IEEE Journal on Exploratory Solid-State Computational Devices and Circuits

Received 9 February 2019; revised 15 March 2019; accepted 1 April 2019. Date of publication 12 April 2019; date of current version 21 May 2019.

Digital Object Identifier 10.1109/JXCDC.2019.2909912

Error-Resilient Spintronics via the Shannon-Inspired Model of Computation

AMEYA D. PATIL[1] (Student Member, IEEE), SASIKANTH MANIPATRUNI[2], DMITRI E. NIKONOV[2] (Senior Member, IEEE), IAN A. YOUNG[2] (Life Fellow, IEEE), AND NARESH R. SHANBHAG[1] (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801 USA
[2]Components Research, Intel Corporation, Hillsboro, OR 97124 USA

CORRESPONDING AUTHOR: A. D. PATIL (adpatil2@illinois.edu)

This work was supported in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) Program sponsored by DARPA, and in part by SONIC, one of the six SRC STARnet Centers sponsored by MARCO and DARPA.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

ABSTRACT The energy and delay reductions from CMOS scaling have stagnated, motivating the search for a CMOS replacement. Spintronic devices are one of the promising beyond-CMOS alternatives. However, they exhibit high switching error rates of 1% or more when operated at energy and delay comparable to CMOS, rendering them incompatible with the deterministic nature of digital implementations. In this paper, we employ a Shannon-inspired model of computation to enhance the tolerance of all-spin logic (ASL)-based implementations to gate-level switching errors. We develop the logic-level path delay reallocation techniques to shape the output error statistics and propose a novel error compensation scheme to achieve 1000× higher tolerance to device-level switching errors while maintaining the classification accuracy of an ASL-based support vector machine (SVM) classifier.

INDEX TERMS All spin logic, beyond-CMOS, machine learning, spintronics, statistical computing.

I. INTRODUCTION

THE PAST few decades have seen tremendous improvement in computational efficiency, in part, due to relentless CMOS scaling to achieve the improved density of transistors while reducing their switching energy and delay and preserving nearly error-free switching behavior. However, as the channel lengths continue to reduce beyond a few tens of nanometers, the energy and delay reductions have stagnated. Hence, it is of great interest to explore new computational devices and new models of computation that leverage the unique properties of such devices to enable continued computational scaling.

In particular, spin-based computational devices built with nanomagnets and spin-polarized transport have emerged as a viable beyond-CMOS option, due to their following favorable attributes: 1) nonvolatility; 2) higher logical efficiency; and 3) high integration density and compatibility with the state-of-the-art back-end electronics manufacturing processes. These devices are the subsets of the beyond-CMOS devices that include devices based on electron spin [1], [2] and magnetoelectric [3], [4] phenomena.

However, spin-based devices are not competitive to CMOS [5], in terms of switching energy and delay, due to their high energy–delay requirements to achieve deterministic switching [6]–[9]. As the switching energy or delay is reduced, their switching error probability increases, rendering them incompatible with the required determinism of the digital logic. Hence, multiple research efforts are underway to improve the energy efficiency of the spin-based implementations.

Recent attempts at improving the energy efficiency of spin-based implementations particularly focus on exploiting unique attributes of spin-based devices to efficiently implement the machine learning algorithms. The examples include exploiting domain wall magnets for analog multiplication [10]–[12] using racetrack memory structures to achieve reconfigurable precision [13], efficient logic operations and data conversion [14], [15], and analog nature of spin currents for efficient dot-product computation [16]. Recently, researchers have also exploited the nanomagnet stochasticity for efficient probabilistic inference implementations. The examples include efficient realization of restricted Boltzmann machines [17], stochastic optimization schemes [18], probabilistic spiking neural networks [19], and stochastic bit-stream computing [20], [21].

In this paper, we explore how one can significantly increase the switching error probability of spin-based logic gates in digital implementations of machine learning classifiers while maintaining their inference accuracy. This problem is akin to the classical problem formulation of achieving reliable computation using unreliable components posed by von Neumann [22], where a reliable logic network was defined as the one whose output exhibits a probability of error $p_e < 0.5$ when designed using ϵ-noisy logic gates, i.e., gates whose outputs are in error with probability ϵ. It was further demonstrated that a reliable logic network can be designed for any logic function provided $\epsilon \leq 0.0073$ and that it is impossible to do so if $\epsilon > 1/6$. Later, tighter upper bounds on ϵ were obtained in a series of papers [23], [24], culminating with those of Evans and Schulman [25]. All these works do not consider the fundamental tradeoff between ϵ, energy, and delay and assume identical ϵ for all gates. Furthermore, they rely on gate-level replication to minimize the error probability of all intermediate binary signals in order to achieve a small p_e, leading to a prohibitive increase in the overhead.

In this paper, we employ the Shannon-inspired model of computation [26] to enhance the tolerance of all-spin logic (ASL)-based classifier implementations to gate-level switching errors while maintaining their inference accuracy. In the Shannon-inspired framework, hardware errors are engineered and then efficiently compensated via the introduction of tailored redundancy, in the spirit of Shannon's theory for communications [27]. The contributions of this paper are as follows.

1) We characterize the ϵ-energy–delay tradeoff for ASL gates to enable nonuniform ϵ assignments across logic gates.
2) We propose logic-level path delay reallocation techniques to assign appropriate error rates to individual gates, such that the resulting output error distributions are shaped to facilitate error compensation.
3) We propose a novel maximum likelihood (ML) error compensation scheme that exploits these shaped output error statistics to compensate for the errors efficiently.
4) We demonstrate a 1000× higher average error rate tolerance and a 3× lower energy-per-decision for an ASL-based digital support vector machine (SVM) implementation while maintaining its system-level classification accuracy.

The rest of this paper is organized as follows. Section II describes the relevant background, while Section III describes a modified ϵ-noisy model to capture the gate-level tradeoff between ϵ, energy, and delay. Section IV describes the proposed Shannon-inspired ASL-based SVM implementation. Section V presents the simulation results, while Section VI concludes this paper.

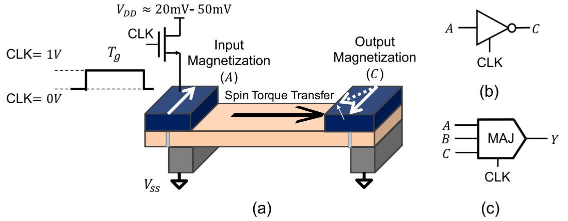


FIGURE 1. **ASL. (a) Diagram of clocked ASL inverter gate [28], [29]. (b) Clocked ASL inverter symbol. (c) Clocked ASL three-input majority gate symbol.**

II. BACKGROUND

A. ALL-SPIN LOGIC DEVICE

Fig. 1(a) shows a diagram of an ASL inverter. It consists of two nanomagnets separated by a conducting channel. The input nanomagnet polarizes the supply current passing through it. This creates a spin concentration gradient and propagates the spin current in the channel. This spin current, in turn, exerts a torque on the magnetization of the output nanomagnet, forcing it to switch.

Since the nanomagnets and the spin channel are metallic, the equivalent electrical resistance across the nanomagnet-channel stack is small (few Ωs), enabling these devices to operate at ultralow supply voltages. However, the electrical current through the input nanomagnet flows irrespective of the output activity, causing high static energy consumption. The nanomagnets, being nonvolatile, retain the magnetization vector state even when the supply current is switched OFF. Hence, [29] and [28] propose to clock these devices via a MOSFET, operating in the linear region, which acts as a switch, turning ON the ASL device only when it needs to compute, as shown in Fig. 1(a). The ON duration T_g of the clock can be externally controlled for each gate. Thus, the energy consumption of the clocked ASL gates is completely determined by T_g and the ON current of the gating MOSFET. Fig. 1(b) and (c) shows the logical symbols for the clocked ASL inverter and the three-input majority gate, respectively. Reference [28] proposed to share a single MOSFET across multiple nanomagnets by electrically stacking their supply terminals in series to significantly amortize the clock pulse generation and MOSFET switching overheads. In this paper, we assume such amortization described in [28] and focus on the impact of gate-level switching errors on the final output.

B. SUPPORT VECTOR MACHINE

Linear SVM [30] is a simple and popular machine learning algorithm for binary classification. The SVM learns a hyperplane to separate the training feature vectors into two regions, each corresponding to one class, as shown in the following:

$$\mathbf{w}^T\mathbf{x} + b \underset{\hat{z}=-1}{\overset{\hat{z}=1}{\gtrless}} 0$$

where $\mathbf{w}$ and b denote the trained weight vector and bias representing the separating hyperplane, respectively, $\mathbf{x}$ denotes the N-dimensional input feature vector, and $\hat{z}$ denotes the predicted label. If the true label is denoted by z, the accuracy of SVM is given by the probability of the classification error $p_e = \Pr\{\hat{z} \neq z\}$, which can be empirically estimated for a given data set.

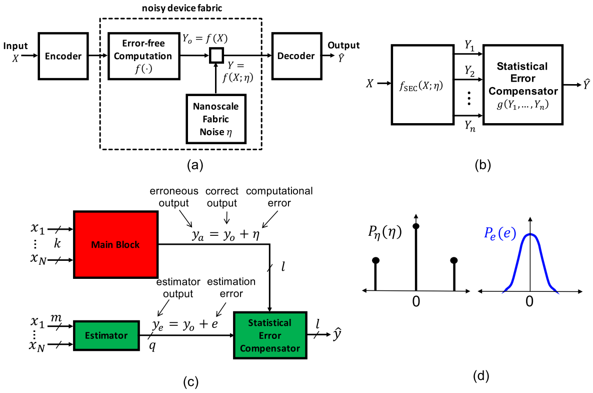


FIGURE 2. **Shannon-inspired model of computation. (a) Model. (b) SEC. (c) ANT, a special case of SEC, where the error compensator combines two unreliable outputs y_a and y_e. (d) Illustrative distributions of computational error η, estimation error e that lead to a low-complexity and accurate error compensator.**

C. SHANNON-INSPIRED MODEL OF COMPUTATION

The Shannon-inspired model of computation [26] [see Fig. 2(a)] comprises an encoder, a noise-free computation of the desired correct output $Y_o = f(X)$ being corrupted by the noise in nanoscale fabrics parameterized by variable η to generate the observed output $Y = f(X;\eta)$ of the error-prone device fabric (the channel), followed by the decoder that recovers the corrected output $\hat{Y}$. In Fig. 2(a), all variables $(X, Y_o, \eta, Y, \hat{Y})$ are random variables. In this paper, we use capital letters to denote random variables and small letters to denote their particular instance. For example, Y denotes a random variable, while y denotes a specific value of Y.

Statistical error compensation (SEC) [see Fig. 2(b)], one class of the design techniques within the Shannon-inspired framework [26], [31], introduces a statistical error compensator block as a decoder, which combines multiple unreliable outputs $Y_1, \dots, Y_n$ to compute the corrected output $\hat{Y}$. Algorithmic noise tolerance (ANT) [see Fig. 2(c)] is a special case of SEC, where the error compensator combines two unreliable outputs y_a and y_e. ANT consists of the main block (MB) designed using unreliable/noisy device fabric that accounts for 85%–90% of the total gate count complexity. It strives to compute correct output y_o but ends up computing y_a due to the unreliability of the underlying device fabric. ANT augments the MB with a low complexity estimator that computes an estimate y_e of the correct output y_o. Under the assumption of the additive noise model, the MB and estimator outputs are described as follows:

$$y_a = y_o + \eta \tag{1}$$

$$y_e = y_o + e \tag{2}$$

where η is a system-level hardware error observed at the MB output and e is the estimation error incurred due to inherent lower complexity of the estimator.

The estimator and the error compensator are designed using reliable, and hence energy-inefficient, circuits, constituting the error compensation overhead in ANT. Hence, their combined complexity (in terms of gate count) needs to be significantly ($\approx$5–10×) smaller than the MB. Previously, it has been shown that the complexity of the error compensator can be reduced by shaping the distributions of η and e, $P_\eta(\eta)$, and $P_e(e)$, respectively, to be disparate from each other, as shown in Fig. 2(b) and (c) [32]–[34]. In particular, a dense $P_e(e)$ is realized by introducing a reduced-precision estimator, while a sparse $P_\eta(\eta)$ is realized by permitting MSB errors in the LSB-first architectures [33]–[36]. Various design techniques to reduce the overhead of the estimator and the error compensator have been proposed [35]–[38].

D. MUTUAL INFORMATION

The mutual information (MI) $I(X;Y)$ between two random variables X and Y quantifies the amount of information conveyed about X by knowing the value of Y, and vice versa. The MI $I(X;Y)$ is defined as follows:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3}$$

where $H(X)$ and $H(X|Y)$ denote the entropy of X and conditional entropy of X, given Y, respectively. The entropy $H(X)$ of a random variable X quantifies the uncertainty about the value of X and is a function of its probability distribution. In this paper, we use MI metric to show that the Shannon-inspired model of computation (see Fig. 2) enhances the MI $I(Y_o; Y_a, Y_e)$, thereby enabling an accurate recovery of y_o from y_a and y_e.

III. MODELING STOCHASTICITY OF ASL DEVICES

In this section, we develop a gate-level model to capture the inherent device-level stochasticity of ASL at the circuit and architecture levels. Even after receiving the supply current I_{ON} ($> I_{\text{crit}}$) at the input nanomagnet, the output nanomagnet of the ASL gate may not switch due to the presence of the Langevin thermal noise [6]–[9], where I_{crit} denotes the minimum current required for nanomagnetic switching. In this paper, we refer to this probabilistic event as the *switching error* and its probability ϵ as the *switching error rate*. In [6], an analytical expression for ϵ was derived by employing the Fokker–Planck equation for magnetization vector switching dynamics governed by the fundamental LLG equation and was validated against the Landau–Lifshitz simulations of a macrospin including appropriate thermal field. This analysis indicates a gate-level tradeoff between switching error rate ϵ, the switching energy E_g, and the switching delay T_g of the ASL gates.

Fig. 3 shows the isoerror rate delay versus energy contours of an ASL inverter at various error rates. As expected, the error rate decreases with increasing energy or delay. In fact, when $I_{\text{ON}} \gg I_{\text{crit}}$, the expression for ϵ [6] can be

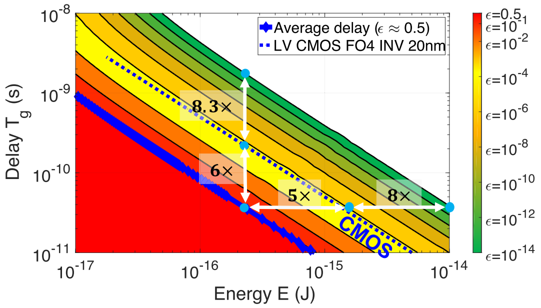


FIGURE 3. Tradeoff between switching error rate ϵ, switching energy E_g, and switching delay T_g for a clocked ASL inverter gate.

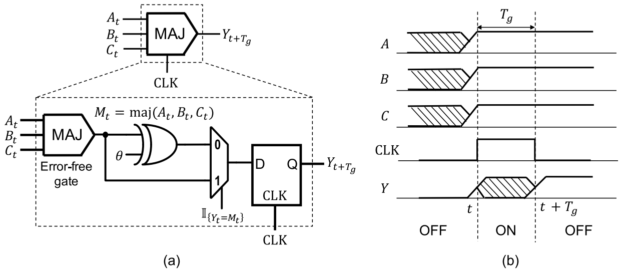


FIGURE 4. Modified ϵ-noisy gate model for clocked ASL. (a) Gate-level schematic emulating the stochastic behavior of a nonvolatile, clocked ASL three-input majority gate. (b) Timing diagram illustrating the phase, where the gate is ON and OFF.

simplified via the Taylor series approximation (as shown in Section I in the Supplementary Material) to

$$\epsilon(E_g, T_g) = \beta \exp\left(-\zeta\sqrt{E_g T_g}\right) \tag{4}$$

where β and ζ are the device-dependent constants described in Section I in the Supplementary Material. A three-input majority ASL gate operates with an error rate of $\epsilon(E_g, T_g)$, if all its inputs are equal, and with higher error rate of $\epsilon(E_g/3, T_g)$, otherwise. In this paper, we conservatively upper-bound the error rate of three-input majority gate to $\epsilon(E_g/3, T_g)$. Equation (4) explains the observed linearity of the contours at higher values of E_g or T_g in Fig. 3. We further note that ASL inverter consumes 8× more energy compared to the 20-nm CMOS FO4 inverter [2] at $\epsilon = 10^{-14}$ and at identical switching delays. Hence, ASL-based conventional digital architectures remain noncompetitive with respect to the present day CMOS. As ϵ is increased beyond 1%, the ASL inverter becomes more energy efficient than CMOS, demonstrating the potential for achieving energy efficiency, if one can tolerate such high gate-level error rates while maintaining the final system-level accuracy.

We develop a modified ϵ-noisy gate model [see Fig. 4(a)] to describe a clocked ASL gate, which comprehends its underlying stochastic behavior while being sufficiently abstract to permit the design and the analysis of complex ASL networks. The modified ϵ-noisy gate model captures: 1) the logic-level manifestation of device-level stochasticity; 2) the input dependence of ASL errors due to the nonvolatility of the nanomagnets, i.e., the ASL gate makes an error only when the output nanomagnet fails to switch when it should, implying a dependence of the error event on the input data; and 3) the role of the CLK terminal in the gate operation.

Fig. 4(b) shows the timing diagram for the modified ϵ-noisy model. The Boolean inputs A, B, and C are applied at time t. The ASL gate generates its output Y at time $t + T_g$, where T_g is the switching delay assigned to the ASL gate. The model comprises of an ideal noise-free Boolean gate whose output $M_t = \mathrm{maj}\{A_t, B_t, C_t\}$ is EXORed with a Bernoulli random variable θ with parameter ϵ, i.e., $\Pr\{\theta = 1\} = \epsilon$. The output selector [implemented using a multiplexer in Fig. 4(b)] computes the final output Y_{t+T_g} by choosing either the output of the EXOR gate $M_t \oplus \theta$ or the error-free output M_t. The D flip-flop models the nonvolatility, i.e., the ability to retain the output when CLK $= 0$. The EXOR gate output is chosen only if $Y_t \neq M_t$, capturing the fact that the switching error can occur only if the output nanomagnet is required to switch.

IV. SHANNON-INSPIRED ASL ARCHITECTURE

In this section, we describe how the Shannon-inspired approach can be applied to clocked ASL networks to increase their tolerance for switching errors. In Section IV-A, we propose the path delay reallocation techniques that exploit the gate-level tradeoff between ϵ, E_g, and T_g to shape the output error statistics and, thereby, ease error recovery. In Section IV-B, we propose a novel fusion block architecture to compensate for the switching errors.

A. SHAPING ERROR STATISTICS

In clocked digital ASL networks, the random switching errors occur at the output of every logic gate, as modeled in Section III. The impact of such gate-level errors accumulates as the input propagates to the final output. For example, consider a clocked ASL-based 8-bit ripple carry adder (RCA) consisting of all ASL gates operating at identical switching delay T_g, switching energy per nanomagnet E_g, and, hence, identical $\epsilon(E_g, T_g)$, as shown in Fig. 5(a). The resulting distribution $P_\eta(\eta)$ of the output error η for a 15-bit RCA is dense, as shown in Fig. 5(b) and (c), for $\epsilon(E_g, T_g) = 10^{-2}$ and $\epsilon(E_g, T_g) = 10^{-1}$, respectively. The Brute force compensation of the errors having such distributions can be computationally expensive, as discussed in Section IV-B. We propose error statistics shaping techniques to impose a structure on $P_\eta(\eta)$ to reduce the complexity of error compensation.

We exploit the error rate, energy, and delay tradeoff of the clocked ASL gates (shown in Fig. 3) to shape the distribution of error η. In particular, we control the gate-level switching delay via clock pulsewidth modulation, as described in Section II-A [28], [29]. Exploiting this degree of freedom, we propose two logic-level delay assignment steps, namely, path delay balancing (PDB) and path delay redistribution (PDR). We begin with a logic gate network with all gate delays equal to T_g. Thus, the critical paths are those with

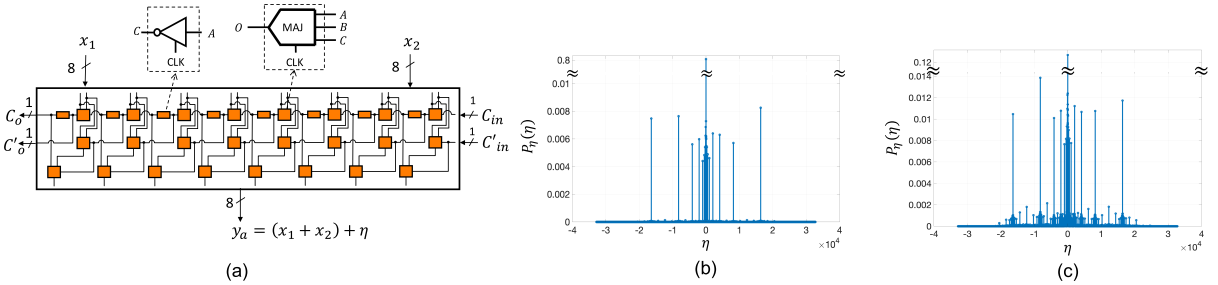


FIGURE 5. RCA with gate-level uniform error rate ϵ assignment operating at total delay of 1.24 ns. (a) Schematic of 8-bit RCA showing all gates operating at $\epsilon = \epsilon_{\text{cp-avg}} = 10^{-2}$. Error distribution $P_\eta(\eta)$ for a 15-bit RCA (b) when $\epsilon = \epsilon_{\text{cp-avg}} = 10^{-2}$ and $E_{\text{RCA15}} = 90$ fJ, and (c) when $\epsilon_{\text{cp-avg}} = 10^{-1}$ and $E_{\text{RCA15}} = 60$ fJ, where E_{RCA15} denotes total switching energy of 15-bit RCA.

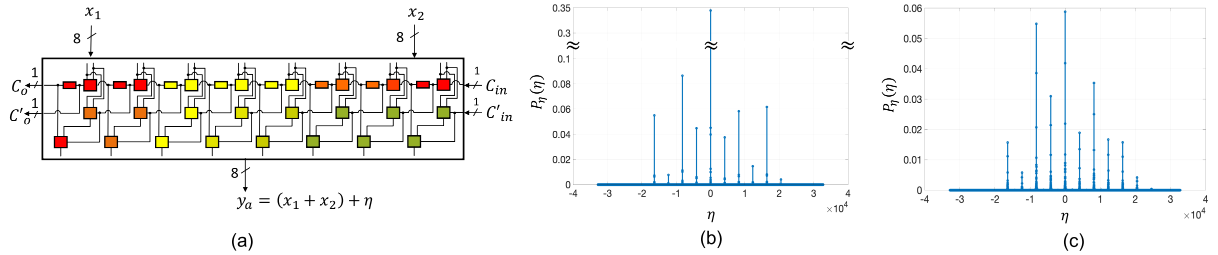


FIGURE 6. RCA with shaped error statistics operating at the total delay of 1.24 ns. (a) Schematic of an 8-bit RCA illustrating spatial distribution of gate-level ϵ after applying PDB and PDR. Error distribution $P_\eta(\eta)$ for a 15-bit RCA (b) when $\epsilon_{\text{cp-avg}} = 10^{-2}$ and $E_{\text{RCA15}} = 90$ fJ, and (c) when $\epsilon_{\text{cp-avg}} = 10^{-1}$ and $E_{\text{RCA15}} = 60$ fJ. The colors in (a) approximately convey the error rates of the gates as per the color code in Fig. 3.

the maximum number of gates N_{cp} and, therefore, have the path delay $T_{\text{cp}} = T_g N_{\text{cp}}$. In PDB and PDR steps, the gate delays are reassigned at a constant switching energy (per nanomagnet) of E_g (moving vertically in Fig. 3) and at a constant throughput (identical critical path delay T_{cp}) as follows.

1) PDB

In PDB, delays of gates lying on the shorter paths are increased, at a constant energy E_g, making every gate to lie on one or more critical paths. Thus, PDB reduces the error rate of the Xgates on shorter paths while leaving the original critical path unaltered and now containing gates with the highest error rates.

2) PDR

In PDR, the gate delays along all critical paths are further redistributed to further enhance the sparsity of $P_\eta(\eta)$ while keeping their path delay constant. In particular, the delays of the few gates in the middle of the critical path are increased (lowering ϵ) at the expense of the reduction in the delays (increasing ϵ) of the gates lying at the beginning and at the end of the critical path. Such delay redistribution increases the error rates of the top few MSBs and bottom few LSBs while reducing the error rates of the other bits in the middle. Doing so results in the increased probability of errors having extreme magnitudes (both very high and very low), leading to a highly sparse $P_\eta(\eta)$.

Section IV in the Supplementary Material describes PDB and PDR algorithms in detail. We define the average device error rate of the clocked ASL network as $\epsilon_{\text{cp-avg}} = \epsilon(E_g, T_{\text{cp-avg}})$, where $T_{\text{cp-avg}} = T_{\text{cp}}/N_{\text{cp}}$. Note that $T_{\text{cp-avg}} = T_g$ when all gates on the critical path have equal delay. Fig. 6(a) illustrates the spatial distribution in gate-level switching error rates (employing the color code from Fig. 3) for an 8-bit clocked ASL-based RCA after applying both PDB and PDR. The resulting $P_\eta(\eta)$ for a 15-bit RCA subject to PDB and PDR is shown in Fig. 6(b) and (c) for $\epsilon_{\text{cp-avg}} = 10^{-2}$ and $\epsilon_{\text{cp-avg}} = 10^{-1}$, respectively. Compared to the distributions in Fig. 5(b) and (c), the distributions in Fig. 6(b) and (c) are sparse, i.e., they have distinct well-separated peaks with relatively smaller spread around them.

Next, we show that the error statistics shaping via PDB and PDR preserves the information in the erroneous output y_a about the correct output y_o, which can be quantified via the MI $I(Y_a; Y_o)$. We empirically estimate $I(Y_a; Y_o)$ for the 15-bit RCA example in Figs. 5 and 6. For an error-free RCA, $I(Y_a; Y_o) = 13.98$ bits, which drops to 6.18 bits, with all gates are operating at an identical error rate of $\epsilon_{\text{cp-avg}} = 10^{-1}$. The resulting $P_\eta(\eta)$ in Fig. 5(c) is dense. The shaped error statistics in Fig. 6(c) enhances MI $I(Y_a; Y_o)$ to 11.15 bits. Noted that there exist multiple methods of shaping $P_\eta(\eta)$ to increase the MI. Furthermore, a high value of $I(Y_a; Y_o)$ only guarantees the existence of an error compensation scheme to reliably recover y_o from y_a. However, such scheme need not to be efficient. In Section IV-B, we derive a near-optimal low-complexity error compensation scheme that exploits the sparsity of $P_\eta(\eta)$.

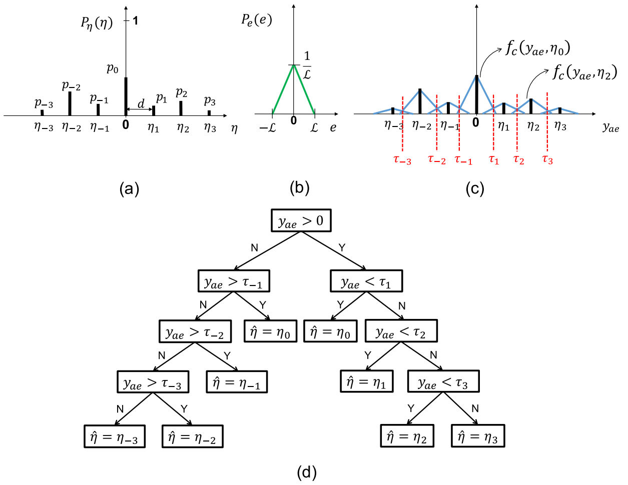


FIGURE 7. ML error compensation. (a) Illustrative $P_\eta(\eta)$ consisting of seven distinct peaks. (b) Illustrative $P_e(e)$. (c) Corresponding $f_c(y_{ae},\eta_i)$ defined in (8). (d) TreeCompensator, the resulting ML error compensator having a decision tree structure.

B. MAXIMUM LIKELIHOOD ERROR COMPENSATOR

The role of the fusion block in SEC is to compute the estimate $\hat{y}$ of the correct output y_o, as a function of two error-prone observations y_a and y_e [see Fig. 2(a)]. One approach to make $\hat{y}$ a *good* estimate of y_o is to choose $\hat{y}$, such that it maximizes the likelihood of the observations y_a and y_e, as follows:

$$\hat{y} = \arg\max_{y} P_{Y_a,Y_e|Y_o}\left\{Y_a = y_a,\, Y_e = y_e \middle| Y_o = y\right\} \qquad (5)$$

where $P_{Y_a,Y_e|Y_o}$ denotes the likelihood of Y_a and Y_e given Y_o and y denotes a free variable in the maximization that is swept over the range of possible values of correct output y_o. Thus, $\hat{y}$ is an ML estimate of y_o. In general, it can be computationally expensive to compute and maximize $P_{Y_a,Y_e|Y_o}$. However, the error statistics shaping described in Section IV-A significantly reduces the computation of the ML estimate $\hat{y}$, as shown in the following.

Noting the independence of η and e conditioned on Y_o in (5), we get

$$\hat{y} = \arg\max_{y} P_\eta(y_a - y)P_e(y_e - y) \qquad (6)$$

and we employ parametric models for $P_\eta(\eta)$ and $P_e(e)$ [35], as shown in Fig. 7(a) and (b), respectively, to simplify (6) to

$$\hat{y} = y_a - \hat{\eta} \qquad (7)$$

with $\hat{\eta}$ given as

$$\hat{\eta} = \arg\max_{\eta_i}\left[\underbrace{p_i \mathbb{1}_{\{\eta_i-\mathcal{L}<y_{ae}<\eta_i+\mathcal{L}\}} f_e(-y_{ae}+\eta_i)}_{f_c(y_{ae},\eta_i)}\right] \qquad (8)$$

where $y_{ae} = y_a - y_e = \eta - e$, $\Pr\{\eta = \eta_i\} = p_i$, $\min_{i,j}|\eta_i - \eta_j| = d$, $\Pr\{|e| < \mathcal{L}\} = 1$, and f_e denotes a functional description of P_e when $|e| < \mathcal{L}$. Detailed derivation of (7) and (8) is given in Section I in the Supplementary Material.

Given y_a and y_e, a Brute force computation of the ML estimate $\hat{y}$ requires evaluating (7) by calculating the RHS of (8) for every η_i and selecting $\eta_i = \hat{\eta}$ that maximizes it. Fig. 7(c) illustrates the plots of $f_c(y_{ae},\eta)$ as a function of y_{ae} for all values of η. It can be observed that $\hat{\eta}$ can be approximately computed via comparisons of y_{ae} with thresholds τ_is. Thus, the ML error compensator has a decision tree structure, as shown in Fig. 7(d), and is henceforth referred to as a TreeCompensator. The thresholds τ_is in the TreeCompensator are the function of error distributions P_η and P_e. For a given implementation, these distributions can be characterized once, either during simulations, or during one-time calibration phase of the prototype chip. Once the thresholds are computed offline and stored, the TreeCompensator can be implemented efficiently using only a few subtracters.

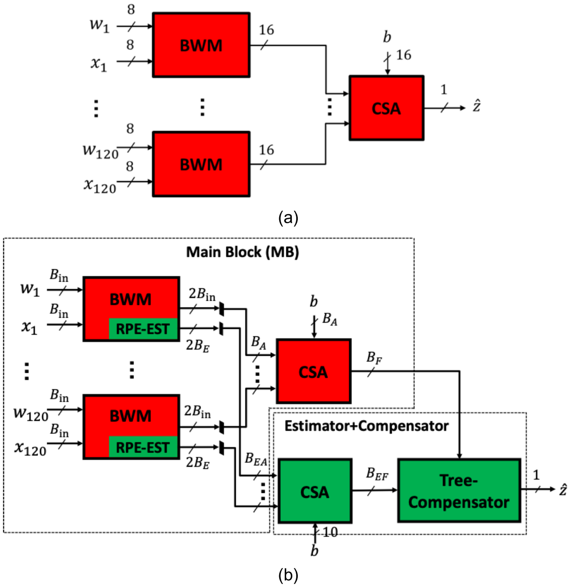


FIGURE 8. Digital clocked ASL-based 120-D SVM classifiers. (a) Conventional serial architecture with uniform delay assignments. (b) Shannon-inspired architecture.

C. DIGITAL CLOCKED ASL-BASED DOT-PRODUCT IMPLEMENTATIONS

Fig. 8(a) shows the conventional serial architecture of an 120-D SVM classifier. It employs 8-bit signed Baugh–Wooley multipliers (BWMs) and a carry save adder (CSA). All gates in this architecture operate at identical error rates. The Shannon-inspired architecture in Fig. 8(b) employs the conventional serial architecture as the MB and applies PDB and PDR to shape its output error distribution. Since PDB and PDR techniques make some gates operate at lower error rate, few reliable intermediate signals in BWMs can be employed as the estimates of the BWM outputs indicated via green reduced-precision embedded estimator (RPE-EST)

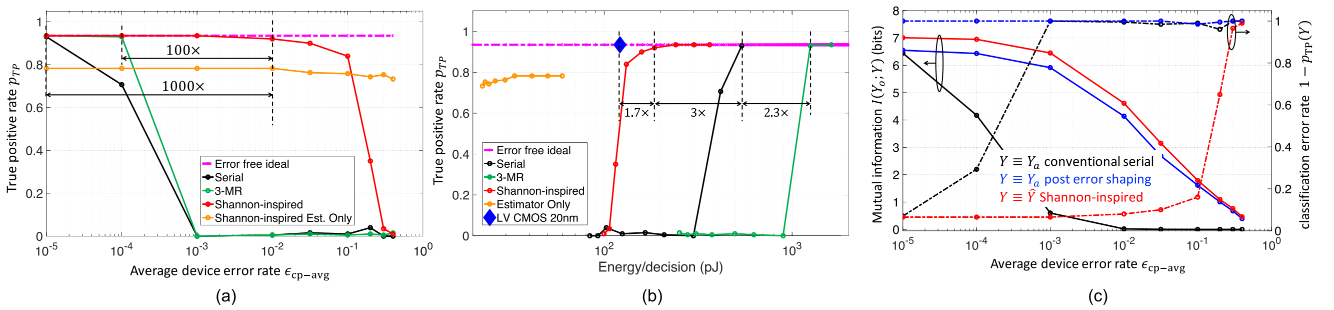


FIGURE 9. **Accuracy versus energy, error rate tradeoff for different digital clocked ASL-based 120-D SVM classifier implementations operating at a fixed decision delay and p_{FA} of 1%. (a) TP rate p_{TP} versus average device error rate $\epsilon_{\text{cp-avg}}$. (b) p_{TP} versus total classifier energy per decision. (c) MI $I(Y_o; Y)$ and corresponding classification error rate $(1 - p_{\text{TP}})$ versus $\epsilon_{\text{cp-avg}}$ curves for serial architecture (black line), after shaping its error η statistics (blue line), and Shannon-inspired architecture (red line).**

blocks in BWMs, similar to techniques discussed in [36] to reduce the estimator overhead. The additional overhead consists of a CSA and a digital clocked ASL implementation of the TreeCompensator derived in Section IV-B to compute the error compensated output $\hat{y}$. The bit precisions in the estimator and the compensator blocks are primarily dictated by the number of dominant peaks in the sparse shape of the η distribution of the MB. The CSA and the compensator overhead amount to 11% of the gate complexity of the MB. We assume a low error rate $\epsilon = 10^{-4}\epsilon_{\text{cp-avg}}$ for all the gates in the CSA and the TreeCompensator [marked green in Fig. 8(b)]. We assume that the TreeCompensator computation can be pipelined since it operates only on the final outputs of the MB and the estimator. This allows the gates in the TreeCompensator to operate at lower energy since its critical path is shorter than that of the MB. More details of the Shannon-inspired architecture are described in Section II in the Supplementary Material.

V. SIMULATION RESULTS

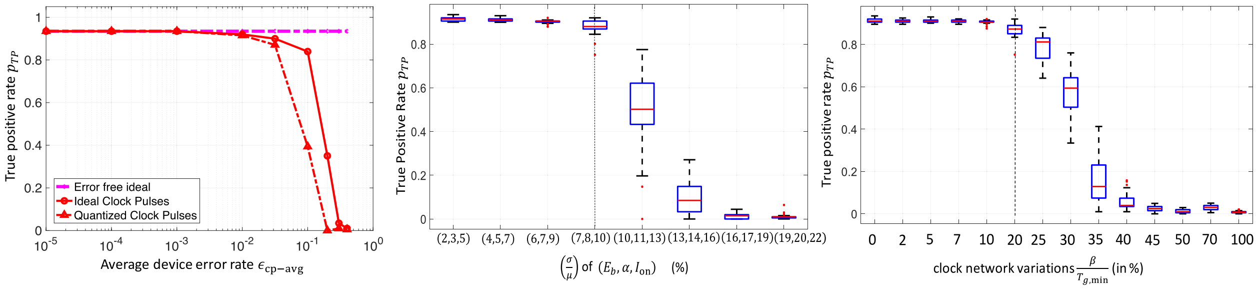
We demonstrate the benefits of the Shannon-inspired model of computation for a digital clocked ASL architecture of SVM classifier used for the electroencephalogram (EEG)-based seizure detection. The accuracy of the classifier is captured in terms of true positive (TP) rate p_{TP} and false alarm (FA) rate p_{FA}, where $p_{\text{TP}} = \Pr\{\hat{z} = 1|z = 1\}$ and $p_{\text{FA}} = \Pr\{\hat{z} = 1|z = 0\}$, and the probabilities are estimated empirically (via leave-one-out cross validation) [39] for the MIT-CHB EEG data set [40] by running extensive Monte Carlo simulations. We compare the Shannon-inspired architecture [see Fig. 8(b)] with: 1) clocked ASL-based conventional serial architecture [see Fig. 8(a)] consisting of 54 332 gates; 2) clocked ASL-based 3-MR architecture that which replicates the conventional serial architecture thrice and takes a bitwise majority vote on their outputs; and 3) 20-nm LV CMOS architecture that consists of exact same full adder-level logic network as that of the serial architecture. We compare p_{TP} versus energy per decision and $\epsilon_{\text{cp-avg}}$ tradeoffs at a fixed decision delay of 9.7 ns and $p_{\text{FA}} = 1\%$. Detailed simulation methodology is described in Section III in the Supplementary Material.

A. ACCURACY VERSUS $\epsilon_{\text{cp-avg}}$ AND ENERGY TRADEOFF

We observe in Fig. 9(a) that the Shannon-inspired architecture [see Fig. 8(b)] can tolerate 1000× higher $\epsilon_{\text{cp-avg}}$ compared to the conventional serial architecture [see Fig. 8(a)] while maintaining p_{TP} close to that of the fixed point ideal error-free architecture. In particular, p_{TP} for the Shannon-inspired architecture is close to 93% even though $\epsilon_{\text{cp-avg}}$ is as high as 1%. The 3-MR architecture tolerates an $\epsilon_{\text{cp-avg}}$ up to 0.01%. It is greater than that of the serial architecture but worse by 100× when compared to the Shannon-inspired architecture. Furthermore, we show that the intermediate estimator-only output [y_e in Fig. 8(b)] achieves lower accuracy, emphasizing the requirement to combine the two erroneous outputs [y_a and y_e in Fig. 8(b)] to achieve close-to-ideal accuracy.

The Shannon-inspired architecture achieves a 3× lower energy compared to the conventional serial architecture [see Fig. 9(b)] while maintaining $p_{\text{TP}} = 93\%$. The 3-MR architecture, however, consumes 2.3× more energy than the serial architecture even though it operates at higher device error rate. This is because the energy overhead of replication offsets the energy reduction achieved by operating at a higher device error rate. However, despite its high error tolerance, the Shannon-inspired architecture still requires 1.7× more energy compared to the 20-nm LV CMOS architecture, pointing to the need to explore devices with improved energy versus error rate tradeoffs and/or the use of increasingly powerful SEC techniques [41]–[43]. We also note in Fig. 9(b) that the estimator block [consisting only of the green CSA block in Fig. 8(b)] consumes 20% of the total energy [“Estimator Only” curve in Fig. 9(b)].

The reason for the effectiveness of the Shannon-inspired model in compensating for errors is the enhancement in MI $I(Y_o; Y_a)$ due to error statistics shaping via PDB and PDR, as shown in Fig. 9(c). Despite error statistics shaping, y_a remains a poor estimate of y_o, as evident from its high classification error rate $(1 - p_{\text{TP}})$. Since $I(Y_o; Y_a)$ is high, it implies that Y_o can be estimated accurately from Y_a. However, such an error compensator need not be efficient. Hence, in the Shannon-inspired model, we rely on two error-prone



(a)

(b)

(c)

FIGURE 10. **Impact of nonidealities and process variations on the Shannon-inspired implementation. (a) p_{TP} versus $\epsilon_{\text{cp-avg}}$ tradeoff for the Shannon-inspired implementation having 46 distinct clock pulsewidths. (b) p_{TP} box plot for different levels of static within-die process variations measured in terms of σ/μ E_b, α, and I_{ON} for the Shannon-inspired implementation having 46 distinct clock pulsewidths. (c) p_{TP} box plot as a function of extent of dynamic clock network variations $\beta/T_{g,\min}$ for the Shannon-inspired implementation having 46 distinct clock pulsewidths and σ/μ of E_b, α, and I_{ON} set at 4%, 5%, and 7%, respectively.**

observations y_a and y_e to estimate y_o both efficiently and accurately. The MI $I(Y_o; \hat{Y})$ is even higher than $I(Y_o; Y_a)$ due additional information about y_o contributed by y_e.

B. IMPACT OF NONIDEALITIES AND PROCESS VARIATIONS

Next, we evaluate the tolerance of the proposed Shannon-inspired architecture to various practical nonidealities, such as a finite number of distinct clock pulsewidths, process variations, and clock pulsewidth variations. While PDB and PDR can potentially assign a unique delay to each gate, in practice, those delays need to be further quantized to take one value out of the finite set of available distinct clock pulsewidths. Fig. 10(a) shows the p_{TP} versus $\epsilon_{\text{cp-avg}}$ curves for the Shannon-inspired architecture after quantizing the ideal clock pulsewidths to 46 distinct pulsewidths for the SVM implementation [see Fig. 8(b)] consisting of 54 332 gates. The number of distinct clock pulsewidths is of the same order as the number of gating domains explored in [28]. We observe negligible deterioration in the accuracy of the Shannon-inspired architecture (in $\epsilon_{\text{cp-avg}} < 1\%$ regime). Such gate clock pulsewidth quantization enables amortization of the clock pulse generation circuitry, including the sharing of the clocking transistors across different nanomagnets [28]. The clock network design is further simplified, since the quantized clock pulsewidths are integer multiples of the shortest reference clock, and multiple parallel dot products (in applications, such as filter banks and neural networks) can share a single clock generation circuitry.

Process variations present an additional challenge in beyond-CMOS systems. We evaluate the tolerance of the Shannon-inspired approach to static within-die variations in three device parameters, namely, energy barrier E_b, damping coefficient α of the nanomagnets, and clocking transistor ON current I_{ON}. We observe in Fig. 10(b) that the Shannon-inspired architecture with quantized clock pulsewidths can tolerate a $3(\sigma/\mu)$ variations of up to 24% in each of the three device parameters. When dynamic variations in the clock pulsewidths are included in addition to their quantization and process variations, we find in Fig. 10(c) that the Shannon-inspired architecture can tolerate a maximum deviation (β) of 20% of the minimum clock pulsewidth ($T_{g,\min}$).

VI. DISCUSSION

In this paper, we demonstrated the benefits of employing the Shannon-inspired model of computation to enhance the tolerance of digital clocked ASL implementations to random gate-level switching errors. While it improves the energy efficiency of digital clocked ASL-based implementations, the same approach can be applied to many other spintronic devices, such as MESO [3] and CoMET [4], as long as they use nanomagnet switching for information processing. The Shannon-inspired techniques have been previously applied to CMOS implementations to further reduce their energy consumption via voltage overscaling [32], [42]. In contrast, ASL/spintronics provides a new way of trading of stochasticity with energy by realizing this energy-accuracy tradeoff at the device level. The Shannon-inspired approach can enhance the ability to perform reliable computation on stochastic device fabrics to enable the use of a highly error prone but scalable physical device.

REFERENCES

[1] B. Behin-Aein, D. Datta, S. Salahuddin, and S. Datta, "Proposal for an all-spin logic device with built-in memory," *Nature Nanotechnol.*, vol. 5, no. 4, pp. 266–270, Apr. 2010.

[2] S. Manipatruni, D. E. Nikonov, and I. A. Young, "Material targets for scaling all-spin logic," *Phys. Rev. Appl.*, vol. 5, no. 1, 2016, Art. no. 014002.

[3] S. Manipatruni, D. E. Nikonov, R. Ramesh, H. Li, and I. A. Young. (2015). "Spin-orbit logic with magnetoelectric nodes: A scalable charge mediated nonvolatile spintronic logic." [Online]. Available: https://arxiv.org/abs/1512.05428

[4] M. G. Mankalale, Z. Liang, Z. Zhao, C. H. Kim, J.-P. Wang, and S. S. Sapatnekar, "CoMET: Composite-input magnetoelectric-based logic technology," *IEEE J. Explor. Solid-State Computat. Devices Circuits*, vol. 3, pp. 27–36, Dec. 2017.

[5] D. E. Nikonov and I. A. Young, "Benchmarking of beyond-CMOS exploratory devices for logic integrated circuits," *IEEE J. Explor. Solid-State Computat. Devices Circuits*, vol. 1, no. 1, pp. 3–11, Dec. 2015.

[6] W. H. Butler *et al.*, "Switching distributions for perpendicular spin-torque devices within the macrospin approximation," *IEEE Trans. Magn.*, vol. 48, no. 12, pp. 4684–4700, Dec. 2012.

[7] C. Grezes *et al.*, "Write error rate and read disturbance in electric-field-controlled magnetic random-access memory," *IEEE Magn. Lett.*, vol. 8, 2017, Art. no. 3102705.

[8] Y. Xie, B. Behin-Aein, and A. Ghosh, "Numerical Fokker-Planck simulation of stochastic write error in spin torque switching with thermal noise," in *Proc. IEEE 74th Annu. Device Res. Conf. (DRC)*, Jun. 2016, pp. 1–2.

[9] A. F. Vincent, N. Locatelli, J. O. Klein, W. S. Zhao, S. Galdin-Retailleau, and D. Querlioz, "Analytical macrospin modeling of the stochastic switching time of spin-transfer torque devices," *IEEE Trans. Electron Devices*, vol. 62, no. 1, pp. 164–170, Jan. 2015.

[10] M. Sharad, C. Augustine, G. Panagopoulos, and K. Roy, "Spin-based neuron model with domain-wall magnets as synapse," *IEEE Trans. Nanotechnol.*, vol. 11, no. 4, pp. 843–853, Jul. 2012.

[11] S. G. Ramasubramanian, R. Venkatesan, M. Sharad, K. Roy, and A. Raghunathan, "SPINDLE: Spintronic deep learning engine for large-scale neuromorphic computing," in *Proc. Int. Symp. Low Power Electron. Design*, 2014, pp. 15–20.

[12] A. Sengupta, Y. Shim, and K. Roy, "Proposal for an all-spin artificial neural network: Emulating neural and synaptic functionalities through domain wall motion in ferromagnets," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 6, pp. 1152–1160, Dec. 2016.

[13] J. Chung, J. Park, and S. Ghosh, "Domain wall memory based convolutional neural networks for bit-width extendability and energy-efficiency," in *Proc. Int. Symp. Low Power Electron. Design*, 2016, pp. 332–337.

[14] Y. Wang *et al.*, "An energy-efficient nonvolatile in-memory computing architecture for extreme learning machine by domain-wall nanowire devices," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 998–1012, Nov. 2015.

[15] Q. Dong, K. Yang, L. Fick, D. Fick, D. Blaauw, and D. Sylvester, "Low-power and compact analog-to-digital converter using spintronic racetrack memory devices," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 907–918, Mar. 2017.

[16] C. Pan and A. Naeemi, "A proposal for energy-efficient cellular neural network based on spintronic devices," *IEEE Trans. Nanotechnol.*, vol. 15, no. 5, pp. 820–827, Sep. 2016.

[17] B. Behin-Aein, V. Diep, and S. Datta, "A building block for hardware belief networks," *Sci. Rep.*, vol. 6, Jul. 2016, Art. no. 29893.

[18] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *Sci. Rep.*, vol. 7, Mar. 2017, Art. no. 44370.

[19] A. Sengupta, M. Parsa, B. Han, and K. Roy, "Probabilistic deep spiking neural systems enabled by magnetic tunnel junction," *IEEE Trans. Electron Devices*, vol. 63, no. 7, pp. 2963–2970, Jul. 2016.

[20] R. Venkatesan, S. Venkataramani, X. Fong, K. Roy, and A. Raghunathan, "Spintastic: Spin-based stochastic logic for energy-efficient computing," in *Proc. Design, Automat. Test Eur. Conf. Exhib.* San Jose, CA, USA: EDA Consortium, 2015, pp. 1575–1578.

[21] X. Jia, J. Yang, Z. Wang, Y. Chen, H. H. Li, and W. Zhao, "Spintronics based stochastic computing for efficient Bayesian inference system," in *Proc. IEEE 23rd Asia South Pacific Design Automat. Conf. (ASP-DAC)*, Jan. 2018, pp. 580–585.

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

[23] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 388–391, Mar. 1991.

[24] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 194–197, Mar. 1988.

[25] W. S. Evans and L. J. Schulman, "On the maximum tolerable noise of k-input gates for reliable computation by formulas," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 3094–3098, Nov. 2003.

[26] N. R. Shanbhag, N. Verma, Y. Kim, A. D. Patil, and L. R. Varshney, "Shannon-inspired statistical computing for the nanoscale era," *Proc. IEEE*, vol. 107, no. 1, pp. 90–107, Jan. 2019.

[27] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.

[28] Z. Pajouhi, S. Venkataramani, K. Yogendra, A. Raghunathan, and K. Roy, "Exploring spin-transfer-torque devices for logic applications," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 9, pp. 1441–1454, Sep. 2015.

[29] V. Calayir, D. E. Nikonov, S. Manipatruni, and I. A. Young, "Static and clocked spintronic circuit design and simulation with performance analysis relative to CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 2, pp. 393–406, Feb. 2014.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[31] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *Proc. 47th Design Autom. Conf.*, Jun. 2010, pp. 859–864.

[32] R. A. Abdallah and N. R. Shanbhag, "An energy-efficient ECG processor in 45-nm CMOS using statistical error compensation," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2882–2893, Nov. 2013.

[33] R. Hegde and N. R. Shanbhag, "Soft digital signal processing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 813–823, Jun. 2001.

[34] S. K. Gonugondla, B. Shim, and N. R. Shanbhag, "Perfect error compensation via algorithmic error cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 966–970.

[35] B. Shim, "Error-tolerant digital signal processing," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 2005.

[36] S. Zhang and N. R. Shanbhag, "Embedded algorithmic noise-tolerance for signal processing and machine learning systems via data path decomposition," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3338–3350, Jul. 2016.

[37] G. V. Varatkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 10, pp. 1399–1412, Oct. 2008.

[38] L. Wang and N. R. Shanbhag, "Low-power filtering via adaptive error-cancellation," *IEEE Trans. Signal Process.*, vol. 51, no. 2, pp. 575–583, Feb. 2003.

[39] A. H. Shoeb, "Application of machine learning to epileptic seizure onset detection and treatment," Ph.D. dissertation, Harvard-MIT Division Health Sci. Technol., Massachusetts Inst. Technol., Cambridge, MA, USA, 2009.

[40] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[41] E. P. Kim and N. R. Shanbhag, "Soft N-modular redundancy," *IEEE Trans. Comput.*, vol. 61, no. 3, pp. 323–336, Mar. 2012.

[42] E. P. Kim, D. J. Baker, S. Narayanan, D. L. Jones, and N. R. Shanbhag, "Low power and error resilient PN code acquisition filter via statistical error compensation," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.

[43] R. A. Abdallah and N. R. Shanbhag, "Robust and energy-efficient DSP systems via output probability processing," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Oct. 2010, pp. 38–44.