RELIABLE SPIN-BASED COMPUTING SYSTEMS

BY

AMEYA D. PATIL

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor Naresh R. Shanbhag

ABSTRACT

Scaling of logic devices has enabled tremendous improvement in computational efficiency. However, computational scaling beyond the electronics based on Moore's law requires the adoption of alternate state variables including spin. Spin based devices offer several advantages such as low device count and non-volatility, and have the potential to beat the energy-delay product of CMOS. However, thermal noise in these devices makes their switching delay a random variable. Deterministic von Neumann style computing requires them to operate at worst case delay (and low error-rate), thereby completely offsetting the energy-delay benefits of spin devices and making them non-competitive against CMOS. In this thesis, we show that, by exploiting inherent device characteristics and architectural-level techniques, it is possible to shape the system-level output error distribution, thereby enabling effective error compensation and reliable system behavior. In particular, we demonstrate that, for a simple binary classifier, 33× improvement in accuracy over conventional design can be achieved while tolerating device error rate of 10%. This work paves a way towards the design of reliable spin-based systems using highly error prone, but energy-efficient spin devices.

To my parents, for their love and continuous support.

ACKNOWLEDGMENTS

First and foremost, I would like to sincerely thank my advisor, Prof. Naresh Shanbhag, for his patient support and invaluable guidance while I carried out this research work. He has been an unequivocal source of inspiration and discussions with him have been crucial in improving my understanding of this topic, research and life in general. I would also like to thank our collaborators Sasikanth Manipatruni, Dmitri Nikonov and Ian Young from Intel for introducing us to spin-based devices and their interesting stochastic behavior, and for staying actively involved in the progress of this work.

I also thank my fellow labmates, Mingu, Sujan, Sai, Yingyan and Charbel, for all the stimulating discussions we had. Life here at UIUC would have been very difficult without the support of all my friends, both in US as well as in India, and I would like to thank them all.

Finally, I would like to extend my gratitude towards my parents, for constantly believing in me and motivating me. The credit for what I am today goes to them.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASL	All spin logic
CMOS	Complementary metal-oxide semiconductor
IPDR	Inter-path delay redistribution
IPDB	Intra-path delay balancing
LSB	Least significant bit
MSB	Most significant bit
NMR	N modular redundancy
RCA	Ripple carry adder
SEC	Statistical error compensation
WER	Write error rate

Chapter 1

INTRODUCTION

1.1 Motivation

Exponential scaling of CMOS-based logic devices in accordance with Moore's law has enabled tremendous improvement in computational efficiency. This has led to innovative designs significantly enhancing sensing, acquisition and monitoring capabilities of embedded platforms such as watches, glasses, flexible substrates etc. (see Figure 1.1(a)). These platforms have a diverse set of sensors continuously gathering a vast amount of data. However, extracting relevant information by processing the raw data in an energy-efficient manner remains a challenge. The portable embedded platforms, being severely constrained in terms of available battery power, often have very limited data-processing resources. Hence, the conventional approach is to transmit all the sensed data to the cloud to extract information. The resulting energy and latency costs are significant. For example, recent projections indicate that the traffic to the cloud consumes 9× more energy compared to that in the data center [1]. Hence, it is necessary to enhance the energy-efficiency of embedded information processing platforms (see Figure 1.1(b) without degrading their portability or battery-life.

Computational energy-efficiency via CMOS scaling has been driven by the fact that the reduction in device dimensions, accompanied by proportionate scaling of supply voltage V_{dd}, enables increased device density and reduced switching energy while keeping power density constant [4]. However, as the device dimensions reduce beyond few tens of nanometers, maintaining this scaling trend is becoming significantly more difficult [5,6]. Some of the challenges involved include increased leakage power and power density [7]. Reduction in V_{dd} requires a proportionate reduction in threshold voltage V_t in order to preserve the performance of CMOS devices [8]. Since leakage current

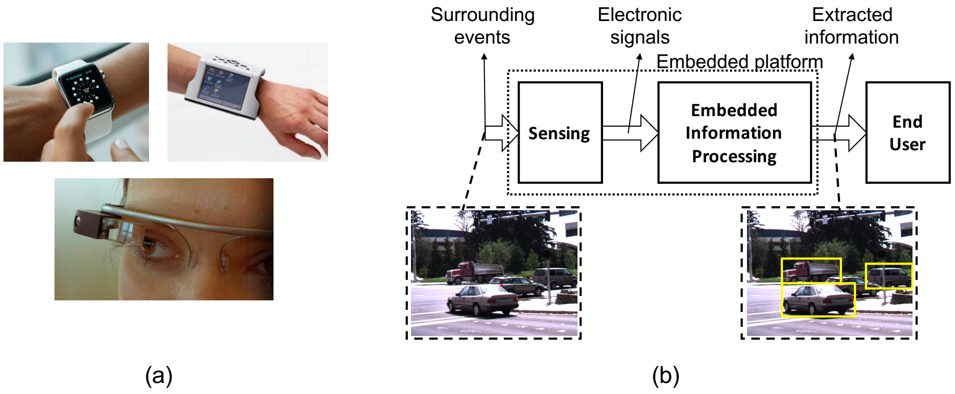


Figure 1.1: Energy-efficient computing challenge in emerging applications: (a) embedded information processing platforms (reproduced from Wikipedia under Creative Commons License), and (b) illustration of concept of embedded information processing (the images are reproduced with permission from http://vision.middlebury.edu/).

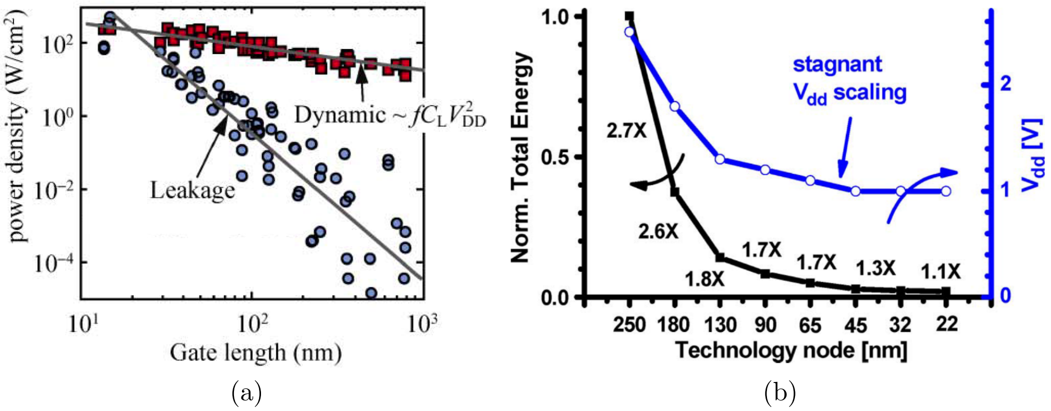


Figure 1.2: Barriers to CMOS scaling: (a) increased contribution of leakage power [2], and (b) stagnant supply voltage V_{dd} and energy scaling [3].

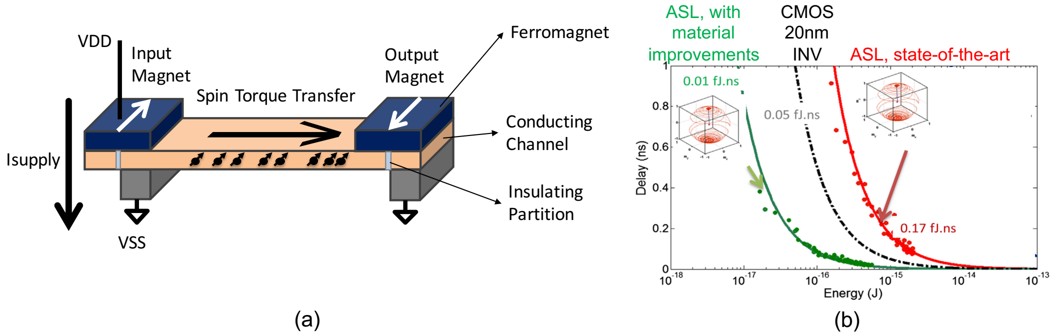


Figure 1.3: An all spin logic (ASL) device: (a) a buffer, and (b) its delay-vs-energy characteristics [13].

increases exponentially with decreasing V_t, the leakage power begins to dominate as device dimensions scale below a few tens of nanometers, as is evident from Figure 1.2(a). This stalls V_t scaling, making V_{dd} stagnant around 1 V for technology nodes beyond 65 nm, as indicated in Figure 1.2(b). Scaling device dimensions without proportionate decrease in V_{dd} causes a prohibitive increase in power density [9].

It has been argued in [10] that the power density barrier is fundamental to charge-based computing irrespective of the choice of any particular device such as CMOS. Hence, there is much interest in exploring the use of alternative state variables such as electron spin for energy-efficient computation [11]. Spintronic devices are especially considered as a promising beyond-CMOS alternative from a scaling perspective [12]. In contrast to present-day CMOS devices, spin-devices store information in terms of aligned magnetic moments (spins) of unpaired electrons in ferromagnets and rely upon spin diffusion in non-magnetic metallic channel connecting two magnets for information transfer.

1.2 Previous work

One example of spin-based devices proposed for Boolean logic computation is all spin logic (ASL) [14]. An ASL device stores binary information in the direction of magnetization state of tiny ferromagnets. Figure 1.3(a) shows a simple ASL buffer consisting of two ferromagnets separated by a nonmagnetic conducting channel. As negative supply voltage is applied to the input magnet, it polarizes the supply current passing through it; i.e., it allows only the electrons having spin aligned to its own magnetization to enter in the channel. This creates a spin concentration gradient in the channel and propagates spin alignment along the same. This, in turn, exerts torque on the magnetization of output magnet, forcing it to switch. The same device works as an inverter for positive supply voltages. All logic gates can be designed using ASL devices [15]. ASL devices offer unique advantages such as high density, low device count and non-volatility, that were previously very difficult to achieve with CMOS technology [12]. Hence, there have been multiple research efforts to build highly energy efficient computing systems using ASL [16]. However, ASL is found to be non-competitive compared to CMOS in terms of energy consumption and delay for Boolean logic implementations [17]. Some of the key reasons for ASL designs being inferior to CMOS are low switching speeds of nanomagnets, exponential decay of spin alignment propagation along the channel, and the fact that ASL gates consume power independent of activity [12, 17].

Multiple research efforts are ongoing to improve the performance of ASL gates. For channel improvements, proposals include the use of graphene [18] and automated domain walls [19]. Manipatruni et al. [13] show that the energy-delay product of an ASL based inverter can be made better than that of a 20 nm CMOS inverter with material engineering of the nanomagnets as shown in Figure 1.3(b). However, in [13], they only consider average switching delay of the nanomagnet. In fact, thermal noise in the output nanomagnets makes switching delay of an ASL gate a random variable. When operated at an average switching delay, the gate has 50% probability of not switching appropriately. Hence, although average delay of the ASL gate improves over CMOS for the same energy of operation, von Neumann style computing would require ASL gates to operate at worst case delay, in order to have negligible probability of switching failure. Worst case delay being significantly greater than average delay, ASL would remain non-competitive with CMOS even after average delay improvements, if conventional von Neumann style computing is employed.

Some research efforts have tried to exploit the randomness in the delay of ASL gates as well as their unique functionality to achieve useful system functionality [20]. For example, in [21], ASL gates are used to design energy-efficient random number generators for the implementation of spin-based stochastic computing. In [22,23], the design of a spin-based neuron has been proposed and an energy-efficient implementation of an artificial neural network has been demonstrated. Such approaches, although interesting and exploratory, do not attempt to develop key insights about controlling and quantifying the impact of switching delay variations on system-level performance and do not establish the possibility of using error-prone spin logic gates for conventional computing without significant deterioration in performance.

1.3 Thesis contributions and organization

In this thesis, we determine the energy penalty of operating ASL gates at the worst-case delay instead of at their average delay to be 25×. We also adopt an ϵ-noisy model in order to capture the impact of randomness in the switching delay on the system-level performance. Using the ϵ-noisy model, we show that it is possible to shape the statistics of errors at the system level by controlling the randomness in the component ASL gates. We propose to employ *statistical error compensation (SEC)* [24] to effectively compensate errors having shaped statistics. This enables the use of highly error prone ($\epsilon \approx 10\%$) ASL gates operating at average delay, while maintaining the system-level performance. We demonstrate that, for a binary classifier, the proposed approach achieves 33× improvement in classification error probability over conventional 7-modular redundancy based design at an average spin-device error rate of $\approx 10\%$. This result represents 100× improvement in average device error rate tolerance compared to the conventional design.

This thesis is organized as follows. In Chapter 2, we briefly review existing error-resiliency techniques that will be employed or compared with later. In Chapter 3, we introduce the ϵ-noisy device model to capture the erroneous behavior of spin devices. In Chapter 4, we propose an approach for shaping error statistics that enables reliable computation using highly erroneous ASL gates via application of statistical error compensation. We demonstrate the benefits of our approach for an ASL-based binary classifier in Chapter 5 and conclude this thesis in Chapter 6.

Chapter 2

BACKGROUND

Conventionally, digital circuits are designed with a constraint that the probability of an individual logic gate making a logic error is negligibly small [4]. Here, we define logic error as an incorrect computation/representation of one or more output bits in a hardware implementation. However, as the logic device dimensions and operating energy scale below a few nm and a few aJ, respectively, the constraint of having negligibly small probability of logic error starts becoming prohibitively expensive [25, 26]. For example, in CMOS-based digital circuits, simply reducing supply voltage till the threshold voltage results in a 5× increase in delay variations, which in turn results in timing violations if the circuit is operated at the average delay [3]. In ASL implementation, as the nanomagnet switching delay is a random variable, logic errors can occur due to the nanomagnets taking more time to switch than the clock period [26]. The conventional approach of introducing design margins to suppress the probability of logic errors often involves large penalty in terms of either energy or throughput. Hence, there have been several research attempts investigating the design of reliable systems using unreliable components, both in theory [27–29] as well as to enable highly energy-efficient practical computing systems [30–32].

In this section, we briefly review two approaches of error-resilient design, namely N-modular redundancy (NMR) and statistical error compensation (SEC).

2.1 N-modular redundancy (NMR)

NMR is a fault tolerant technique, in which a logic circuit consisting of erroneous gates is replicated N times and the final output is obtained by taking a bitwise majority vote of corresponding N outputs. For example,

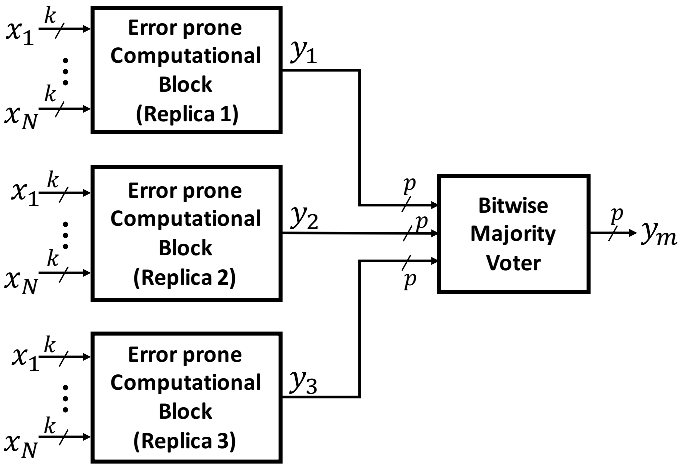

x_1
k
x_N
k
Error prone Computational Block (Replica 1)
y_1
x_1
k
x_N
k
Error prone Computational Block (Replica 2)
y_2
x_1
k
x_N
k
Error prone Computational Block (Replica 3)
y_3
p
p
p
Bitwise Majority Voter
p
y_m

Figure 2.1: Block diagram of N-Modular Redundancy (NMR) for $N = 3$.

as illustrated in Figure 2.1, when $N = 3$ (denoted as 3-MR), three identical replicas of an erroneous computational block having N k-bit inputs generate three outputs y_1, y_2 and y_3, which are p bit binary numbers. Each bit of final output y_m is generated by taking a majority vote over corresponding bits of y_1, y_2 and y_3.

This approach of using replication and majority voting to enhance error resiliency of logic circuits was first proposed by von Neumann [27]. He showed that reliable networks, i.e., networks having system-level error probability $p_{e,sys} < 0.5$, can be designed via replication of individual noisy components and majority voting of their outputs if the component error probability $\epsilon \leq 0.0073$ and that reliable computation is impossible if $\epsilon \geq \frac{1}{6}$. A series of works [28, 33] investigated tighter upper bounds on ϵ until a precise formula for threshold ϵ_o was derived by Evans and Schulman [29], such that if $\epsilon < \epsilon_o$ any Boolean function can be reliably computed using k-input ϵ-noisy gates. In practice, NMR is observed to enhance the fault tolerance effectively if the probability of component noisy gates being in error is sufficiently small (<1%) [34] albeit with at least $N\times$ area and high energy penalty. This high area and energy penalty prohibits the application of NMR as a fault tolerant technique for energy-constrained applications.

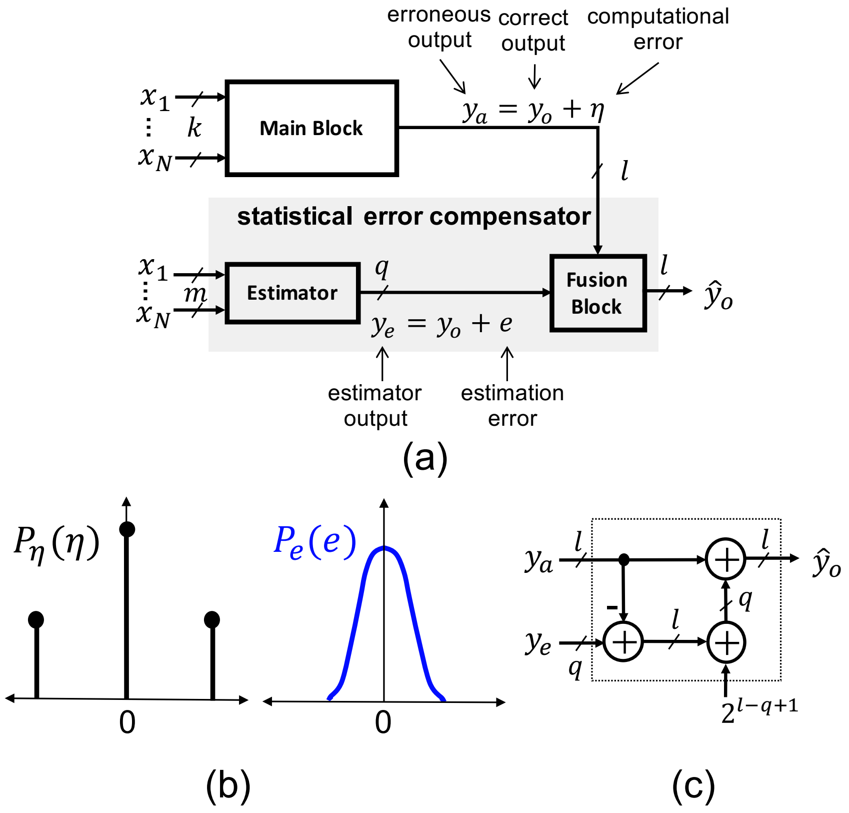


Figure 2.2: Statistical error compensation (SEC): (a) block diagram, (b) illustration of distinction in distributions of η and e for effective error compensation via a low-complexity fusion block, and (c) architecture of the fusion block implementing the algorithmic error cancellation (AEC) technique.

2.2 Statistical error compensation (SEC)

SEC is an approach where the constraint of having zero probability of logic error is relaxed in order to achieve significantly energy-efficient implementation, while maintaining system-level performance by using algorithmic techniques to compensate for logic errors [24]. Figure 2.2(a) shows a block diagram of SEC where the erroneous hardware implementation is referred to as the *main block*. The output of the main block is denoted by y_a, while the output of the corresponding error-free implementation is denoted by y_o. Computational error η captures the impact of logic errors in the hardware on the final output and is defined as the difference between y_a and y_o. It is a random variable with probability distribution $P_\eta(\eta)$. In order to compensate for η, a *statistical error compensator*, comprising an estimator and a fusion block, is introduced. The estimator is a low-complexity block (typically 5%-to-20% of the main block complexity [35]) generating a statistical estimate of y_o. Thus, we have

$$y_a = y_o + \eta \tag{2.1}$$

$$y_e = y_o + e \tag{2.2}$$

where y_e and e denote the output of the estimator and the estimation error, respectively.

The fusion block implements an estimation function, which computes an estimate of y_o (denoted by $\hat{y}_o$) in terms of y_a and y_e. Several estimation functions have been proposed [36]. In this thesis, we consider an estimation function, called algorithmic error cancellation (AEC) [37], which is given as follows:

$$\hat{y}_o = y_a - 2^{k-m+1} \left\lfloor \frac{y_a - y_e}{2^{k-m+1}} + \frac{1}{2} \right\rfloor \tag{2.3}$$

where k and m are design parameters of the main block and the estimator (bit precisions if corresponding blocks are adders) respectively and $\lfloor\ \rfloor$ denotes floor operation. It has been shown that the above estimation function is a low-complexity approximation of maximum a-posteriori (MAP) estimation, which theoretically maximizes the posterior probability $P(\hat{y}_o = y_o | y_a, y_e)$ [36]. The complexity of the fusion block is quite small, as evident in Figure 2.2(c), which shows the corresponding architecture. For AEC to be effective, the estimation error e needs to be bounded while distribution of η needs to be sparse, as illustrated in Figure 2.2(b).

Chapter 3

MODELING ASL DEVICES AS ϵ-NOISY DEVICES

In this chapter, we first introduce an approximate analytical expression characterizing the randomness in the switching delay of ASL devices. We then describe an ϵ-noisy model that can be used to capture the impact of random switching delay on the circuit and system-level behavior.

3.1 Switching error probability of ASL devices

The ASL gate is operated by passing the supply current I_{supply} through the input nanomagnet for duration T_g (referred to as gate delay) in order to achieve the appropriate switching of output nanomagnet, as illustrated in Figure 3.1(a). However, thermal noise in the nanomagnet makes the switching delay, denoted by T, a random variable. The switching is erroneous when T is greater than T_g. The resulting error is referred to as write error and its probability of a occurance is the write error rate (WER). Hence, $WER = \Pr\{T > T_g\}$. Naturally, if the probability density function of switching delay T is denoted by $f_T(t)$, the WER is the area under $f_T(t)$ curve for $t > T_g$, as illustrated in Figure 3.1(b). The average delay $T_{avg} = \mathbb{E}[T]$, where expectation is taken with respect to $f_T(t)$.

The WER depends upon material parameters, volume of the nanomagnet, I_{supply} as well as T_g, and its approximate analytical expression is given as follows [26]:

$$\text{WER}(T_g(ns), I_{supply}(\mu A)) = 1 - \exp\left[\frac{-\beta_1(I_{supply} - I_{crit})}{I_{supply}e^{\beta_2 T_g(I_{supply}-I_{crit})} - I_{crit}}\right] \quad (3.1)$$

where β_1, β_2 and I_{crit} are device dependent constants and their values for $E_b = 70\ kT$ are given in Table 3.1. In particular, E_b is an energy barrier

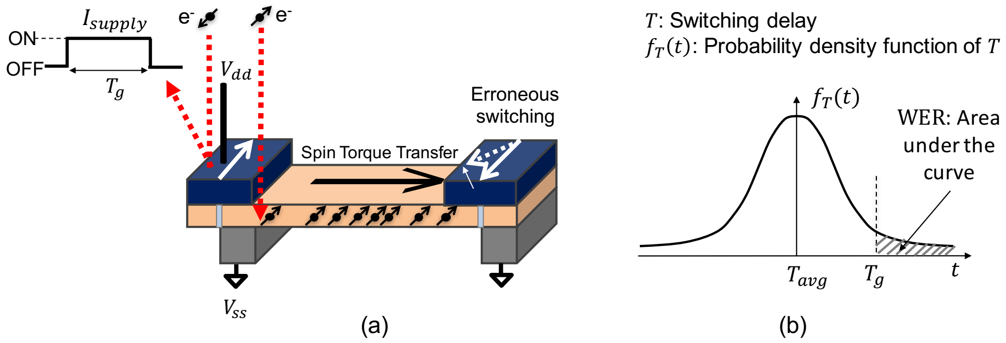


Figure 3.1: Illustration of: (a) operation of an ASL device, and (b) relationship between delay and the write error rate (WER) of an ASL device.

Table 3.1: Values of device dependent constants in (3.1)

Name	Symbol	Value	Unit
Energy barrier	E_b	$E_b = 70\ kT$	J
Critical current	I_{crit}	35.23925782850948	μA
Boltzmann's constant	k	1.38×10^{-23}	JK^{-1}
Temperature	T	300	K
-	β_1	172.7180770190638	-
-	β_2	0.06669890278819406	$\mathrm{A}^{-1}\mathrm{s}^{-1}$

between two stable states of the magnetization vector of the ferromagnet, while I_{crit} denotes the critical current required to switch the nanomagnet. This choice of E_b guarantees the probability of retention error to be smaller than one in a billion device hours of operation [12].

It is also to be noted that equation (3.1) was originally derived in [26] as a switching error rate of a ferromagnet under the influence of spin torque. By considering it as the error probability of an ASL device, we are ignoring any effects of the conduction channel between the two magnets. This assumption allows us to assume the same error model for all types of ASL-based logic gates.

3.2 Energy-robustness tradeoff for ASL devices

The energy consumption during the switching of an ASL-based logic gate is given by

$$E = I_{supply}^2 R T_g \tag{3.2}$$

where R is the resistance of logic gate and its value is assumed to be $10\,\mathrm{k}\Omega$ in this thesis, irrespective of the type of logic gates or number of inputs.

Figure 3.2(a) shows the delay vs. energy characteristics of an ASL device for a constant WER computed using equations (3.1), (3.2). The average delay for a given I_{supply} is estimated as follows:

$$T_{avg} = \int_0^\infty (1 - \mathrm{WER}(t, I_{supply}))\, dt \tag{3.3}$$

It can be observed that, for the same delay, the switching energy for an ASL gate operating at average delay (WER=0.5) is 25× smaller than when operating at a WER= 10^{-14}. This clearly indicates the energy-robustness trade-off for ASL gates, thereby underscoring the large energy penalty associated with the constraint of highly reliable operation of ASL gates.

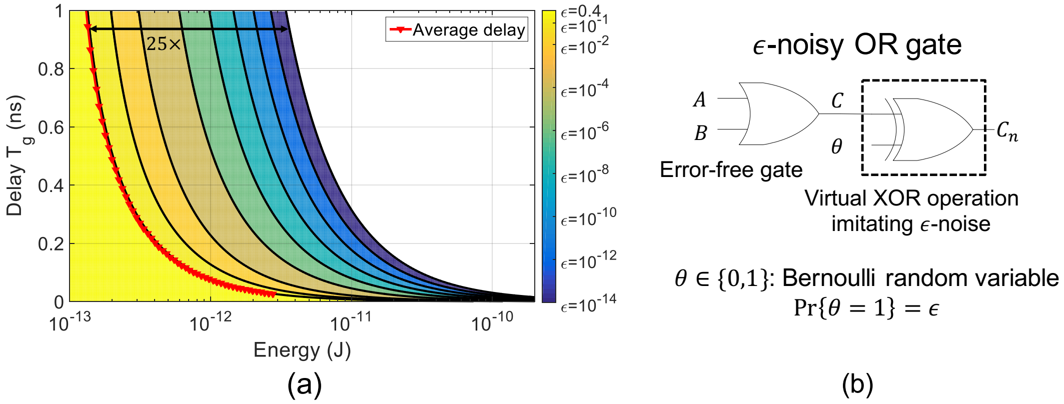


Figure 3.2: The ϵ-noisy model: (a) plot showing the trade-off between the WER ϵ, the switching energy E, and the delay T_g for an ASL device, and (b) illustration of the ϵ-noisy OR gate module.

3.3 The ϵ-noisy model for ASL logic gates

The ϵ-noisy model for logic gates was first introduced by von Neumann [27] by defining ϵ as the probability of error at the output of a given logic gate. Thus, ϵ is the error rate of the logic gate. Operation of an ϵ-noisy gate is equivalent to an XOR operation on the output of the corresponding error-free gate and a Bernoulli random variable θ with $\Pr\{\theta = 1\} = \epsilon$ as shown in Figure 3.2(b).

We adopt the ϵ-noisy gate model of Figure 3.2(b) for ASL-based logic gates by assuming $\epsilon = \text{WER}(T_g, I_{supply})$. Thus, ASL-based ϵ-noisy logic gates are unique in the way that the error rate of each individual gate can be controlled by its delay T_g as well as supply current I_{supply}. Since switching errors are caused by random thermal perturbations of magnetization state of an output nanomagnet, all ASL-based logic gates make errors independent of each other. We also assume switching errors to be independent of the inputs.

Chapter 4

SHAPING ERROR STATISTICS

In this chapter, we describe our proposed technique for significantly enhancing error-resiliency of any given logical architecture, when component ASL-based gates are highly error-prone. While there have been many research attempts exploring the theoretical aspects of achieving reliable computation using ϵ-noisy gates [28, 29, 33], they all consider the case of having identical error rate ϵ for all the component gates. However, in our approach, we exploit a physics-based approximate analytical expression of error rate ϵ for ASL-based gates to control it effectively for different logic gates via appropriate choice of gate delay T_g for each gate. Recall that gate delay T_g for ASL-based gates is the duration for which supply current is applied to the gate. Since ASL gates are non-volatile, it is more energy-efficient if ASL gates are turned on only for the time of operation [38]. Hence, the designer has direct control over the gate delay T_g. This fact is exploited to shape the error distribution in the proposed techniques.

We first describe our model of computation using ϵ-noisy gates. We then describe the proposed design technique to achieve a desired error distribution at the system level, thus paving a way towards enhanced error resiliency.

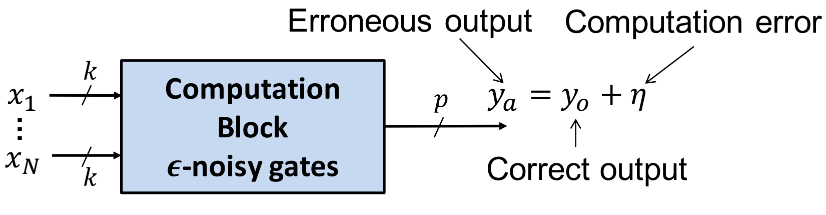


Figure 4.1: A model of computation using ϵ-noisy gates.

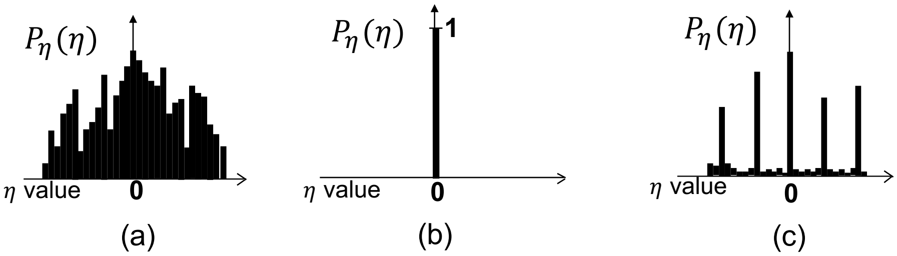


Figure 4.2: Illustration of $P_\eta(\eta)$ when: (a) all gates are highly error prone, (b) all gates are reliable, and (c) when it is shaped to be sparse.

4.1 Model of computation

Consider a computation block in Figure 4.1 consisting of ASL-based ϵ-noisy logic gates. We denote N inputs of this block as x_1, $x_2 \ldots, x_N$, where each x_i is a k-bit binary number. Since the logic gates are ϵ-noisy, the output of this computation block is erroneous and it is denoted as y_a. We define correct output y_o as the output when $\epsilon = 0$ for all gates, i.e., when all the logic gate are reliable. Adopting the definitions in the SEC framework, the computation error η is defined as the difference between erroneous output (y_a) and correct output (y_o); η is a random variable having probability mass function (PMF) $P_\eta(\eta)$. The system-level performance depends upon the distribution of η. For example, at one extreme, if all the logic gates are equally error prone with $\epsilon \approx 40\%$, the η will have a dense distribution (illustrated in Figure 4.2(a)) resulting in very low system-level performance. At the other extreme, if all the logic gates are highly reliable, the system-level performance will be maintained but will also result in very high energy consumption (corresponding in Figure 4.2(b)). We propose design techniques to shape the η statistics to be sparse (as illustrated in Figure 4.2(c)) and then use SEC-based error-compensation to achieve energy-efficient implementation while maintaining system-level performance even though constituent ϵ-noisy gates are operating at high error-rate. Sparse PMF of η enables low-complexity and effective error compensation, which is a key requirement to minimize the energy overhead of the error compensation block.

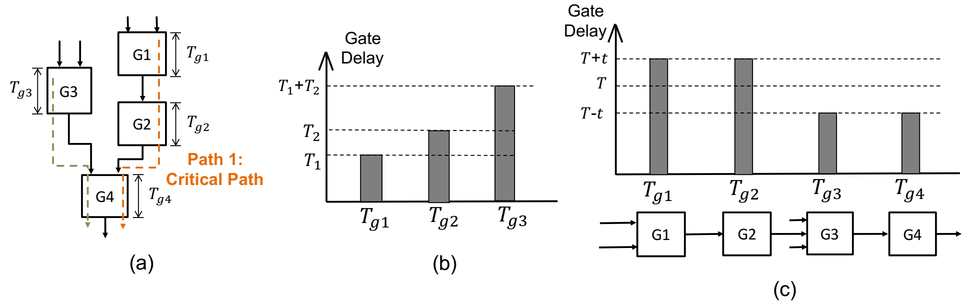


Figure 4.3: Illustration of: (a) path and critical path terminology, (b) delay assignments for inter-path delay balancing (IPDB), and (c) delay assignments for intra-path delay redistribution (IPDR).

4.2 Path delay balancing and redistribution

In this section, we propose inter-path delay balancing (IPDB) and intra-path delay redistribution (IPDR) techniques. These are the architectural-level techniques to shape the PMF of η to be sparse. These techniques enable effective error compensation using SEC.

A path in a logic circuit is a chain of cascaded logic gates starting at a primary input and ending at one of the primary outputs. A critical path in a logic circuit is the path with the maximum delay. If all gates are assumed to have the same delay, the critical path will have the maximum number of cascaded gates. Figure 4.3(a) illustrates the concept of path and critical path. We define path delay as the sum of delays of individual cascaded gates in that path while average gate delay along that path is the ratio of path delay to the number of gates along that path. Given a combinational logic circuit, we classify the constituent logic gates in two classes, namely, the logic gates in the critical path and those that are not in the critical path. We apply the IPDB technique to those gates that are not in the critical path and the IPDR technique to those gates that are in the critical path.

Definition 1. *Inter-path delay balancing (IPDB) is a technique in which the delays of all the gates that are not in the critical path are increased such that every gate lies on at least one path whose delay is equal to the critical path delay.*

Thus, an application of IPDB makes the gates on the critical path be the most error prone gates in the circuit. This is because gate error rate ϵ reduces

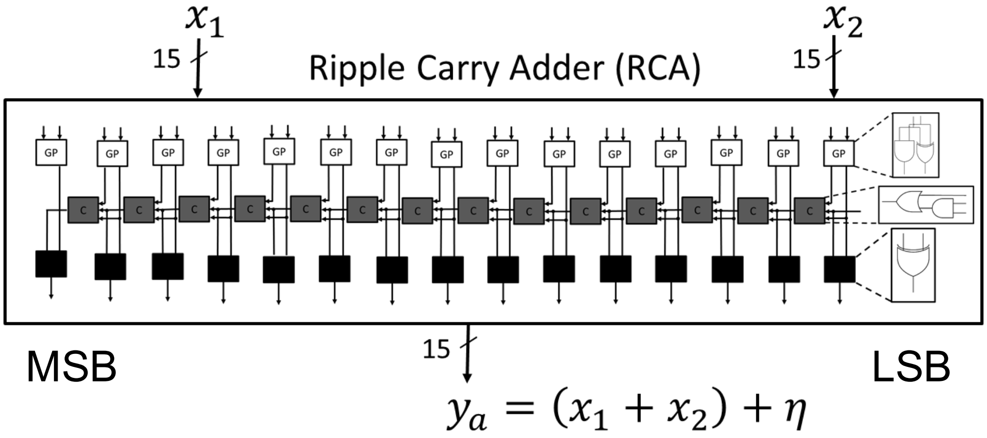


Figure 4.4: Architecture of a ripple carry adder (RCA).

exponentially with increasing gate delay T_g. Figure 4.3(b) illustrates possible delay assignments for IPDB in the logic circuit in Figure 4.3(a).

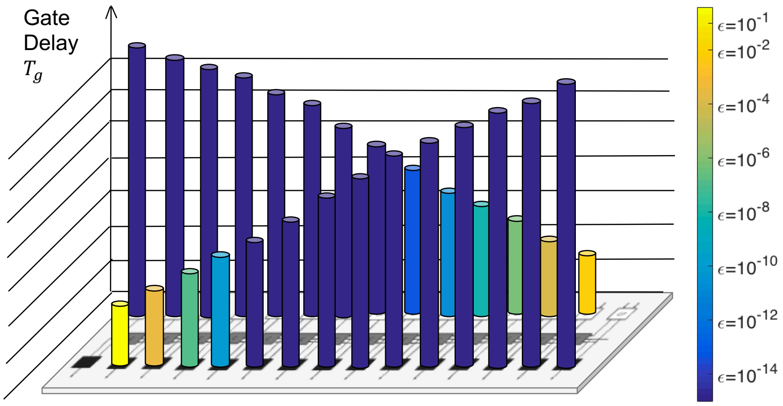
Definition 2. *Intra-path delay redistribution (IPDR) is a technique in which the delays of gates in the critical path are made unequal such that the critical path delay is unaltered. This unequal delay assignment is made such that the distribution of η becomes sparse.*

IPDR is illustrated in Figure 4.3(c), where instead of operating all gates with equal delay T, gates G_1 and G_2 are operated slower at the expense of faster operation of gates G_3 and G_4.

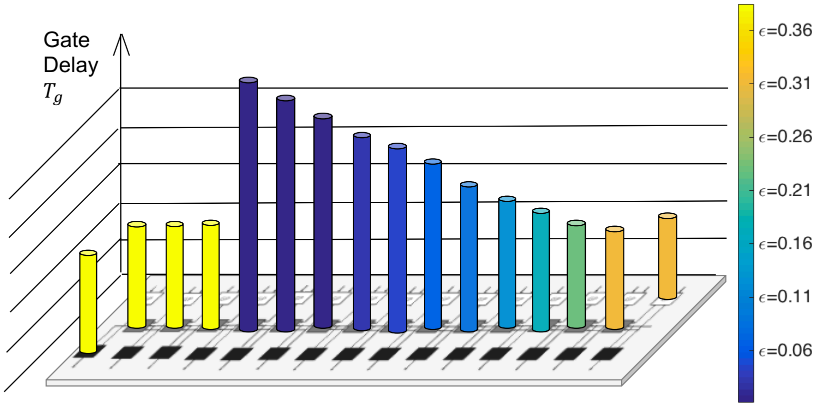
It is to be noted that once the IPDR is applied, the delays of gates on the non-critical paths need to be adjusted to keep the delays of all the paths equal. Also, even after the application of IPDB and IPDR, the average gate delay along the critical path remains the same. We define the average error rate ($\epsilon_{crit-avg}$) as the error rate corresponding to the average gate delay along the critical path ($T_{crit-avg}$), i.e. $\epsilon_{crit-avg} = \epsilon(T_{crit-avg}, I_{supply})$.

4.3 Error statistics shaping for a ripple carry adder

We now apply IPDB and IPDR techniques to a ripple carry adder (RCA) in order to demonstrate the error statistics shaping at the output of the adder. Figure 4.4 shows the architecture of a 15-bit RCA. Its critical path consists of carry evaluation blocks of individual full adders and it starts at the inputs



(a)

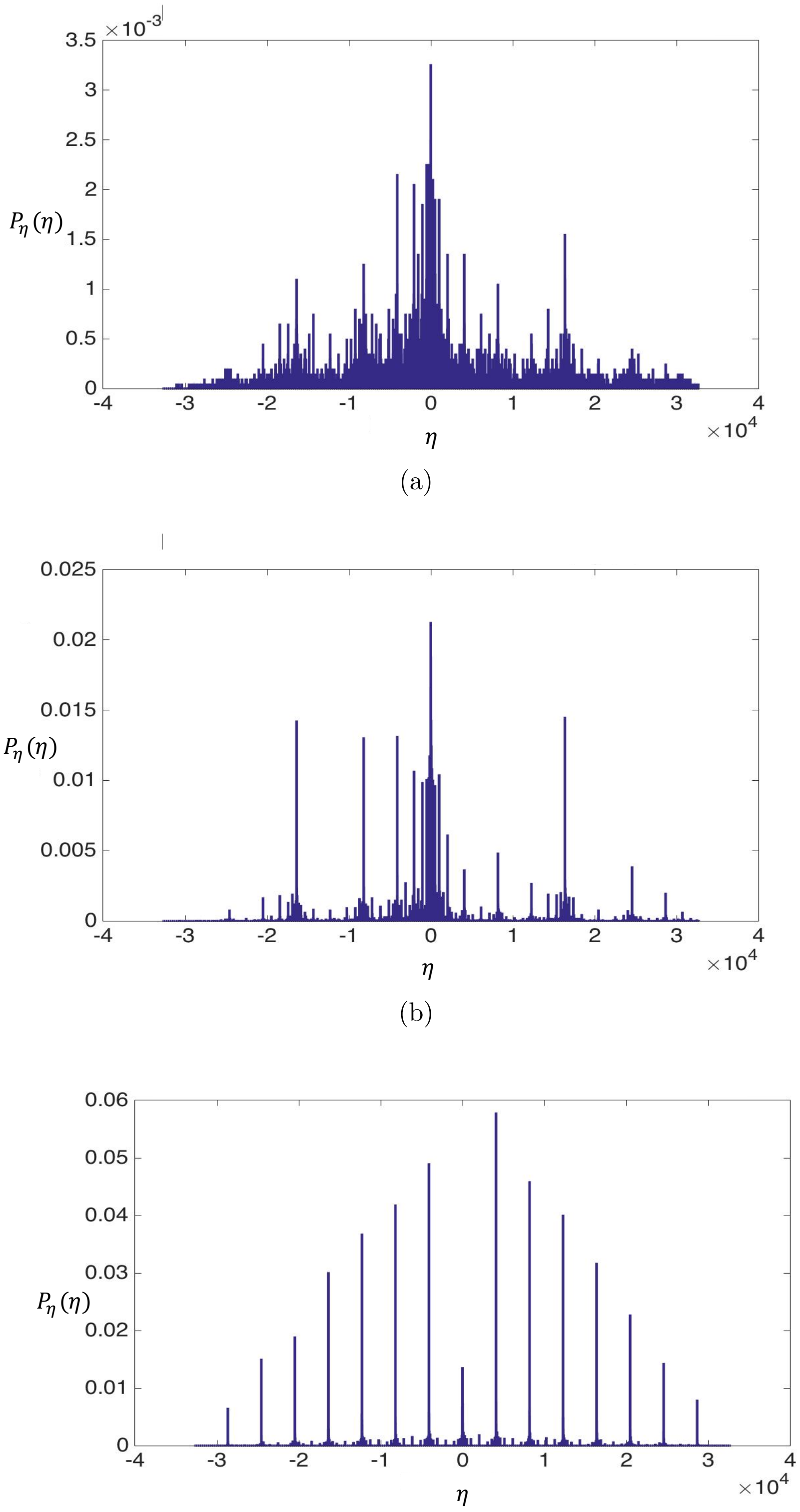


(b)

Figure 4.5: Illustration of gate delay assignments for RCA after application of: (a) IPDB for all non-critical paths, and (b) IPDR for the critical path.

of the least significant bit (LSB) full adder and terminates at the output of the most significant bit (MSB) full adder. Figure 4.5(a) illustrates the gate delay assignments for all the gates in non-critical paths after application of IPDB, while Figure 4.5(b) illustrates the same for the gates in the critical path after application of IPDR. It is to be noted that, in this case, sparse distribution is achieved by assigning smaller gate delays for the gates in the first few MSB full adders, thus making those MSBs more erroneous compared to remaining output bits.

The statistics of error η at the output of RCA under various gate delay assignments is shown in Figure 4.6. It can be observed in Figure 4.6(a) that, when all gate-delays are equal, distribution of error η is very dense, making error compensation extremely difficult and severely degrading the system-level performance. After application of IPDB, error distribution does improve partially (Figure 4.6(b)), while IPDB and IPDR together achieve the most sparse error distribution (Figure 4.6(c)). Thus, after application of IPDB and IPDR, one can employ SEC to effectively compensate the errors with shaped statistics and maintain system-level performance, as described in the next chapter.



(a)

(b)

(c)

Figure 4.6: Distribution of error η in RCA: (a) having equal gate delays, (b) with IPDB, and (c) with IPDB and IPDR for $\epsilon_{crit-avg} = 0.1$.

Chapter 5

APPLICATION: SPIN-BASED BINARY CLASSIFIER

In this chapter, we apply error statistics shaping techniques and SEC to a binary classifier in order to demonstrate the improvement in system-level performance compared to conventional and NMR-based designs.

5.1 Architecture of spin-based binary classifier

We consider the following classification problem. For two fixed k-bit binary numbers w_1 and w_2, the classification decision Z_o on k-bit input x is made as follows:

$$Z_o = \begin{cases} 0, & \text{if } x - w_1 \geq w_2 - x \\ 1, & \text{otherwise} \end{cases}$$

where $w_1 \leq x \leq w_2$. The corresponding architecture is shown in Figure 5.1. Two 1's complement and RCA blocks constitute the feature extractor (FE), which is designed using erroneous ASL gates, while the final comparator is assumed to be error-free.

Figure 5.2 shows the architecture of the SEC for a binary classifier. We apply IPDB and IPDR techniques to the FE block in order to shape the error statistics at the outputs d_{1a} and d_{2a}. We use reduced precision replica (RPR) estimator [35] so that the estimator error distribution remains bounded over smaller range. Since the critical path of the estimator is smaller than that of the main block, the ASL gates in the estimator can be run slower and, hence, at lower error rate. In particular, we make the estimator gates slow enough such that the critical path delay of the estimator becomes equal to that of the main block. For simulations, we choose $k = 15$ and $m = 4$. We also make the gates in the fusion block reliable by setting their error rate $\epsilon = 10^{-6}$. These choices are justified since both estimator and fusion block have proven to be

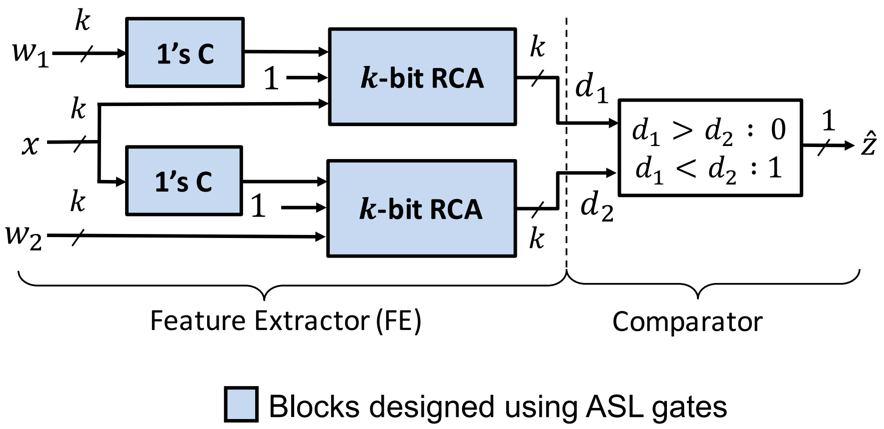


Figure 5.1: ASL-based binary classifier.

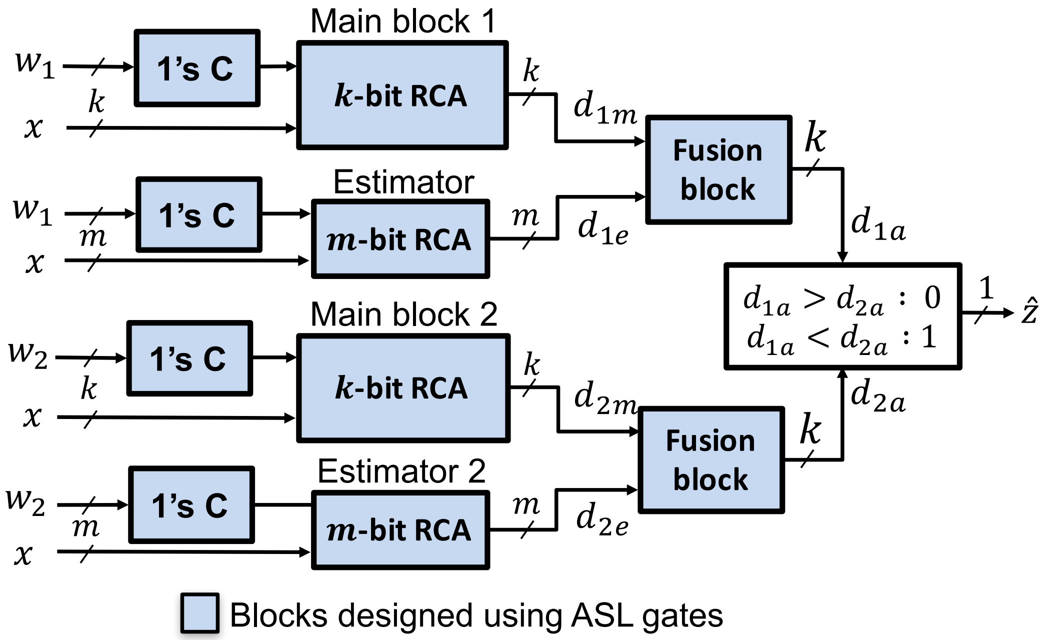


Figure 5.2: Architecture of the binary classifier with SEC.

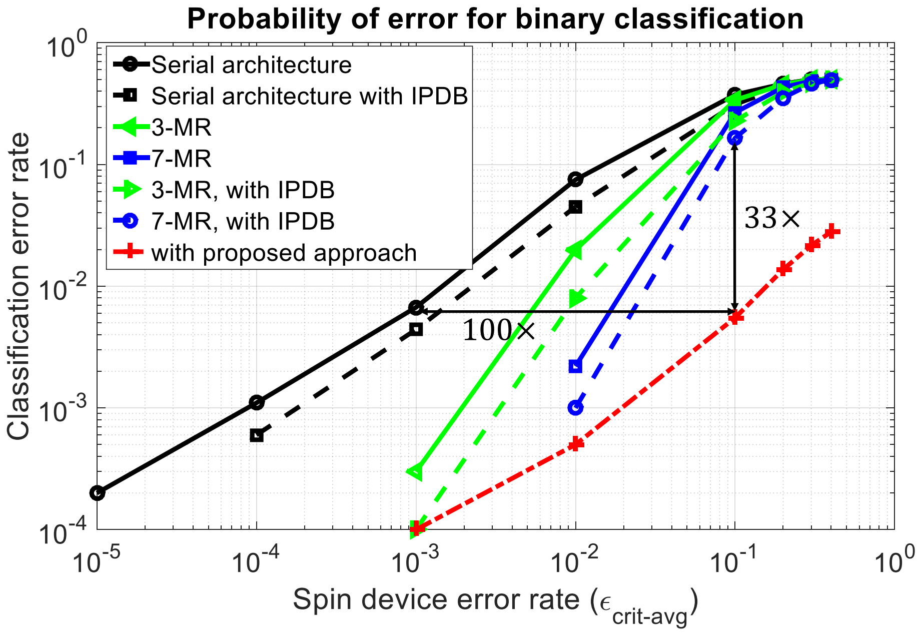


Figure 5.3: Simulation results comparing the performance of serial architecture, NMR and proposed approach of shaping error statistics and SEC.

of much smaller complexity for large system implementations [37], allowing the associated energy overhead to be amortized.

We refer to the conventional design consisting of gates having equal delays (and hence equal error rates) as the serial architecture. In the case of NMR, we replicate the serial architecture N times and take bitwise majority vote on the output of each replica. We assume a fixed error rate of $\epsilon \leq 10^{-6}$ for the component ASL gates in the majority voter.

5.2 Simulation results

We study the performance of different techniques in terms of classification error rate defined as $\Pr\{\hat{Z} \neq Z_o\}$, where $\hat{Z}$ is the decision made by the ASL-based implementation while Z_o is the correct decision. We measure the average spin device error rate of a design in terms of error rate corresponding to average gate delay along the critical path of the design ($\epsilon_{crit-avg}$).

As observed in Figure 5.3, the proposed approach of statistics shaping techniques together with SEC achieves 33× improvement in performance as compared to the NMR-based design with $N = 7$, when the average spin-device error rate is as high as 10%. This demonstrates the effectiveness of IPDB and IPDR in shaping the error statistics and of SEC in compensating for the errors. Such significant improvement in system-level performance translates into 100× improvement in the tolerance of average error rate of spin-devices compared to the conventional design while achieving the same system-level performance. In particular, the proposed approach achieves the classification error rate of 0.6% even though the average spin device error rate ($\epsilon_{crit-avg}$) is 10%. It is also to be noted that the performance of NMR-based implementations improves faster as $\epsilon_{crit-avg}$ is reduced, indicating the effectiveness of NMR at low error rates, albeit with high area and energy overhead. This result also confirms that application of only IPDB indeed improves the system-level performance marginally over the conventional design. This is because, after applying IPDB, the highly erroneous gates are restricted to the critical path while all the gates on other paths operate at lower error rate.

Chapter 6

CONCLUSION

In this thesis, we proposed an approach to shape the error statistics of a spin-based system via architectural-level delay assignment techniques. Such statistics shaping enables low-complexity, effective error compensation via the SEC framework, allowing the use of highly error prone but energy efficient devices.

This work began at device-level and culminated in achieving improvements in system-level performance. In Chapter 3 we used a physics-based analytical framework to justify the use of an ϵ-noisy model for error prone ASL devices in order to capture their behavior at circuit/system-level. In Chapter 4, we proposed delay assignment techniques, namely IPDB and IPDR, to achieve error statistics shaping at the output of a given logic network. We applied these techniques to a scalar binary classifier in Chapter 5 to demonstrate 33× improvement in robustness over NMR-based design and 100× higher device error rate tolerance over the conventional design.

Although this work establishes the prospects of maintaining system-level performance in the presence of error-prone components, the energy benefits of such technology need to be carefully studied. In particular, techniques such as IPDB and IPDR need to applied such that delay adjustments do not incur energy overheads. The cost of an additional control unit, required to appropriately control gate delays, also needs to be evaluated.

This ability to control the error statistics opens up many possibilities for future work. For example, one can exploit this ability to enhance the generalization behavior with limited training data in machine learning algorithms. One can also explore the use of noise to enhance certain information processing tasks.

REFERENCES

[1] J. Baliga, R. W. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2011.

[2] E. Pop, "Energy dissipation and transport in nanoscale devices," *Nano Research*, vol. 3, no. 3, pp. 147–169, 2010.

[3] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[4] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits.* Prentice hall Englewood Cliffs, 2002, vol. 2.

[5] A. A. Chien and V. Karamcheti, "Moore's law: The first ending and a new beginning," *Computer*, no. 12, pp. 48–53, 2013.

[6] C. Mack et al., "Fifty years of Moore's law," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 24, no. 2, pp. 202–207, 2011.

[7] S. Borkar, "Design challenges of technology scaling," *Micro, IEEE*, vol. 19, no. 4, pp. 23–29, 1999.

[8] Y. Taur, "CMOS design near the limit of scaling," *IBM Journal of Research and Development*, vol. 46, no. 2.3, pp. 213–222, 2002.

[9] R. Ronen, A. Mendelson, K. Lai, S.-L. Lu, F. Pollack, and J. P. Shen, "Coming challenges in microarchitecture and architecture," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 325–340, 2001.

[10] V. V. Zhirnov, R. K. Cavin, J. A. Hutchby, and G. I. Bourianoff, "Limits to binary logic switch scaling-a gedanken model," *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1934–1939, 2003.

[11] D. Nikonov and I. Young, "Benchmarking of beyond-CMOS exploratory devices for logic integrated circuits," *Exploratory Solid-State Computational Devices and Circuits, IEEE Journal on*, vol. 1, pp. 3 – 11, 2015.

[12] J. Kim, A. Paul, P. Crowell, S. J. Koester, S. S. Sapatnekar, J.-P. Wang, C. H. Kim et al., "Spin-based computing: device concepts, current status, and a case study on a high-performance microprocessor," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 106–130, 2015.

[13] S. Manipatruni, D. E. Nikonov, and I. A. Young, "Material targets for scaling all-spin logic," *Physical Review Applied*, vol. 5, no. 1, p. 014002, 2016.

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

[15] C. Augustine, G. Panagopoulos, B. Behin-Aein, S. Srinivasan, A. Sarkar, and K. Roy, "Low-power functionality enhanced computation architecture using spin-based devices," in *Nanoscale Architectures (NANOARCH), 2011 IEEE/ACM International Symposium on.* IEEE, 2011, pp. 129–136.

[16] M. Sharad, K. Yogendra, A. Gaud, K.-W. Kwon, and K. Roy, "Ultra-high density, high-performance and energy-efficient all spin logic," *arXiv preprint arXiv:1308.2280*, 2013.

[17] Z. Pajouhi, S. Venkataramani, K. Yogendra, A. Raghunathan, and K. Roy, "Exploring spin-transfer-torque devices for logic applications," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 34, no. 9, pp. 1441 – 1454, 2015.

[18] M. Zeng, L. Shen, H. Su, C. Zhang, and Y. Feng, "Graphene-based spin logic gates," *Applied Physics Letters*, vol. 98, no. 9, p. 092110, 2011.

[19] S.-C. Chang, S. Dutta, S. Manipatruni, D. E. Nikonov, I. A. Young, and A. Naeemi, "Interconnects for all-spin logic using automotion of domain walls," *Exploratory Solid-State Computational Devices and Circuits, IEEE Journal on*, vol. 1, pp. 49–57, 2015.

[20] K. Roy, D. Fan, X. Fong, Y. Kim, M. Sharad, S. Paul, S. Chatterjee, S. Bhunia, and S. Mukhopadhyay, "Exploring spin transfer torque devices for unconventional computing," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 5, no. 1, pp. 5–16, 2015.

[21] R. Venkatesan, S. Venkataramani, X. Fong, K. Roy, and A. Raghunathan, "Spintastic: spin-based stochastic logic for energy-efficient computing," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition.* EDA Consortium, 2015, pp. 1575–1578.

[22] D. Fan, M. Sharad, A. Sengupta, and K. Roy, "Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing," *arXiv preprint arXiv:1402.2902*, 2014.

[23] K. Yogendra, D. Fan, and K. Roy, "Coupled spin torque nano oscillators for low power neural computation," *Magnetics, IEEE Transactions on*, vol. 51, no. 10, 2015.

[24] N. R. Shanbhag, R. A. Abdallah, R. Kumar, and D. L. Jones, "Stochastic computation," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 859–864.

[25] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *Micro, IEEE*, vol. 25, no. 6, pp. 10–16, 2005.

[26] W. Butler, T. Mewes, C. K. Mewes, P. Visscher, W. H. Rippard, S. E. Russek, and R. Heindl, "Switching distributions for perpendicular spin-torque devices within the macrospin approximation," *Magnetics, IEEE Transactions on*, vol. 48, no. 12, pp. 4684–4700, 2012.

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

[28] B. Hajek and T. Weller, "On the maximum tolerable noise for reliable computation by formulas," *IEEE Transactions on Information theory*, vol. 37, no. 2, pp. 388–391, 1991.

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

[30] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Proceedings of the 1999 International Symposium on Low Power Electronics and Design*. ACM, 1999, pp. 30–35.

[31] G. V. Varatkar and N. R. Shanbhag, "Error-resilient motion estimation architecture," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 10, pp. 1399–1412, 2008.

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

[33] N. Pippenger, "Reliable computation by formulas in the presence of noise," *Information Theory, IEEE Transactions on*, vol. 34, no. 2, pp. 194–197, 1988.

[34] M. L. Shooman, "N-modular redundancy," *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*, pp. 145–201, 2002.

[35] B. Shim, S. R. Sridhara, and N. R. Shanbhag, "Reliable low-power digital signal processing via reduced precision redundancy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 497–510, 2004.

[36] B. Shim, "Error-tolerant digital signal processing," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2005.

[37] S. K. Gonugondla, B. Shim, and N. R. Shanbhag, "Perfect error compensation via algorithmic error cancellation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 966–970.

[38] 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," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 61, no. 2, pp. 393–406, 2014.