manuscript received February 25, 2014; revised July 18, 2014 and August 28, 2014; accepted September 07, 2014. Date of publication October 15, 2014; date of current version December 09, 2014. This paper was recommended by Guest Editor S. Bhunia.

The authors are with the Microelectronics and Sensors Division (MICAS), Department of Electrical Engineering (ESAT), KU Leuven, 3001 Leuven, Belgium (e-mail: bert.moons@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2014.2361070

Although SC has been known for decades, very few physical implementations have been made. Recently SC has been used in LDPC decoding [6], in basic image processing systems [7], [8], in fault-tree analyses [9], [10], and in filters [11], [12]. Alaghi and Hayes [13] and Qian and Riedel [14], [15] have proposed synthesis approaches for classes of combinational circuits, hereby enabling a formal approach to generate complex and in some cases reconfigurable arithmetic functions. Previous research however only considers SC circuits with only a few (<3) stages.

This work focusses on multi-stage SC. The paper's main novelties are the following. We link the advantages of SC to emerging technologies. We present a systematic breakdown of the three different types of errors in SC and analyse the inherent signal loss in multiple stages. From this approach we derive a formal design methodology for multi-stage circuits. We use this methodology to implement the first complex 1D-DCT circuit and use it as a basic block in a JPEG encoder. This circuit is used to assess SC's accuracy in multi-stage circuits. Finally we show multi-stage stochastic computing circuits require very long data streams to achieve high accuracy, resulting in high energy dissipation. We quantify the system level energy consumption, both in a 40 nm CMOS and a 26 nm TFET technology.

This paper is organized as follows. Section II gives an overview of stochastic numbers, arithmetic blocks and low level design of stochastic circuits. We link the usage of SC to circuit design in emerging technologies. Section III discusses different sources of variation and noise in digital systems: inherent, spatial and transient uncertainty. This section also analyses the performance in terms of accuracy of a single stage SC multiplier and compares it to an equivalent binary implementation, both under influence of different noise sources. Section IV discusses the reasons for decreasing signal power in multi-stage SC. This meticulous analysis allows proposing a methodological strategy in Section V, which can be used to design new SC systems and evaluate the accuracy of existing circuits. Section VI concludes this work.

II. Stochastic Computing

A. Basic Theory

Stochastic numbers (SN) are bit-streams containing $N1$ 1's and $N0$ 0's denoting the unipolar (UP) number $p = N1/(N1+N0)$. Since p will always lie in the real-number interval [0,1], it can be interpreted as the probability that the bit-stream X outputs a 1 at X_i: $p = P(X_i = 1)$. A bipolar (BP) interpretation of the bit-stream is possible by transforming p onto the [-1,1] interval ($s = 2p - 1$). The precision of the stochastic number is determined by the length of the bit-stream. A bit-stream of ($L = 256 = 2^8$) bits has a maximal theoretical precision of eight binary bits. The most basic example of stochastic computing is given in Fig. 2. This figure illustrates stochastic UP multiplication. Multiplication in the UP format can be implemented with an AND-gate. The AND-gate of Fig. 2 multiplies the sequential bit-stream $x = 0{,}1{,}1{,}0{,}1{,}0$ with stream $y = 0{,}0{,}0{,}1{,}1{,}0$. Stream x represents real number 3/6 since three out of six bits are a logical 1. The result of this computation is 1/6, which is obviously correct. If streams x and y are correlated, the result could deviate.

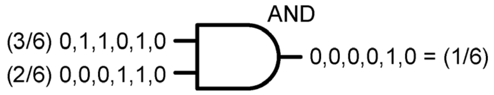


Fig. 2. Example of UP stochastic multiplication with an AND-gate.

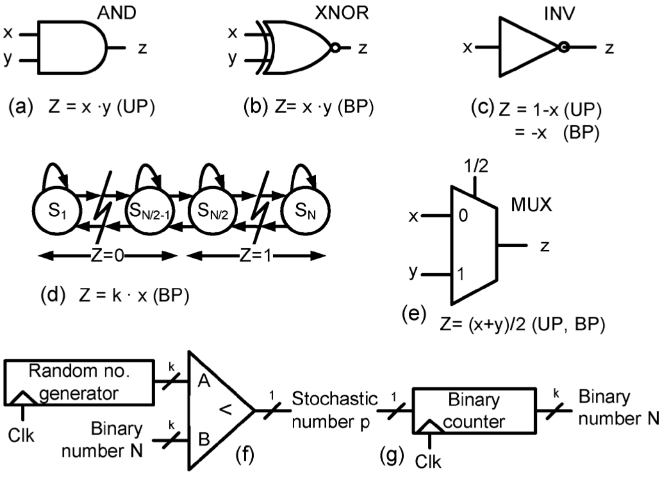


Fig. 3. Examples of basic SC arithmetic gates. (a) Bipolar multiplier. (b) FSM-based linear gain function. (c) Scaled adder. (d) Binary-to-stochastic converter. (e) Stochastic-to-binary converter.

A more typical SC system exists out of a binary-to-stochastic (BTS) conversion unit, stochastic arithmetic and a stochastic-to-binary (STB) converter [Fig. 3(f) and (g)]. The BTS unit can be easily implemented using LFSR pseudo random number generators [16]–[18]. These can be proven to generate near-exact approximations of the wanted binary input value. For conversion from SC to binary a simple counter suffices. The used stochastic arithmetic gate depends on the number interpretation. Multiplication can be done by using an AND-gate in the UP format [Fig. 2 and Fig. 3(a)], or an XNOR in the BP format [Fig. 3(b)]. Scaled addition can be implemented using a MUX-gate in both cases [16] by driving the selector input with a stochastic number p [Fig. 3(e)]. The resulting output will be $p_x \times (p) + p_y \times (1-p)$. Using $p = 1/2$ thus outputs $(p_x + p_y)/2$. The INV-gate implements $(1 - p)$ in the UP and $(-s)$ in the BP format [Fig. 3(c)]. More complex gates such as comparators and linear gain functions are nontrivial in SC (in contrast to binary logic) and can be implemented using the synthesis approaches from [13] and [14] or by using an FSM-based system [Fig. 3(d)] [19]. Basic blocks for stochastic division and stochastic square roots have also been presented [20].

B. Circuit Level Aspects and Comparison With Binary

Classical binary systems can be pipelined to increase throughput. This is not possible in SC systems, since all operations are single stage and the computing style is inherently sequential. This is a big disadvantage of SC which makes it difficult to achieve high throughput. The energy and accuracy of SC systems can be compared to standard binary for the same overall delay. The total delay D of a multi-stage binary and SC system can be summarized as

$$D_{\text{bin}} = \frac{S}{f_{\text{bin}}} \tag{1}$$

$$D_{\text{SC}} = \frac{L}{f_{\text{SC}}P} + \frac{(S-1)}{f_{\text{SC}}} \tag{2}$$

where f_{SC} is the SC clock frequency, f_{bin} is the binary clock frequency, S is the number of stages, L is the number length, and P is the degree of parallelization. To reduce energy and total delay, SC arithmetic functions can be parallelized with a certain factor P. This implies multiplying the amount of gates by P. Each gate will then compute a shorter bit-stream of length L/P, hereby reducing the total time for computation. This can be done with little overhead due to the typically low gate area.

The overall delay increases linearly with S, both in the binary and the SC case. However, since L/P will dominate over $S-1$ in most cases, the total SC delay will not be a strong function of S. Due to the short data paths of SC gates, f_{SC} can be much higher than f_{bin}. Therefore it is possible to achieve a low overall delay D_{SC}, even if very long L are used. It is clear from previous equations, that equal delay can be reached after fewer stages, if the parallelization degree P increases or if the frequency ratio $f_{\text{SC}}/f_{\text{bin}}$ increases. However, P is limited by area constraints, since higher P means an increased number of used gates. f_{SC} is limited by energy considerations. If f_{SC} is required to be very high, the system's minimal supply voltage will have to increase. This results in higher energy dissipation. The energy dissipation in a SC-gate can always be modelled as

$$E_{\text{SC}} = k(f_{\text{SC}}, \alpha, V, C) \cdot L \tag{3}$$

where k is the energy per bit-operation, a function of the required SC operation frequency f_{SC}, supply voltage V, circuit activity α, and the switching capacitance C. L is the stochastic number length. A SC system uses long bit-streams to achieve high accuracy (Section III-A) and thus requires high clock-speeds to reach a certain computing delay. Despite this need of high f_{SC}, the SC supply voltage V can still be well below the nominal voltage. This is due to the usage of extremely short data paths. The parallelization degree P has no direct influence on the energy dissipation, it only influences delay and area.

For a real design using SC-logic we assume the overall system delay to be specified by the application. The number of stages is also fixed and determined by the circuit architecture. The stream length L will be determined by the required accuracy as explained below. P minimizes SC delay within the area of the equivalent binary system. f_{SC} is then chosen in order to minimize energy. If it is too high, the energy per bit-operation k will increase, leading to a low global energy-efficiency.

C. Stochastic Computing in Emerging Technologies

SC's tolerance to transient errors makes it an interesting logic type for emerging technologies. Current research indicates future integrated circuits will suffer from reduced noise margins or cycle-to-cycle variations, making digital systems more sensitive to random telegraph noise (RTN) or radiation. All these effects can result in transient errors. We give three examples from literature.

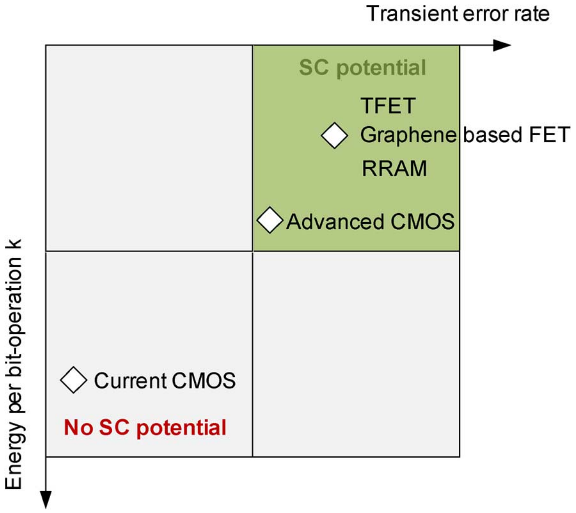


Fig. 4. Integrated circuit technologies in the Energy-Transient Error Rate space. Current CMOS is not suitable for SC due to its high energy per bit-operation k (Section III, Section V-C). Emerging technologies, with low k and high transient error rate, can benefit from stochastic computing.

First, sub-22 nm CMOS suffers from increased voltage scaling. Using lower supply voltages reduces noise margins and increases the relative effect of resistive and inductive supply drops [21]. In general, the soft error rate increases when supply voltage is lowered [22].

Second, resistive RAM technologies such as HfO_2 RRAM show intrinsic cycle-to-cycle switching variability [23]. Since RRAM allows creating nonvolatile flip-flops [24], these switching variations might not only cause transient errors in memory, but also in arithmetic circuits.

Third, new narrow band-gap devices such as tunnel-FETs (TFET) or graphene nanoribbon FETs [25] also suffer from reduced noise margins. They are attractive for channel replacement due to their mobility enhancement compared to silicon, but are more sensitive to RTN because of their narrow bandgaps [26]–[28]. Simulations on a ring oscillator show graphene nanoribbon FETs have a 25×–144× advantage in EDP compared to silicon implementations [25]. Modelling from [29] and [30] shows the delay in CMOS increases about two orders of magnitude more than the delay in TFETs for the same voltage scaling $(0.6 - 0.2\ \text{V})$. TFET thus has high performance at very low power. This allows better exploitation of SC's ultra-short data paths; voltage can be scaled further, while maintaining performance. Fig. 4 illustrates that the reviewed emerging technologies can be a good match with SC.

D. Multi-Stage Stochastic Circuits

SC is known for its tolerance of soft transient circuit variations. Previous work has proven circuits with a few stages can achieve very good accuracy and energy-efficiency. The implementation of a stochastic edge detector in [7] is an example of SC outperforming a binary implementation. But, there has been no previous research on more general multi-stage circuits. Due to SC's randomness and the computation of correlated bitstreams, SC is inherently inaccurate, a problem which becomes more stringent in systems with more stages. There are two combined effects in multi-stage SC circuits. First, inherent noise in stochastic outputs is binomially distributed. This noise is high, compared to binary, where inherent inaccuracy is caused by quantization errors. Noise due to spatial and transient circuit variations also exists, but has little impact on SC's accuracy. Second, stochastic signal power tends to decrease after multiple stages. The combination of these two effects (high noise and decreasing signal power) leads to a decreasing SNR in multi-stage circuits, and thus to a decreasing accuracy. We will elaborate on these effects in Sections III–V.

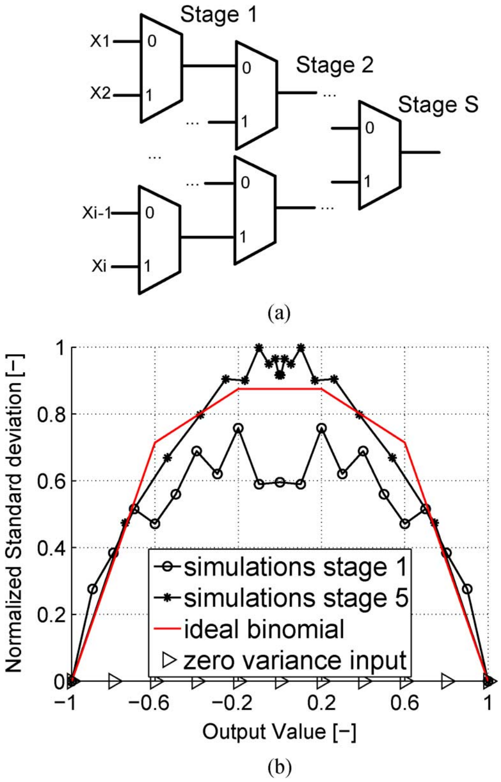
III. Noise in Stochastic Computing

There are three major sources of errors in advanced technology digital computations: errors inherent to the used logic type (type I), errors due to static spatial circuit variations (type II), and errors due to dynamic transient circuit variations (type III). The three types of errors discussed are independent random processes, their effects are additive and should be combined to evaluate global performance.

Examples of type I errors are quantization faults in the binary logic type, and faults due to stochastic correlation in the SC logic type. Interestingly, Alaghi and Hayes [31] try to exploit this correlation to create new ad-hoc stochastic functions. Type II errors stem from spatial circuit variations. These are variations that are random in space, but fixed in time, such as random doping fluctuations or any kind of inter- or intra-die variations. These are already omnipresent in current transistor technologies and will become more important in future technologies. Since the critical path of a SC multiplier is fixed and very short (in contrast to the critical path of a binary multiplier), it is expected that the influence of spatial variations on SC output accuracy is limited (see Section III-D2). Type III errors stem from fast transient circuit variations. These are variations that are random in time and space, such as random bit-flips, radiation effects, or supply-voltage ringing. In current technologies spatial variations are still the dominant source of uncertainty, but transient variations are becoming more important in more advanced CMOS or in emerging post-Si technologies when dopant levels and voltage headroom further decreases. As the following paragraphs will indicate, SC's probabilistic aspect makes it less vulnerable to this type of variations than binary systems.

A. Type I: Inherent Noise

Practical implementations of SC circuits use LFSR random number generators for their binary to stochastic (BTS) transformation. These generate numbers that can be guaranteed to be near-exact [16]. The variance of any SN after the BTS-generator will be zero. However, uncontrollable correlation between two computed stochastic numbers will still randomize the SN's in SC circuitry. After several stages, the zero variance LFSR-generated number will converge into a binomial distributed number.



(a)

(b)

Fig. 5. SC output values are binomially distributed after several stages. (a) Multi-stage SC test setup. (b) Computed standard deviation after several stages.

To illustrate this, we explicitly simulate variance propagation. Fig. 5(a) shows our set-up existing out of S stages of SC adder-circuits. Note that this test-circuit represents the same circuit as path I in the DCT implementation (Fig. 9), in which S equals three.

Fig. 5(b) shows the variance at the outputs of the different stages and compares it with the variance of a binomial distributed process. This shows that the binomial distribution is indeed a good approximation and suitable for first-order accuracy analysis, even when pseudo-random LFSR-generators are used. If ideal random number generators are used, the binomial approximation will be exact. Ma [32] discusses the modelling of inherent noise in SC as a hypergeometric process. This is however not needed for our first-order analysis.

Using the binomial model, the variance of a stochastic number is then a function of its UP value p and the SN length L

$$\sigma_{UP}^2 = \frac{\sigma_{BP}^2}{4} = \frac{p(1-p)}{L}. \tag{4}$$

Note that σ^2 is maximal where $p = 0.5$ and $s = 0$. The noise is thus largest where BP numbers have the lowest signal power. The only way to reduce this variance is by using longer stochastic numbers. If uniformly distributed input values are assumed, the mean noise power across all possible input values can be computed as

$$\sigma_{\text{mean-UP}}^2 = \int_0^1 \frac{p(1-p)}{L} \mathrm{d}p = \frac{1}{6L}. \tag{5}$$

If basic stochastic blocks (AND, MUX, XOR, XNOR, INV) are used, the noise remains binomial due to correlation effects. If FSM-based constant-multiplicand [19] (with multiplicand $c > 1$) blocks are used, the variance scales accordingly.

For example, a constant multiplication gives $\sigma_{\text{out}} = c \cdot \sigma_{in}$ if $c > 1$. Using this block thus leads to noise that is even higher than binomial.

These result should be compared to the inherent inaccuracy in binary systems

$$\sigma^2_{\text{mean-binary}} = \frac{\text{LSB}^2}{12} = \frac{1}{12 \cdot 2^{2n}} \tag{6}$$

where n is the binary word length. From this first-order estimation, it already becomes clear that very long bit-streams are needed to achieve the same absolute noise power

$$\sigma^2_{\text{mean-UP}} = \sigma^2_{\text{mean-binary}} \iff L = 2^{2n+1}. \tag{7}$$

B. Type II: Noise Due to Spatial Circuit Variations

Spatial circuit variations such as random doping fluctuations may also cause inaccuracy in digital systems. Circuit designers cope with these uncertainties by introducing extra static design margins in the form of higher supply voltages or conservative layouts. Designs in technologies with high spatial variations therefore typically have a low energy-efficiency. In binary systems, faults due to spatial variations should always be prevented, since these will typically lead to timing errors on the MSB critical path. Faults on these paths are large in magnitude and therefore result in a high root mean square error (RMSE).

In SC, there is a possibility to trade-off energy, area and precision. Because of SC's sequential nature, errors due to spatial variations will be small and on the order of LSB. A limited introduction of errors due to these variations may be tolerable if the associated energy-gain, due to smaller design margins such as lower supply voltage, is sufficient. Consider a single SC multiplier, computing a SN of length L. The output accuracy of this single AND-gate is determined by variations which are randomly distributed. But after production it static and fixed in time and space. The resulting output value is a sample of the distribution $N_1(\mu_L, \sigma_L)$. μ_L is the expected mean deviation of the ideal value p, σ_L is its standard deviation. By parallellizing this computation by a factor $P = 2$, a second AND-gate is introduced. Both gates now compute SN's of $L_2 = L/2$. The resulting output is determined by *two* samples of the random distribution

$$N_2 = N(\mu_{L2}, \sigma_{L2}) + N(\mu_{L2}, \sigma_{L2}) \tag{8}$$

$$N_2 = N\left(\mu_L, \frac{\sigma_L}{\sqrt{2}}\right). \tag{9}$$

If P gates are used, the distribution of the output value becomes

$$N_P = N\left(\mu_L, \frac{\sigma_L}{\sqrt{P}}\right). \tag{10}$$

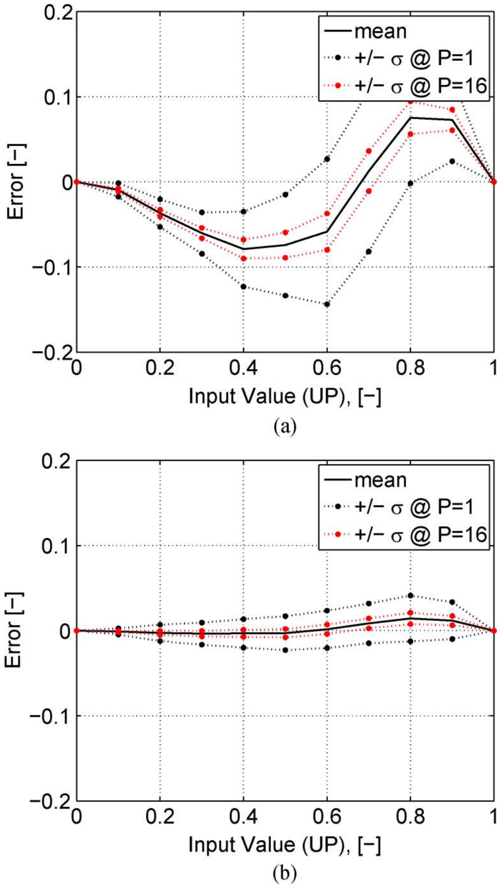


Fig. 6. Mean errors and standard deviations in function of the UP output value of a SC multiplier. (a) 60% energy gain and (b) 30% energy gain due to voltage overscaling. Only the standard deviation is a function of P, the mean is not. Mean RMSE in (a) is 7.5% when $P = 1$ and 5.3% when $P = \infty$. Implementation (b) has respectively $\text{RMSE} = 1.8\%$ and $\text{RMSE} = 0.65\%$.

Parallellizing can thus effectively reduce the RMSE due to spatial variations. However, there will always be an upper limit on RMSE, determined by μ_L. In these equations, μ_L and σ_L are determined by the amount of spatial variations (fixed by the supply voltage and circuit technology) and by the SN-value. Fig. 6 shows the errors due to spatial variations in a SC multiplier (input^2) under different circumstances and for different P as a function of the input value. These plots are generated using Monte-Carlo simulations in Spice. Fig. 6(a) and (b) shows the errors when the supply voltage V is dropped 33% and 16.5% from the error-less voltage. This is equivalent to a drop in energy dissipation of respectively 60% and 30%. Observe the zero mean at a 0.7 input, resulting in $0.7^2 \approx 0.5$. At this input value, the AND-gate ideally outputs as many zeros as ones. The number of too-slow pull-ups will then equal the number of too-slow pull-downs, resulting in a zero mean deviation. Lower input values lead to a negative mean deviation, since the number of zero's in these output streams is larger than the number of ones. Therefore, there will be more failed zero to one transitions due to too slow pull-ups, resulting in a negative error. Lower supply voltages decrease the energy dissipation, but increase RMSE.

Since spatial variations are fixed in time, a single AND-gate will always make the same errors after production. Faults thus become repetitive and deterministic. It is therefore better to tune out spatial variations completely by using higher supply-voltages.

C. *Type III: Noise Due to Transient Circuit Variations*

Transient circuit variations such as random bit-flips, cycle-to-cyle variations, radiation effects or supply-voltage ringing may cause severe faults in digital circuits. In current technologies, spatial variations are still the dominant source of uncertainty, but transient variations will become more important in the future (Section II-C). System simulations allow assessing these type of errors. In SC they can be efficiently modelled by using an XOR-gate on every logical output node. Input stream p_a will be distorted at rate p_t, where p_t will be very low. The resulting p_{out} equals $xor(p_a, p_t)$. If a bit of bit-stream p_t equals 1, the corresponding bit of stream p_a will invert. It is easy to understand that SC will not suffer greatly from transient circuit variations. If this problem is considered in the bipolar format, the XOR computes a multiplication and inversion (see Section II). If we transform these unipolar numbers to their equivalent bipolar representation, the outcome of the BP number $s_a = 2p_a - 1$ distorted at a rate $s_t = 2p_t - 1$, with $p_t = 1e - 3$ can be easily computed

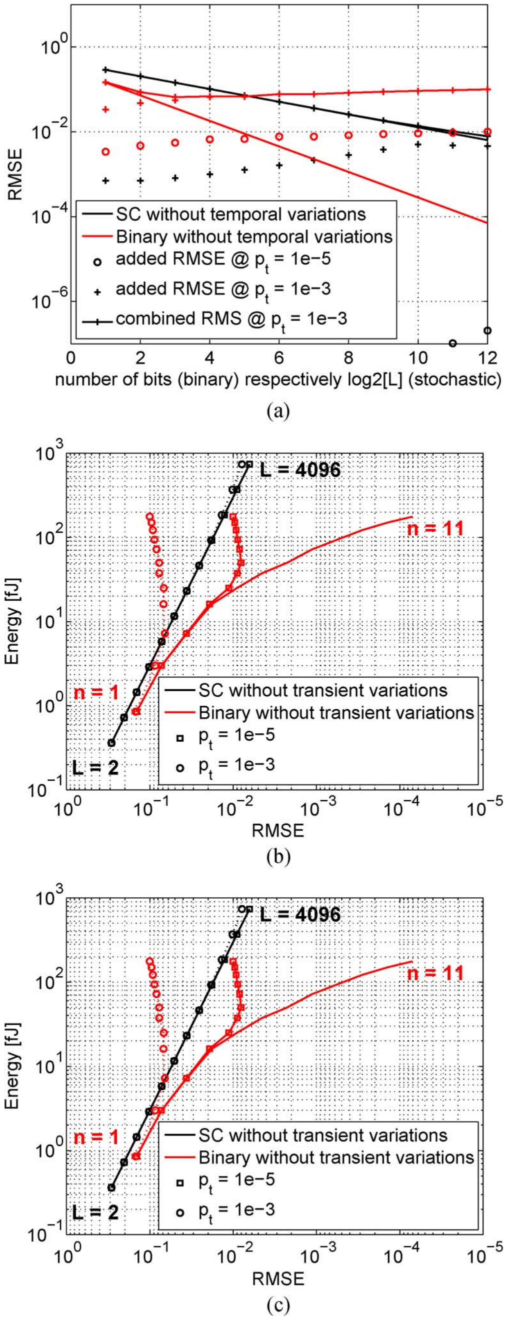
$$s_{\text{out}} = \text{xor}(s_a, s_t) = 0.998 \cdot s_a \tag{11}$$

which is an extra decrease in signal power (see Section IV), but in this case only a small distortion.

D. *Quantitative Comparison of Noise Levels in Single Stage SC and Binary Multipliers*

In order to quantitatively compare the noise in SC and binary digital electronics due to circuit variations, we first simulate a single stage SC multiplier as well as a standard array multiplier (without pipelining). Both systems are equally exposed to the previously mentioned sources of uncertainty. Type III (transient) errors are simulated on the system level. Type II (spatial) variations require transistor level Spice simulations.

1) Simulation Set-Up: Circuit simulations for SC are set-up as follows: two random bit-streams p_a and p_b are multiplied using a 40 nm AND-gate. At a given clock frequency, supply voltage is swept. For every voltage step the accuracy impact due to spatial variations is recorded. This dependency is only a function of the used circuit, spatial variations and frequency. The minimal supply voltage at which no type II errors occur is used to further assess the impact of type I and III errors. The binary multiplier works at a low frequency f_{bin} of 31 MHz $(\text{period} = 32 \text{ ns})$ to operate near the minimum energy point. The SC-multiplier hence requires a much higher f_{SC} of 496 MHz $(\text{period} = 2 \text{ ns})$ at a parallelization degree of $P = 16$. All SC and binary circuit simulations are done using Spice with 15 Monte-Carlo runs, which offer sufficient resolution for the targeted first-order analysis. To mimic more advanced technologies with more uncertainty, extra Vt- and β-mismatch is added using a verilog-A behavioral model. Transient circuit variations such as bit-flips or cycle-to-cycle variations are modelled with an XOR-gate on every logical output node. Input stream p_a will be inverted at a distortion rate p_t.



(a)

(b)

(c)

Fig. 7. Comparison of the effect of three different noise sources—inherent, spatial, and transient—on the output RMSE of a binary and stochastic single stage multiplier. (a) Transient variations + inherent RMSE. (b) Transient variations + inherent RMSE. (c) Spatial variations + inherent RMSE, A_{vt} and A_β are pelgrom's constants.

2) Simulation Results: Fig. 7 shows the results of our simulations. Fig. 7(a) shows the inherent RMSE and the influence of transient circuit variations, both for the binary and the SC implementation. The full lines plot inherent RMSE versus the bitwidth n for binary systems as a function of the stream length L for SC systems. The markers show the *added* RMSE due to transient circuit variations at different rates p_t as a function of n or L. The marked-lines show the total combined RMSE. By introducing transient errors, the achieved RMSE will be higher for the same n or L. This figure clearly shows that SC multipliers outperform binary as they can reach much lower RMSE under the same circumstances, by using longer bit-streams. This insensitivity was already illustrated visually in Fig. 1.

Fig. 7(b) further illustrates this feature by plotting the energy consumption as a function of RMSE for binary and stochastic implementations. Even at the relatively low transient error rate of $1e-5$, it is impossible to achieve an RMSE lower than $3e-8$ using a binary system. This corresponds to a binary accuracy of five bits. At a p_t of $1e-3$, only two bit binary accuracy can be reached. The accuracy degrades further when using larger bit-widths. This degradation is due to two reasons. First, the number of logical/flippable nodes increases quadratically in a binary carry-save multiplier. Second, MSB-nodes flip at the same rate as LSB-nodes, but contribute much more to the global RMSE. SC clearly has an advantage over binary computing in the case of transient circuit variations. The contribution of type III variations to the global RMSE at a flip rate of $1e-5$ is negligible. SC's reduced hardware complexity leads to less logical nodes. Furthermore, flipped bits always lead to an LSB error. Low mean errors can be achieved by using longer bit-streams, or equivalently, by investing more energy (see Section II-B and Section V-C).

Fig. 7(c) plots the energy of a multiplication operation using both the SC and the binary logic type, as a function of achieved RMSE for different amounts of type II variations. These simulation results also take type I variations into account. From (3) and (5) the quadratic relationship between energy dissipation and noise power $\mathrm{RMSE}^2 = \sigma^2$ could be predicted

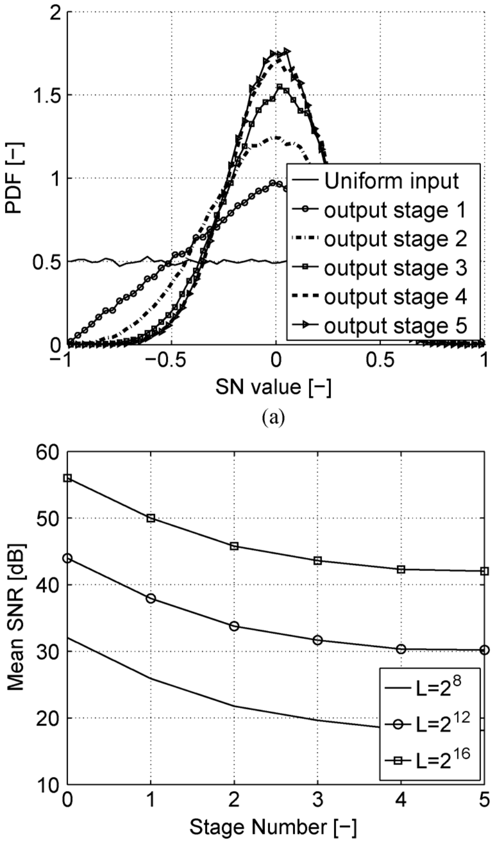
$$E_{\mathrm{SC}} = \frac{k}{6 \cdot \mathrm{RMSE}^2}. \tag{12}$$

For SC in the 40 nm case, k equals 0.13 fJ/bit-operation. In the case with $A_{vt} = 5.0e09\ \mathrm{Vm}$ and $A_{\beta} = 5.0e-9\ \mathrm{m}$, k equals 0.24 fJ/bit-operation. For $n \geq 2$, a best fit for the 40 nm binary case can be given by

$$E_{\mathrm{bin}} = c \cdot (2^n)^{\frac{1}{2}} \tag{13}$$

$$E_{\mathrm{bin}} = c \cdot \left(\frac{1}{\sqrt{12}\mathrm{RMSE}}\right)^{\frac{1}{2}} \tag{14}$$

where c equals 3.3. In the case with highest A_{vt} and A_{β}, c equals 15.5. SC has no advantage over binary for any RMSE in the 40 nm case for low to moderate spatial variations. It is clear that SC only outperforms binary multiplication in terms of energy usage when very high RMSE are tolerated and high spatial variations are present. High RMSE can be allowed in some image processing applications, such as edge detection [7]. At high RMSE, the energy usage is generally similar between the two systems, but it rises quicker in the binary multiplier than in SC with increasing spatial variations. However, due to the quadratic dependence of RMSE in SC, binary logic still performs better. Furthermore, the energy usage in binary can be reduced by pipelining the multiplier; this effectively reduces the impact of type II variations on delay and energy. Pipelining is not possible in the SC multiplier, since its arithmetic blocks only have a single stage and are inherently sequential.



(a)

(b)

Fig. 8. SNR degradation in multi-stage SC systems. (a) PDF of stochastic numbers after several stages. (b) SNR power ratio after several stages.

The usage of this proposed framework allows a quick evaluation of the single-stage energy-efficiency of SC in a new technology if parameters k, c and p_t are known. In technologies with sufficiently low k and high p_t, SC will be preferable to binary computation. We will quantify this by comparing an actual implementation of a binary and a SC DCT block in 40 nm CMOS and TFET in Section V.

However, to do this the previous analysis is not sufficient for multi-stage systems, since it does not incorporate the decrease of signal power after several gates. The following sections elaborate on this effect.

IV. Decreasing Signal Power in Multi-Stage Stochastic Computing

A second effect concerning accuracy in SC, is the decrease of mean signal power after several stages (Fig. 8). This is evident, since stochastic numbers always have an amplitude smaller than 1. Scaled addition $(s_c = (s_a + s_b)/2)$ for example will have $s_c < max(s_a, s_b)$, can only make numbers smaller in amplitude and therefore decreases mean signal power $P_{\mathrm{sig}} = s^2$. Due to this effect, the previous assumption of uniformly distributed output values cannot be made. Fig. 8(a) shows how the probability density function (PDF) of the output

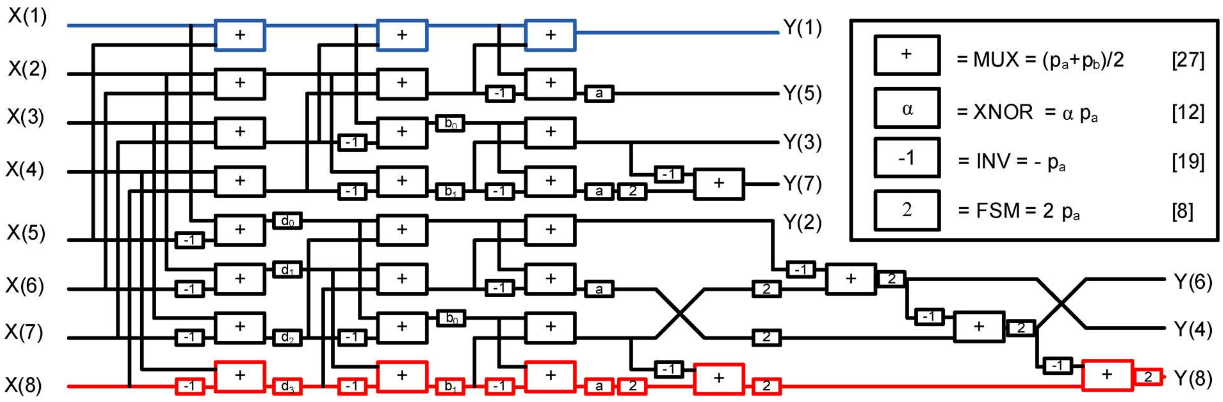


Fig. 9. DCT architecture based on Hou [33]. α are constants with $|\alpha| < 1$. Number of each gate type is also indicated. Path I is blue, path II is red.

values changes in the multi-stage SC circuit of Fig. 5(a). At the first stage, all gates receive uniformly distributed input values on the $[-1,1]$ interval. As these signals pass more stages, their PDF becomes narrower around $s = 0$. Numbers with larger amplitudes cease to appear and the mean signal power drops. If multipliers are used, the signal power will drop even faster. This analysis is in contrast to binary systems, where there is no reduction in signal power after multiple stages.

The combination of effects Section III (high inherent noise at low amplitudes) and Section IV (decreasing signal power after multiple stages) will lead to low signal-to-noise (SNR) power ratios in multi-stage systems. Hereby the noise power P_{noise} is defined by the deviation σ^2_{sig} due to the different noise sources of Section III. This SNR decrease is illustrated in Fig. 8(b), where the SNR in the multi-stage adder system of Fig. 5(a) is plotted at the output of every stage. Note that the mean SNR clearly scales with L and drops after several stages S.

V. Accuracy Evaluation of Multi-Stage SC Circuits

Using the results of the previous sections, a general method to evaluate SC's accuracy in circuits suffering from type I and type III variations can be summarized in a methodological design strategy.

A. Methodological Design Strategy

To validate the accuracy of the SC system, the following four-step methodology is proposed.

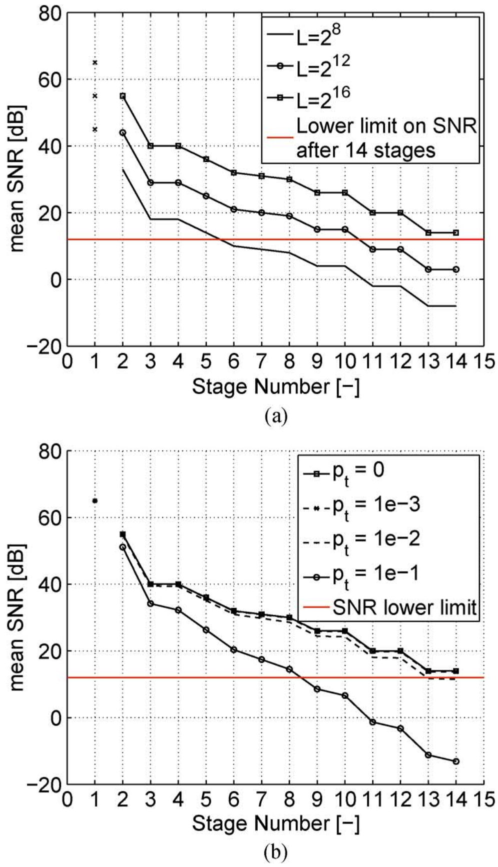
1) **Evaluate the probability density function of the output signal** starting from a uniform input distribution. This can be done numerically.
2) **Compute output noise power** by modelling inherent noise in SC as a binomial process, and simulating transient errors with an expected flip rate p_t. When more complex blocks, such as $\times c$ (with $c > 1$) are used, both the signal and the standard deviation of the noise at this stage are multiplied by c.
3) **Calculate the mean output SNR** from the known output and noise distributions.
4) **Compare the achieved, with the required SNR/precision and choose SC length** L. This precision will be application dependent.

B. Accuracy Evaluation of 1D-DCT Stochastic Block

As a practical example we perform the proposed accuracy analysis on the complex DCT of Fig. 9. This block is a classical DCT implementation based on the work of Hou [33]. This DCT block contains several paths with different numbers of stages and is a part of a JPEG encoder. In the algorithm, quantization and inverse decoding are performed in an ideal way. We will discuss two data-paths in the DCT block. First, the shortest path $X(1)$ *to* $Y(1)$ (path I), indicated on Fig. 9 as a blue line, consisting out of three stages ($S = 3$). Second, the longest path $X(8)$ *to* $Y(8)$ (path II), indicated as a red line on the figure, consisting out of 14 stages ($S = 14$). The required output precision depends on the implemented algorithm. If a full precision DCT-block is wanted, the accuracy requirements will be high. However, in the JPEG compression algorithm, the outputs of the 2D-DCT blocks are quantized. Due to this quantization, the required output precision of path I and II is respectively 4 bit ($6.02 \cdot n \approx 24$ dB SNR) and 2 bit binary precision (12 dB SNR).

Path I exists out of three stages of scaled stochastic adders. This is the same circuit as the one from Section III and Fig. 5(a), so the results of this analysis can be used directly. It is clear from Fig. 8(b), that both the $L = 2^{12}$ and the $L = 2^{16}$ implementations achieve more than 24 dB SNR at $S = 3$. An $L = 2^{12}$ implementation thus suffices for path I. Path II is more complex and contains 14 stages of different SC circuitry, including $\times 2$ blocks. Its precision requirements, however, are somewhat weakened (12 dB). Fig. 10(a) shows the results of our accuracy assessment design methodology on this path when no transient circuit variations are present. The mean SNR is computed after every stage for $L = 2^8$, $L = 2^{12}$, and $L = 2^{16}$ implementations.

The figure shows, that only the $L = 2^{16}$ implementation achieves better than 12 dB mean SNR at the last stage. The $L = 2^8$ and $L = 2^{12}$ implementations do not suffice. Observe that the mean SNR is still relatively high when only a few stages are used. The fact that only the $L = 2^{16}$ implementation is sufficiently accurate is further illustrated in Fig. 11(a)–(d). Where the full results of the JPEG compression with stochastic DCT-blocks are shown and compared. The accuracy of the implementations can be verified visually or more formally by comparing the achieved compression ratios (CR) and RMSE de-



(a)

(b)

Fig. 10. SNR after every stage in path II of Fig. 9. (a) SNR for different L, no transient variations. (b) SNR for $L = 2^{16}$ for different p_t.

TABLE I
RMSE IN JPEG FOR DIFFERENT ERROR RATES AND IMPLEMENTATIONS

Implementation	$L = 2^8$	$L = 2^{12}$	$L = 2^{16}$	Binary
pt =1e-1	47.9%	47.1%	14.5%	41.7%
pt =1e-2	38.6%	12.9%	3.4%	31.1%
pt =1e-3	37.9%	12.8%	2.3%	12.1%
pt =1e-5	37.8%	12.7%	2.3%	2.5%
pt =0	37.7%	12.7%	2.3%	2%

TABLE II
ENERGY DISSIPATION IN DIFFERENT 40 nm DCT IMPLEMENTATIONS

Implementation	$L = 2^8$	$L = 2^{12}$	$L = 2^{16}$	Binary
Parallellism	$P = 2$	$P = 32$	$P = 512$	-
$area$ [-]	0.95e3	15.3e3	244e3	15e3
f [MHz]	150	150	150	17
E_{total} [fJ]	4.5e3	72.3e3	1156.7e3	2.7e3
$E_{relative}$ [-]	1.67	26.72	427.52	1
RMSE @ pt <1e-6	37.7%	12.7%	2.3%	2%

TABLE III
ENERGY DISSIPATION IN DIFFERENT TFET [30] DCT IMPLEMENTATIONS

Implementation	$L = 2^8$	$L = 2^{12}$	$L = 2^{16}$	Binary
Parallellism	$P = 2$	$P = 32$	$P = 512$	-
$area$ [-]	0.95e3	15.3e3	244e3	15e3
$f_{relative}$ []	150	150	150	17
$E_{relative}$ [-]	1.01	16.16	258.6	1
RMSE @ pt =1e-2	38.6%	12.9%	3.4%	31.1%
RMSE @ pt =1e-3	37.9%	12.8%	2.3%	12.1%
RMSE @ pt =1e-5	37.8%	12.7%	2.3%	2.5%

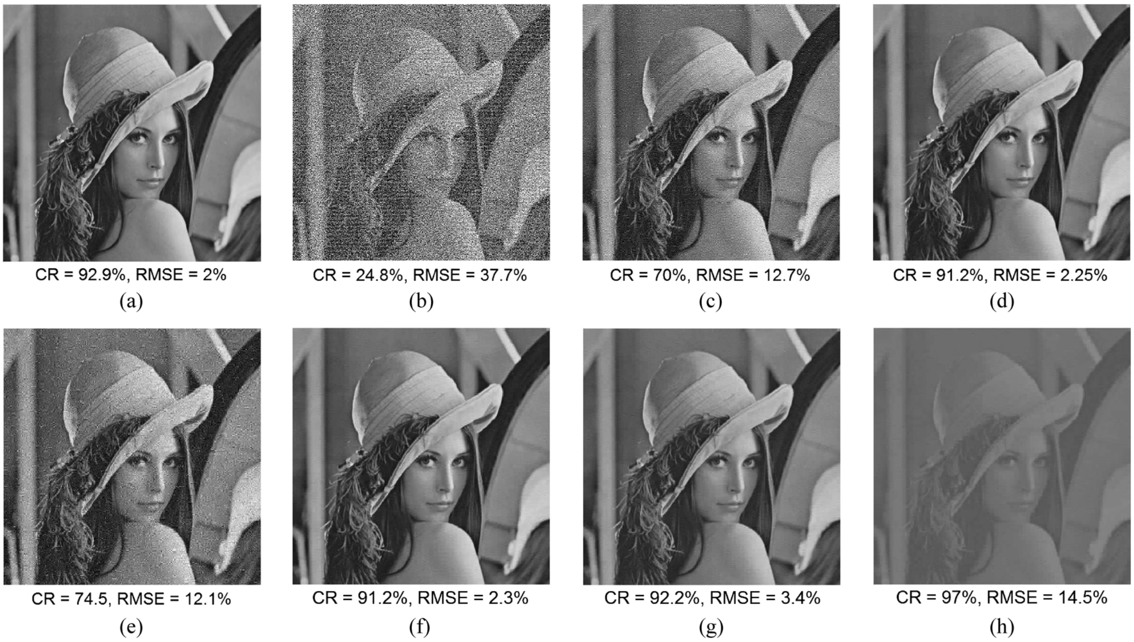
viations of the uncompressed picture. Since the channel $Y(8)$ represents high spatial frequencies, high noise levels on this channel will introduce nonexistent high-frequency terms that cannot be compensated for in the JPEG quantization step. This explains the visually noisy images. Only the accuracy of the $L = 2^{16}$ implementation is reasonable and well in range of the ideal version, as was predicted by our accuracy evaluation method.

We can repeat the same analysis for a system suffering from severe transient circuit variations. The deviation due to transient errors leads to an extra decrease in signal power and associated SNR. As long as this added effect is small enough, the stochastic implementation will suffice. This is again illustrated for path II in Fig. 10(b), where SNR is plotted as a function of the stage S and the transient bit-flip rate p_t. Even at the very high $p_t = 1e-2$, the SC implementation achieves high accuracy if $L = 2^{16}$. SNR only drops slightly faster than in the case without transient variations. At $p_t = 1e - 3$ the difference is only 0.24 dB after 14 stages.

The effects of these transient variations are further illustrated in Fig. 11(e)–(h), where the output accuracy of a binary and a SC-implementation is compared. For example, in Fig. 11(e) and (f), all logical nodes are flipped at a rate of $p_t = 1e - 3$. In the binary implementation, this leads to severe accuracy degradation, both visually and formally in terms of CR and RMSE. No significant accuracy degradation is seen in the SC implementation. An $L = 2^{16}$ is still needed to cope with SC's inherent inaccuracy, but the added noise due to the transient variations is negligible and not visible. An SC of length $L = 2^{16}$ does not suffice any more when a $p_t = 1e - 1$ is applied [Fig. 11(h)], as was predicted by our method in Fig. 10(b). Table I gives an overview of the achieved RMSE for every JPEG implementation. Observe that the accuracy of SC systems is ultimately limited by their number length L. In the case of a transient error rate $p_t = 1e - 3$, SC is the better choice in terms of accuracy. The binary implementation has 12.1% RMSE, while SC can still achieve high accuracy (2.3% RMSE) at this error rate.

C. Energy Evaluation of 1D-DCT Stochastic Block

Previous system level evaluation shows that a minimal $L = 2^{16}$ is needed for JPEG compression. This high L will lead to a high energy dissipation in the JPEG implementation since E scales linearly with L (3). To illustrate this, we choose an example implementation operating near the minimum energy point of the SC circuitry. If a 1D-DCT delay of 1000 ns is needed, the SC system should operate at 150 MHz at a $P = 512$ (2). Tables II and III show the estimated energy dissipation and circuit area for different implementations of the DCT block, in 40 nm CMOS and 26 nm TFET [30]. These estimations



CR = 92.9%, RMSE = 2%
(a)

CR = 24.8%, RMSE = 37.7%
(b)

CR = 70%, RMSE = 12.7%
(c)

CR = 91.2%, RMSE = 2.25%
(d)

CR = 74.5, RMSE = 12.1%
(e)

CR = 91.2%, RMSE = 2.3%
(f)

CR = 92.2%, RMSE = 3.4%
(g)

CR = 97%, RMSE = 14.5%
(h)

Fig. 11. JPEG compression results. Only the $L = 2^{16}$ circuit achieves high accuracy. Both visually and in terms of compression ratio (CR) and RMSE. Figures a–d show the results without transient variations. (a) Ideal binary JPEG compression. (b) SC $L = 2^8$. (c) SC $L = 2^{12}$. (d) SC $L = 2^{16}$. Images (e)–(h) show the results for different flip rates p_t. (e) Binary compression at $p_t = 1e - 3$. (f) SC $L = 2^{16}$ at $p_t = 1e - 3$. (g) SC $L = 2^{16}$ at $p_t = 1e - 2$. (h) SC $L = 2^{16}$ at $p_t = 1e - 1$. At $p_t = 1e - 3$, the binary accuracy is already unacceptable, while the SC implementation can withstand a $p_t = 1e - 2$. This was predicted by our method.

only include the energy usage of the combinational arithmetic, not of any flip-flops that are needed for data-path synchronization. Added flip-flops will come with a larger increase in energy usage in SC due to the high number of needed switches.

We discuss energy consumption both in the simulated 40 nm technology and in emerging technologies.

1) 40 nm Technology: Using Spice, we can simulate the energy per bit-operation for every SC arithmetic block. Operating near the minimum energy point in a 40 nm technology they consume $k_{\text{MUX}} = 0.18$, $k_{\text{XNOR}} = 0.13$, $k_{\times 2} = 1.41$, and $k_{\text{INV}} = 0.0625$ fJ/bit-*operation*. Observe the high energy cost of the $\times 2$ block [19]. This is in contrast to a binary implementation, where this block is essentially free. Table II shows the derived energy dissipation for the full DCT-circuit. It is clear that even for the $L = 2^8$ version, the energy-efficiency of SC is much lower than the binary DCT. An implementation using words of $L = 2^{12}$ (at $P = 32$) consumes the same area as the conventional implementation. Note that the $L = 2^{16}$ energy consumption is worst-case. Several paths in the DCT can be implemented using $L = 2^{12}$ or $L = 2^{10}$, making the energy gap smaller in a real implementation. For the realistic $p_t \simeq 0$ in 40 nm CMOS, the achieved RMSE and energy dissipation is however lowest in the binary implementation. There is no incentive to opt for a stochastic system in this case.

2) Emerging Technologies: Relying on TFET models found in literature [28], [30], we can estimate the corresponding relative energy dissipation in a TFET technology (Table III). In TFET, a realistic p_t lies between $1e - 2$ and $1e - 5$. This estimation is based on the work of [34], which compares delay variations under influence of RTN in advanced CMOS technologies and on [27] which gives numbers for TFET. According to [34], 45 nm CMOS technology has less than 5% delay variation due to RTN in large data paths. [27] shows TFET transistors can have up to 300% $\Delta I_D / I_D$ due to RTN.

For error rates $p_t = 1e - 3$ and higher the binary implementation is no longer a solution, since it cannot achieve sufficient accuracy. For these error rates only SC is accurate, albeit at a high energy cost.

Note that the energy gap between SC and binary slightly decreases in TFET compared to CMOS. In emerging technologies, the energy coefficients k and c are predicted to be much lower than in current CMOS (see Section II-C). The binary minimum energy coefficient c (Section III-D) will however increase relatively compared to k. This relative increase of c is due to two reasons. First, c increases relatively to k due to the difference in slope of the energy-delay curves between CMOS and TFET technologies [30]. Second, there is a relative increase of c due to the higher spatial variations in TFET than in CMOS technology [28]. More static variations lead to a higher c, as we prove in our simulations of Fig. 7(c). The energy consumption of the binary multiplier increases drastically with increasing spatial variations (c increases), while the energy consumption of the SC implementation stays close to the nominal value (k remains similar). Both the absolute energy penalty and the relative energy penalty compared to binary of SC systems will thus decrease in emerging technologies.

It is hence clear that only applications with a limited number of circuit stages (low L) can benefit from SC technology at high energy efficiency. SC's main advantage in emerging technologies therefore remains its inherent robustness to transient errors.

VI. Conclusion

Stochastic computing is a promising circuit technology, as it is robust against soft errors. But, it should not be considered a low energy alternative for binary arithmetic. This is due to SC's inherent accuracy loss. To analyze this accuracy, two effects should be considered: the occurrence of noise and the decrease of signal power in multi-stage SC systems. This paper carefully analyzed and formalized these effects, resulting in a methodological design flow and energy-efficiency estimation methodology for multi-staged stochastic circuits.

This paper categorizes and discusses three types of noise. First, SC is inherently inaccurate due to randomization effects, even if constant, near-exact number generators are used. We demonstrate that after several stages, the inherent noise in SC can be modelled as a binomial process, which has noise levels that are much higher than the inherent quantization noise in binary systems. Second, spatial circuit variations can lead to errors. They can be tuned out by carefully balancing the systems supply voltage. Third, there are transient circuit variations. This type of noise only leads to very limited distortion in SC, while it strongly affects traditional binary computation. When transient circuit variations are present, SC will greatly outperform binary implementations. The paper further explained how multi-stage SC circuits decrease mean signal power and that variance is highest at low bipolar amplitudes. This combination leads to low SNR at SC outputs. This can be compensated by using longer bit-streams, leading to higher energy dissipation.

To correctly assess these combined effects, we formalized this noise and signal assessment towards a multi-stage SC design methodology. The methodology has been validated and tested on a multi-stage DCT block as part of a JPEG encoder.

This analysis shows stochastic computing can be an alternative to binary in emerging technologies suffering from severe transient circuit variations. However only in applications with a limited number of stages or low RMSE requirements this can be achieved at limited energy penalty.

References

[1] N. Shanbhag *et al.*, "Stochastic computation," in *Design Automat. Conf.*, Jun. 2010, pp. 859–864.

[2] B. R. Gaines, "Stochastic computing systems," in *Advances in Information Systems Science*, J. Tou, Ed. New York: Springer, 1969, pp. 37–172.

[3] R. Gaines, "Stochastic computing," in *Proc. AFIPS Spring Joint Computer Conf.*, Atlantic City, NJ, 1967, pp. 149–156.

[4] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies*, vol. 34, pp. 43–98, 1956.

[5] A. Alaghi and J. Hayes, "Fast and accurate computation using stochastic circuits," *Design, Automat. Test Eur.*, 2014.

[6] A. Naderi, S. Mannor, M. Sawan, and W. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5617–5626, Nov. 2011.

[7] A. Alaghi, C. Li, and J. Hayes, "Stochastic circuits for real-time image processing applications," in *Proc. 50th Annu. Design Automat. Conf.*, 2013, p. 136.

[8] P. Li and D. Lilja, "Using stochastic computing to implement digital image processing algorithms," in *Proc. Int. Conf. Comput. Design*, 2011, pp. 154–161.

[9] H. Aliee and H. Zarandi, "Fault tree analysis using stochastic logic: A reliable and high speed computing," in *Proc. Annu. Reliabil. Maintainabil. Symp.*, 2011, pp. 1–6.

[10] H. Aliee and H. Zarandi, "A fast and accurate fault tree analysis based on stochastic logic implemented on field-programmable gate array," *IEEE Trans. Reliability*, vol. 62, no. 1, pp. 13–22, Mar. 2013.

[11] Y.-N. Chang, "Architectures for digital filters using stochastic computing," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 2697–2701.

[12] N. Saraf, K. Bazargan, D. J. Lilja, and M. D. Riedel, "IIR filters using stochastic arithmetic," *Design, Automat. Test Eur.*, pp. 1–6, 2014.

[13] A. Alaghi and J. Hayes, "A spectral transform approach to stochastic circuits," in *Proc. 30th Int. Conf. Comput. Design*, 2012, pp. 315–321.

[14] W. Qian and M. Riedel, "The synthesis of robust polynomial arithmetic with stochastic logic," in *Proc. 45th Design Automat. Conf.*, 2008, pp. 648–653.

[15] W. Qian and M. Riedel, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Computers*, vol. 60, no. 11, pp. 93–105, Jan. 2011.

[16] A. Alaghi, "Survey of stochastic computing," *ACM Trans. Embed. Comput. Syst.*, vol. 12, 2013.

[17] P. Jeavons *et al.*, "Generating binary sequences for stochastic computing," *IEEE Trans. Inf. Theory*, vol. 40, no. 3, pp. 716–720, May 1994.

[18] B. Zelkin, "Arithmetic unit using stochastic data processing," Patent US 6 745 219.

[19] B. Brown and H. Card, "Stochastic neural computation I: Computational elements," *IEEE Trans. Comput.*, vol. 50, no. 9, pp. 891–905, Sep. 2001.

[20] S. L. Toral *et al.*, "Stochastic pulse coded arithmetic," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2000, vol. 1, pp. 599–602.

[21] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuit and microarchitecture," in *Proc. 40th Design Automat. Conf.*, 2003, pp. 338–342.

[22] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *Proc. IEEE Int. Reliabil. Phys. Symp.*, 2011.

[23] A. Fantini, L. Goux, R. Degraeve, D. J. Wouter, N. Raghavan, G. Kar, A. Belmonte, Y.-Y. Chen, B. Govoreanu, and M. Jurczak, "Intrinsic switching variability in HfO2 RRAM," in *IEEE Int. Memory Workshop*, 2013, pp. 30–33.

[24] I. Kazi, P. Meinerzhagen, P.-E. Gaillardon, D. Sacchetto, A. Burg, and G. De Micheli, "A ReRAM-based non-volatile flip-flop with sub-VT read and CMOS voltage-compatible write," in *Proc. IEEE 11th New Circuits Syst. Conf.*, 2013, pp. 1–4.

[25] M. R. Choudhury, Y. Yoon, J. Guo, and K. Mohanram, "Graphene nanoribbon FETs: Technology exploration for performance and reliability," *IEEE Trans. Nanotechnol.*, vol. 10, no. 4, pp. 727–736, Jul. 2011.

[26] S. Datta, H. Liu, and V. Narayanan, "Tunnel FET technology: A reliability perspective," *Microelectron. Reliabil.*, vol. 54, no. 5, pp. 861–874, 2014.

[27] M.-L. Fan, S.-Y. Yang, V. Pi-Ho, Y.-N. Chen, P. Su, and C.-T. Chuang, "Single-trap-induced random telegraph noise for FinFET, Si/Ge nanowire FET, tunnel FET, SRAM and logic circuits," *Microelectron. Reliabil.*, vol. 54, no. 4, pp. 698–711, 2014.

[28] U. Avci, D. H. Morris, S. Hasan, and R. Kotlyar, "Energy efficiency comparison of nanowire heterojunction TFET and SI MOSFER at $\lg = 13\ \mathrm{nm}$, including p-TFET and variation considerations," in *Proc. IEEE Int. Electron Devices Meet.*, 2013, pp. 33.4.1–33.4.4.

[29] V. Saripalli, K. A. Mishra, S. Datta, and V. Narayanan, "An energy-efficient heterogenous CMP based on hybrid TFET CMOS-cores," in *Proc. 48th Design Automat. Conf.*, 2011, pp. 729–734.

[30] S. Datta, R. Bijesh, H. Liu, D. Mohata, and V. Narayanan, "Tunnel transistors for energy efficient computing," in *Proc. IEEE Int. Reliabil. Phys. Symp.*, 2013.

[31] A. Alaghi and J. Hayes, "Exploiting correlation in stochastic circuit design," in *Proc. IEEE 31st Int. Conf. Comput. Design*, 2013, pp. 39–46.

[32] C. Ma *et al.*, "Understanding variance propagation in stochastic computing systems," in *Proc. IEEE 30th Int. Conf. Comput. Design*, 2012, pp. 213–218.

[33] H. Hou, "A fast recursive algorithm for computing the discrete cosine transform," in *Appl. Digital Image Process. IX*, 1986, vol. 35, no. 10, pp. 14–25.

[34] H. Luo *et al.*, "Temporal performance degradation under RTN: Evaluation and mitigation for nanoscale circuits," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2012, pp. 183–188.

Bert Moons (S'13) was born in Antwerp, Belgium, in 1991. He received the B.S. degree in electrical engineering, in 2011, and the M.S. degree, in 2013, both from the KU Leuven, Leuven, Belgium, where he is currently working toward the Ph.D. degree on context-aware and run-time adaptable digital circuits for error-tolerant processing in low power applications.

He joined the ESAT-MICAS laboratories in 2013 as a Research Assistant after he received a grant from the Flemish agency for innovation by science and technology (IWT).

Mr. Moons received the Resmiq student paper award at the New Circuits and Systems conference (NEWCAS) in 2014.

Marian Verhelst (SM'13) received the Ph.D. degree in electrical engineering from the KU Leuven, Leuven, Belgium, in 2008.

In 2005, she resided for three months at the Berkeley Wireless Research Centre (BWRC), University of California, Berkeley, CA, USA. From 2008 to 2011, she was with Intel Labs, Portland, OR, USA. In the Wireless Communications Research Lab, she worked on digitally-enhanced analog and RF circuits for performance enhancement, self-test, and self-calibration. In 2012, she returned to Belgium and became a Professor at the ESAT-MICAS group of KU Leuven. Her research group focusses on smart, self-adaptive system architectures and circuits for ubiquitous sensing and computing.