New Metrics for the Reliability of Approximate and Probabilistic Adders

Jinghang Liang, *Student Member*, Jie Han, *Member*, Fabrizio Lombardi, *Fellow IEEE*

Abstract—Addition is a fundamental function in arithmetic operation; several adder designs have been proposed for implementations in inexact computing. These adders show different operational profiles; some of them are approximate in nature while others rely on probabilistic features of nanoscale circuits. However, there has been a lack of appropriate metrics to evaluate the efficacy of various inexact designs. In this paper, new metrics are proposed for evaluating the reliability as well as the power efficiency of approximate and probabilistic adders. Reliability is analyzed using the so-called sequential probability transition matrices (SPTMs). Error distance (ED) is initially defined as the arithmetic distance between an erroneous output and the correct output for a given input. The mean error distance (MED) and normalized error distance (NED) are then proposed as unified figures that consider the averaging effect of multiple inputs and the normalization of multiple-bit adders. It is shown that the MED is an effective metric for measuring the implementation accuracy of a multiple-bit adder and that the NED is a nearly invariant metric independent of the size of an adder. The MED is, therefore, useful in assessing the effectiveness of an approximate or probabilistic adder implementation, while the NED is useful in characterizing the reliability of a specific design. Since inexact adders are often used for saving power, the product of power and NED is further utilized for evaluating the tradeoffs between power consumption and precision. Although illustrated using adders, the proposed metrics are potentially useful in assessing other arithmetic circuit designs for applications of inexact computing.

Index Terms— Adders, Inexact computing, Reliability, Error masking, Approximate logic, Imprecise arithmetic, Mean error distance, Normalized error distance, Power, Energy efficiency.

1 Introduction

Methodologies for inexact (or soft) computing rely on the feature that many applications can tolerate some loss of precision and therefore, the solution can tolerate some degree of uncertainty [1-3]. Deterministic, explicit, and precise models and algorithms are not always suitable to solve these types of problems. However, inexact computing applications are mostly implemented using digital binary logic circuits, thus operating with a high degree of predictability and precision. A framework based on a precise and specific implementation can still be used with a methodology that intrinsically has a lower degree of precision and an increasing uncertainty in operation. While this may be viewed as a potential conflict, such an approach tailors the significant advantage of inexact computing (and its inherent tolerance to some imprecision and uncertainty) to a technology platform implemented by conventional digital logic and systems. The paradigm of inexact computation relies on relaxing fully precise and completely deterministic building blocks (such as a full adder) when for example, implementing bio-inspired systems. This allows nature inspired computation to redirect the existing design process of digital circuits and systems by taking advantage of a decrease in complexity and cost with possibly a potential increase in performance and power efficiency. In imprecise computation, however, traditional measures for performance, power and reliability are often conflicting, so new figures of merit for assessing the tradeoffs involved in such design process are needed to better understand the operation of approximate/probabilistic circuits.

One of the fundamental arithmetic operations in many applications of inexact computing is addition [4, 5]. Soft additions are generally based on the operation of deterministic approximate logic or probabilistic imprecise arithmetic (categorized in [6] as design-time and run-time techniques). Several recently proposed adder architectures are representatives of these types. The bio-inspired lower-part OR adder (LOA) [7] is based on approximate logic, whose truth table is slightly different from the original truth table of a full adder. The approximate mirror adders (AMAs) proposed in [8] save power by reducing the number of transistors in a mirror adder design. The use of these architectures results in approximations to the addition, making it deterministically different from the precise operational outcome. Another design is known as the probabilistic full adder (PFA) [9, 10]; its implementation is based on probabilistic CMOS, a technology platform for modeling the behavior of nanometric designs as well as reducing power consumption.

In light of these advances, design metrics are urgently needed to evaluate the effectiveness of the architectures; as shown in latter sections, the traditional metric of reliability (defined as the probability of system survival) is not appropriate for use in evaluating approximate designs. Moreover, it is useful to have a design metric with no dependency on the number of operands in an addition.

- *J. Liang and J. Han are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2V4. (E-mail: jinghang@ualberta.ca, jhan8@ualberta.ca)*
- *F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, USA. (E-mail: lombardi@ece.neu.edu)*

Manuscript received Dec. 11, 2011.

Digital Object Indentifier 10.1109/TC.2012.146

The objective of this paper is to propose new metrics for assessing adder designs with respect to reliability and power efficiency for inexact computing. A new figure of merit referred to as error distance (ED) is initially proposed to characterize the reliability of an output of an adder. ED is then used to obtain two new metrics: the mean error distance (MED) and the normalized error distance (NED). The MED and NED can be obtained using sequential probability transition matrices (SPTMs) and are able to evaluate the reliability of both probabilistic and deterministic adders. It is shown that the MED is an effective metric in evaluating the implementation of a multiple-bit adder. The NED is a stable metric that is almost independent of the size of an implementation; this feature brings a new perspective for the evaluation and comparison of different adder designs. The power and NED product is further used to evaluate the power and precision tradeoff. An adder implementation with reduced precision, referred to as the lower-bit ignored adder (LIA), is investigated as a baseline design for assessing the LOA, AMAs and PFAs. A detailed analysis and simulation results are presented to assess the reliable performance of these adders using the proposed new metrics.

Some preliminary results of this work have been previously presented in [11]; [11] introduced the concepts of ED and MED and presented a comparative study of reliability and static power for the LOA, PFAs and LIA. This paper is a significant extension of [11] and includes the following novel contributions:

- The recently proposed approximate mirror adders (AMAs) as transistor-level designs are considered and included in the analysis of the proposed metrics.
- The notion of sequential probability transition matrices (SPTMs), as well as their formulation, is presented in detail as a framework for evaluating the reliability of sequential circuits.
- A normalized error distance (NED) is proposed; as a nearly invariant metric, the NED is almost independent of the size of an adder implementation, so it is an important metric for characterizing the reliability of an approximate or probabilistic design.
- The product of power and NED is proposed to quantitatively evaluate the tradeoff of power and precision. Moreover, the power saving and NED ratio is used in the analysis of the efficiency of a design when trading off precision for power.

The rest of the paper is organized as follows. Section 2 contains a review and Section 3 presents sequential probability transition matrices (SPTMs). Section 4 presents the notion of error distance (ED) and the evaluations of mean error distance (MED) and normalized error distance (NED). Section 5 discusses the power and precision tradeoff. Section 6 concludes the paper.

2 Review

In most digital systems, sequential circuits are utilized, so this paper primarily deals with sequential adders. A sequential adder, as the one shown in Fig. 1, consists of a k-bit full adder concatenated with a k-bit register. In this section, different implementations of a full adder are reviewed; particular emphasis is devoted to features relevant to soft and inexact computing.

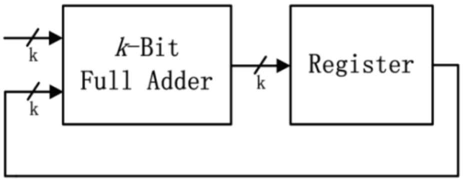


Fig. 1. A k-bit sequential adder.

2.1 Conventional Full Adder (CFA)

Fig. 2 shows a one-bit (precise) conventional full adder (CFA); the CFA is commonly connected in a ripple-carry implementation, i.e., by cascading the circuit of Fig. 2 in a linear array.

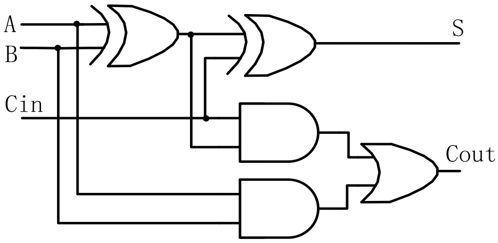


Fig. 2. A one-bit conventional full adder (CFA).

2.2 Lower-Part-OR Adder (LOA)

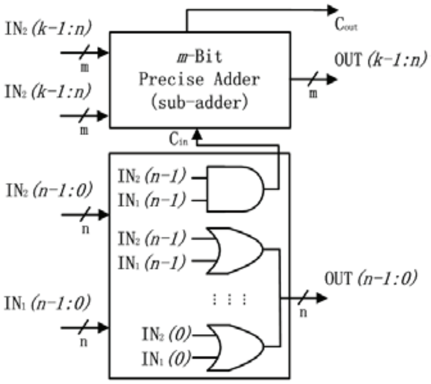


Fig. 3. Hardware structure of the lower-part-OR adder (LOA) [7].

Differently from conventional designs that strictly operate according to the exact function (as defined by its truth table), an approximate logic implementation alters some entries in the truth table. This feature allows balancing precision with other performance metrics. The recently proposed LOA is based on such a design [7]. A LOA divides a k-bit addition into two smaller parts, i.e., two modules of m-bits and n–bits. As shown in Fig. 3, the m-bit module of a LOA uses a smaller but precise adder (referred to as the *sub-adder*) to compute the exact values of the m most significant bits of the result (also referred to as the *upper part*). Additional OR gates are used to approximately compute the n least significant bits (also referred to as the lower part) of the sum by applying a bitwise OR operation on the respective input bits. An additional AND gate is used to generate the carry-in for the precise sub-adder when the most significant bits of both the lower-part inputs are "1." As this implementation ignores the "trivial" carries in the lower part of the LOA adder, it may result in a loss of precision. Albeit using different structures, the approximate adder designs in [12] and [13] belong to the same category.

2.3 Approximate Mirror Adder (AMA)

A Mirror Adder (MA) is not based on the complementary structure of CMOS logic. It is based on a special arrangement of the transistors and is yet another common design for implementing conventional adders. When approximate logic is applied to the MA cells, approaches such as IMPACT [8] have been reported to tradeoff precision for power and area. Three implementations of an approximate mirror adder (AMA) are proposed in [8] by removing different numbers of transistors. The truth table of these three approximate implementations is shown in Table 1. Similarly, AMAs can be used in the least significant n-bit (or the lower part) of an approximate sequential adder.

TABLE 1. TRUTH TABLE OF CONVENTIONAL MIRROR ADDER AND ITS APPROXIMATE IMPLEMENTATIONS [8]; ENCLOSED ENTRIES INDICATE INCORRECT OUTPUTS.

Inputs			Accurate		AMA1		AMA2		AMA3	
			Cout	Sum	Cout	Sum	Cout	Sum	Cout	Sum
0	0	0	0	0	0	①	0	0	0	0
0	0	1	0	1	0	1	0	1	0	⓪
0	1	0	0	1	①	⓪	0	⓪	0	1
0	1	1	1	0	1	0	⓪	①	⓪	①
1	0	0	0	1	0	1	①	⓪	0	⓪
1	0	1	1	0	1	0	1	0	1	0
1	1	0	1	0	1	0	1	0	1	①
1	1	1	1	1	1	⓪	1	1	1	1

2.4 Probabilistic Full Adder (PFA)

Probabilistic CMOS (PCMOS) is a technique for achieving savings in power consumption, while balancing performance at nanometric ranges. In a k-bit PFA, the most significant m-bit adders are implemented using perfectly reliable (and thus deterministic) gates, while the least significant n bits are implemented in PCMOS. Although new adder architectures have been proposed to optimize the use of PCMOS technology [14], probabilistic implementations of a conventional full adder are considered in this work. The structure of this type of adder is shown in Fig. 4.

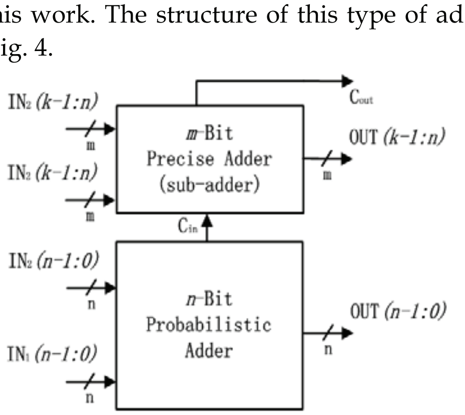


Fig. 4. Hardware structure of the probabilistic full adder (PFA).

3 Sequential Probability Transition Matrices

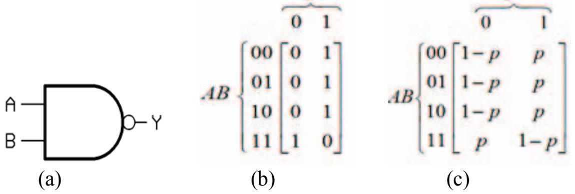
3.1 Definitions

Prior to presenting the analysis, several definitions are first introduced. In a probabilistic design, the signal probability is defined as the probability of a signal being logical "1." An error may occur to the function of a gate and the probability of this occurrence is given by a gate error rate (or probability). Unless otherwise noted, the von Neumann type of error (i.e., flipping the correct signal) is considered in this paper. In a combinational circuit, the output probability for a specific input vector i is defined as the joint probability that all of the outputs are "1" when the input vector is i. Any output that deviates from the correct value due to the effect of errors, is considered as a faulty output. The output reliability for input i is defined as the joint probability that all outputs are correct for i. This definition of reliability considers the correlation among signals and is therefore used in this paper. Circuit reliability is defined as the average output reliability over all applicable input vectors.

In an approximate design, the operation is deterministic and a faulty output is due to a functional approximation. This deterministic as well as the aforementioned probabilistic behavior can analytically be assessed for circuit reliability using the so-called sequential probability transition matrices (SPTMs), as presented next.

3.2 Probabilistic Transfer Matrices (PTMs)

The PTM approach represents a computational framework for the evaluation of circuit reliability in the presence of both deterministic and probabilistic errors [15]. A PTM is a matrix, in which the $(i,j)^{\text{th}}$ entry represents the conditional probability of the output vector of value j, determined by the circuit structure for the input vector i. Since a fault-free circuit has correct outputs with probability 1, it has an ideal transfer matrix (ITM), in which an entry is either 0 or 1. A two-input NAND gate's ITM and PTM are shown in Fig. 5.



(a) (b) (c)

Fig. 5. An ITM and PTM for a two-input NAND gate: (a) a two-input NAND gate, (b) ITM of the NAND gate, (c) PTM of the NAND gate (the gate has a probability of 1-p to produce an error).

A circuit PTM can be obtained by combining the gate PTMs using simple rules of matrix operations and the connectivity of the gates. Matrix multiplications are used for gates connected in series, while tensor products are used for gates connected in parallel. Special PTMs and ITMs are constructed for fanouts, interconnects and swapping of wires by taking into account their topologies. Since signal representation and propagation are incorporated into the formulation of PTMs, a circuit PTM contains accurate and comprehensive information about the probabilistic behavior of a circuit. A PTM can also be extended to account for more complex circuit structures. In [16], PTMs are used to investigate the effects of soft errors in sequential circuits. In this paper, sequential PTMs (SPTMs) are defined and a detailed formulation is presented by considering the topology and structure of sequential circuits for reliability evaluation.

3.3 Formulation of Sequential PTMs (SPTMs)

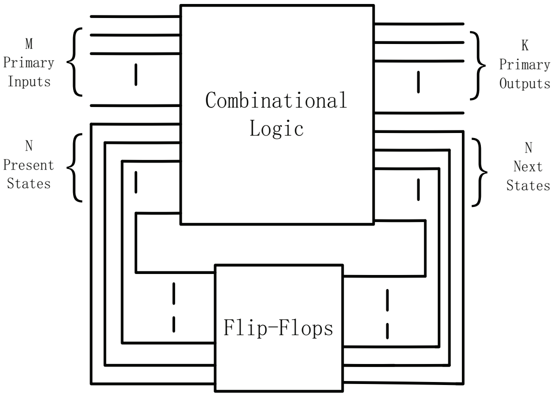


Fig. 6. Mealy model of a sequential circuit.

Consider a general Mealy model of a sequential circuit (Fig. 6). In this circuit, there are *M*+*N* inputs: *M* of them are Primary Inputs and the remaining *N* inputs are Present States (i.e., the feedback signals from the flip-flops). There are also *K*+*N* outputs: *K* of them are Primary Outputs, while the remaining *N* outputs are Next States, which will be stored in the flip-flops and then fed back into the inputs during the next clock cycle.

For this circuit, SPTMs define several matrices mapping from the *M*+*N* inputs to the *N* next states (denoted by $\boldsymbol{\Phi}_{ns}$) and to the *K* primary outputs (denoted by $\boldsymbol{\Phi}_{po}$), also from the *N* next states to the *N* present states. In the SPTM $\boldsymbol{\Phi}_{ns}$, the entries denote the transition probabilities from the $2^{\mathrm{M+N}}$ input combinations to the 2^{N} next states. $\boldsymbol{\Phi}_{ns}$ is therefore a $2^{\mathrm{M+N}} \times 2^{\mathrm{N}}$ matrix, given by

$$\boldsymbol{\Phi}_{ns} = \begin{bmatrix} \mathrm{p}(0|0) & \mathrm{p}(1|0) & \dots & \mathrm{p}(\mathrm{j}|0) & \dots & \mathrm{p}(2^{\mathrm{N}}-1|0) \\ \mathrm{p}(0|1) & \mathrm{p}(1|1) & \dots & \mathrm{p}(\mathrm{j}|1) & \dots & \mathrm{p}(2^{\mathrm{N}}-1|1) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathrm{p}(0|\mathrm{i}) & \mathrm{p}(1|\mathrm{i}) & \dots & \mathrm{p}(\mathrm{j}|\mathrm{i}) & \dots & \mathrm{p}(2^{\mathrm{N}}-1|\mathrm{i}) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mathrm{p}(0|\,2^{\mathrm{M+N}}-1) & \mathrm{p}(1|\,2^{\mathrm{M+N}}-1) & \dots & \mathrm{p}(j|\,2^{\mathrm{M+N}}-1) & \dots & \mathrm{p}(2^{\mathrm{N}}-1|2^{\mathrm{M+N}}-1) \end{bmatrix} \tag{1}$$

where p(j|i) represents the conditional transition probability that the next state has the j^{th} value, given that the inputs have the i^{th} value. Similarly, the entries in $\boldsymbol{\Phi}_{po}$ are given by the transition probabilities from the $2^{\mathrm{M+N}}$ input combinations to the 2^{K} primary output combinations.

Furthermore, the circuit SPTM $\boldsymbol{\Phi}_{c}$ is defined for the mappings between the *M*+*N* inputs and the *K*+*N* outputs. In the likely circuit scenario, the primary outputs and the next states are correlated due to the fanouts of signals to both the primary inputs and the next states (an example is evident in the half adder circuit of Fig. 7). So, $\boldsymbol{\Phi}_{c}$, like any other SPTM, can be obtained in a similar way as the combinational PTMs, i.e. by combining the gate PTMs following the matrix operation rules and the connectivity of the gates in a circuit.

If the flip-flops are also subject to errors, the SPTM $\boldsymbol{\Phi}_{ff}$ must be defined. For a single flip-flop with an error rate of $\boldsymbol{\varepsilon}$, its SPTM $\boldsymbol{\Phi}_{\mathbf{1}-ff}$ is given by

$$\boldsymbol{\Phi}_{\mathbf{1}-ff} = \begin{bmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{bmatrix}. \tag{2}$$

For *N* independent flip-flops, the SPTM $\boldsymbol{\Phi}_{ff}$ is obtained as

$$\boldsymbol{\Phi}_{ff} = \boldsymbol{\Phi}_{\mathbf{1}-ff}{}^{\otimes N}, \tag{3}$$

i.e., $\boldsymbol{\Phi}_{ff}$ is the *N*th tensor product of $\boldsymbol{\Phi}_{\mathbf{1}-ff}$. Its entries are the transition probabilities from the 2^{N} input combinations and the 2^{N} output combinations of the flip-flops.

At time *t*, the primary-input vector is given by

$$\mathbf{P_{pi}}(\mathrm{t}) = (\mathrm{P_{pi}}(0,\mathrm{t}), \mathrm{P_{pi}}(1,\mathrm{t}), \dots \dots, \mathrm{P_{pi}}((2^{\mathrm{M}}-1),\mathrm{t}), \tag{4}$$

where $\mathrm{P_{pi}}(\mathrm{i},\mathrm{t})$ is the probability that the primary inputs have the i^{th} value at time *t*. The present-state vector is given by

$$\mathbf{P_{ps}}(\mathrm{t}) = (\mathrm{P_{ps}}(0,\mathrm{t}), \mathrm{P_{ps}}(1,\mathrm{t}), \dots \dots, \mathrm{P_{ps}}((2^{\mathrm{M}}-1),\mathrm{t}), \tag{5}$$

where $\mathrm{P_{ps}}(\mathrm{i},\mathrm{t})$ is the probability that the present states have the i^{th} value at time *t*.

Therefore inputs can be represented by a vector of length $2^{\mathrm{M+N}}$ given by

$$\mathbf{P_{in}}(\mathrm{t}) = \mathbf{P_{pi}}(\mathrm{t}) \otimes \mathbf{P_{ps}}(\mathrm{t}), \tag{6}$$

where ⊗ indicates the tensor product.

For the *N* next states, their joint distribution is described by a vector $\mathbf{P_{ns}}(\mathrm{t})$ of length 2^{N}, and

$$\mathbf{P_{ns}}(\mathrm{t}) = \mathbf{P_{in}}(\mathrm{t}) * \boldsymbol{\Phi}_{ns}. \tag{7}$$

The present-state vector at time *t+1*, $\mathbf{P_{ps}}(\mathrm{t}+1)$, can be calculated as follows:

$$\mathbf{P_{ps}}(\mathrm{t}+1) = \mathbf{P_{ns}}(\mathrm{t}) * \boldsymbol{\Phi}_{ff}. \tag{8}$$

Similarly, the primary-output vector $\mathbf{P_{po}}(\mathrm{t})$ is given by

$$\mathbf{P_{po}}(\mathrm{t}) = \mathbf{P_{in}}(\mathrm{t}) * \boldsymbol{\Phi}_{po}. \tag{9}$$

3.4 Reliability evaluation using SPTMs

In a sequential circuit, the cumulative effect of errors is modeled by matrix operations (as according to the Markov process); specifically, error propagation is described by matrix multiplications, while the combination of signals is described by tensor products. By considering the SPTMs defined previously for the sequential circuit of Fig. 6, the input vector at time *t+1*, $\mathbf{P_{in}}(\mathrm{t}+1)$, can be computed by

$$\begin{aligned} \mathbf{P_{in}}(\mathrm{t}+1) &= \mathbf{P_{pi}}(\mathrm{t}+1) \otimes \mathbf{P_{ps}}(\mathrm{t}+1) \\ &= \mathbf{P_{pi}}(\mathrm{t}+1) \otimes \big(\mathbf{P_{in}}(\mathrm{t}) * \boldsymbol{\Phi}_{ns} * \boldsymbol{\Phi}_{ff}\big). \end{aligned} \tag{10}$$

It can be shown that

$$\mathbf{P_{in}}(\mathrm{t}+1) = \mathbf{P_{in}}(\mathrm{t}) * \big(\mathbf{P_{pi}}(\mathrm{t}+1) \otimes (\boldsymbol{\Phi}_{\mathbf{ns}} * \boldsymbol{\Phi}_{ff})\big), \tag{11}$$

as per the Theorem in the Appendix.

Let

$$\boldsymbol{\Phi}_{in}(t) = \mathbf{P_{pi}}(\mathrm{t}+1) \otimes (\boldsymbol{\Phi}_{\mathbf{ns}} * \boldsymbol{\Phi}_{ff}), \tag{12}$$

(11) can be written as

$$\mathbf{P_{in}}(\mathrm{t}+1) = \mathbf{P_{in}}(\mathrm{t}) * \boldsymbol{\Phi}_{in}(t). \tag{13}$$

$\boldsymbol{\Phi}_{in}(t)$ is a time-dependent SPTM, whose entries are the transition probabilities from the $2^{\mathrm{M+N}}$ input combinations at time *t* to the 2^{M+N} input combinations at time *t+1*. Therefore, (13) describes the (error) characteristics of the sequential circuit (under its current primary inputs) and captures the temporal correlation between the inputs at different time steps.

At time *t+1*, the primary output vector can now be computed as

$$\begin{aligned}\mathbf{P_{po}}(t+1) &= \mathbf{P_{in}}(t+1) * \boldsymbol{\Phi}_{po} = \mathbf{P_{in}}(t) * \boldsymbol{\Phi}_{\mathbf{in}}(t) * \boldsymbol{\Phi}_{po}\\ &= \mathbf{P_{in}}(t-1) * \boldsymbol{\Phi}_{in}(t-1) * \boldsymbol{\Phi}_{in}(t) * \boldsymbol{\Phi}_{po}\\ &= \cdots\\ &= \mathbf{P_{in}}(t-k) * \boldsymbol{\Phi}_{in}(t-k) * \boldsymbol{\Phi}_{in}(t-k+1) * \ldots *\\ &\qquad \boldsymbol{\Phi}_{in}(t-1) * \boldsymbol{\Phi}_{po}\\ &= \cdots\\ &= \mathbf{P_{in}}(0) * \boldsymbol{\Phi}_{in}(0) * \boldsymbol{\Phi}_{in}(1) * \ldots * \boldsymbol{\Phi}_{in}(t-1) * \boldsymbol{\Phi}_{po}\\ &= \mathbf{P_{in}}(0) * \left(\textstyle\prod_{k=0}^{t} \boldsymbol{\Phi}_{in}(k)\right) * \boldsymbol{\Phi}_{po.}\end{aligned} \tag{14}$$

Furthermore, assume that in the error-free case, the ideal output vector is given by $\mathbf{I_{po}}(t+1)$; so if the primary inputs at every time step and the initial state are known, the output reliability of a sequential circuit at time *t+1* can be computed by:

$$\mathbf{R_{po}}(t+1) = \mathbf{P_{po}}(t+1) \cdot \mathbf{I_{po}}(t+1), \tag{15}$$

i.e., by the dot product of the two vectors.

Example: Consider the sequential half adder circuit shown in Fig. 7. Given the gate PTMs, the circuit PTM can be found by first dividing the combinational part of the circuit into several levels and then evaluating the PTMs at each level. Alternatively, $\boldsymbol{\Phi}_{ns}$ and $\boldsymbol{\Phi}_{po}$ can be obtained for the next state and output logic.

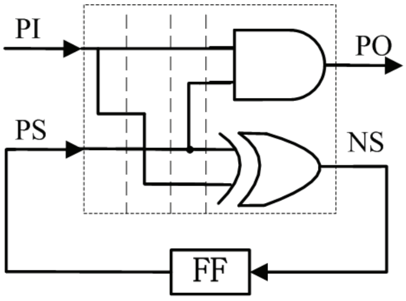


Fig. 7. SPTM evaluation of a sequential half adder with one primary input, one flip-flop and one primary output.

Assume a gate error rate of 0.01, we obtain

$$\boldsymbol{\Phi}_{po} = \begin{bmatrix}0.99 & 0.01\\ 0.99 & 0.01\\ 0.99 & 0.01\\ 0.01 & 0.99\end{bmatrix}, \tag{16}$$

and

$$\boldsymbol{\Phi}_{ns} = \begin{bmatrix}0.99 & 0.01\\ 0.01 & 0.99\\ 0.01 & 0.99\\ 0.99 & 0.01\end{bmatrix}. \tag{17}$$

Also assume that the flip-flop has an error rate of 0.02, i.e., $\boldsymbol{\varepsilon} = 0.02$, then

$$\boldsymbol{\Phi}_{ff} = \boldsymbol{\Phi}_{1-ff} = \begin{bmatrix}0.98 & 0.02\\ 0.02 & 0.98\end{bmatrix}. \tag{18}$$

Given the primary input at each time step, the reliability of the primary output can be calculated as follows. Let the initial state of the flip-flop be $\mathbf{P_{ps}}(0) = (0.99, 0.01)$. If the primary-input vector is given by $\mathbf{P_{pi}}(0) = (0.05, 0.95)$, then at *t=0*, the input vector can be calculated as

$$\begin{aligned}\mathbf{P_{in}}(0) &= \mathbf{P_{pi}}(0) \otimes \mathbf{P_{ps}}(0)\\ &= (0.0495, 0.0005, 0.9405, 0.0095).\end{aligned} \tag{19}$$

At *t=1*, the present-state vector is given by

$$\begin{aligned}\mathbf{P_{ps}}(1) = \mathbf{P_{ns}}(0) * \boldsymbol{\Phi}_{ff} &= \mathbf{P_{in}}(0) * \boldsymbol{\Phi}_{ns} * \boldsymbol{\Phi}_{ff}\\ &= (0.0851, 0.9149).\end{aligned} \tag{20}$$

If the primary input vector at *t=1* is given by $\mathbf{P_{pi}}(1) = (0.98, 0.02)$, then the input vector $\mathbf{P_{in}}(1)$ is given by

$$\begin{aligned}\mathbf{P_{in}}(1) &= \mathbf{P_{pi}}(1) \otimes \mathbf{P_{ps}}(1)\\ &= (0.0834, 0.8966, 0.0017, 0.0183).\end{aligned} \tag{21}$$

The primary output vector is then obtained as

$$\mathbf{P_{po}}(1) = \mathbf{P_{in}}(1) * \boldsymbol{\Phi}_{po} = (0.9721 \quad 0.0279). \tag{22}$$

(22) gives the output distribution after one clock cycle for the probabilistic sequential adder. The above process can also be computed directly from (14) for *t=0*.

Since the ideal error-free output vector is $\mathbf{I_{po}}(1) = (1, 0)$, it can be obtained that

$$\mathbf{R_{po}}(1) = \mathbf{P_{po}}(1) \cdot \mathbf{I_{po}}(1) = 0.9721, \tag{23}$$

i.e., the output reliability at this time is 0.9721.

Table 2 shows the simulation results of some sequential circuits obtained using the SPTM approach. Albeit with an exponential complexity, the SPTM approach produces accurate evaluation results. Alternatively, the reliability can be evaluated using probabilistic gate models [17] or stochastic computational models [18]. As can be seen in Table 2, the reliability values are very different at different clock cycles (denoted by "T") for the probabilistic and approximate designs. For example, for the LOA and AMAs, the reliability is mostly 0, which indicates that these designs are totally unreliable. However, they can be useful in many applications, where a partial loss of precision can be tolerated. Therefore, these new paradigms need to be better assessed as efficient design alternatives. Toward this end, new metrics are proposed subsequently to characterize the extent of loss of precision and its gain in reducing power consumption.

4 Mean and Normalized Error Distances

4.1 Error Distance

In this section, a *new metric* is proposed for evaluating the reliability of an adder and SPTMs are used in the computation of this metric. Consider as an example the case in which the exact output Sum of an adder is "100101" and other values can result as inexact outputs. For example, both "100100" and "110101" represent inexact values. However, these two output values have different implications when compared to the correct value: "100100" means the output is different by 1 (or at a distance of 1) from the correct value, while "110101" is different by 16 (or at a distance of 16) from the correct value. So, an output can take erroneous values that are substantially different from the correct one. This is determined by the error effects on the addition; for example, a lower bit error has less impact on the output of an adder.

TABLE 2. THE RELIABILITY OF SOME SEQUENTIAL CIRCUITS OBTAINED USING SPTMS. THE INPUT VECTOR IS RANDOMLY GENERATED AT EACH CLOCK CYCLE, DENOTED BY "T."

Circuits	Characteristics				Reliability (ε=0.005)				
	Gates	Inputs	Outputs	DFFs	T=1	T=2	T=3	T=4	T=5
2-bit counter	2	0	2	2	0.990	0.980	0.971	0.961	0.952
Simplified semaphore	6	0	2	2	0.983	0.965	0.948	0.932	0.916
s27	10	4	1	3	0.975	0.990	0.985	0.960	0.948
2-bit probabilistic sequential adder	10	4	1	2	0.976	0.962	0.948	0.935	0.922
2-bit Lower-part OR sequential adder	3	4	1	2	1	0	0	0	0
2-bit AMA1 based sequential adder	22 (transistors)	4	1	2	0	1	1	0	0
2-bit AMA2 based sequential adder	22 (transistors)	4	1	2	1	0	0	0	0
2-bit AMA3 based sequential adder	N/A	4	1	2	0	0	0	0	0

Under these circumstances, the metric of circuit reliability has limited usefulness in assessing an adder, because it considers only the presence of an error, but not the error's implication on the performed addition. A new metric referred to as *error distance* (ED) is therefore proposed to better characterize the reliability of an adder.

In general, the ED between two binary numbers, a (erroneous) and b (correct, i.e., golden), is defined as the arithmetic distance between these two numbers, i.e.,

$$ED(\mathrm{a,b}) = |\mathrm{a}-\mathrm{b}| = \left|\sum_i \mathrm{a}[i] * 2^{\mathrm{i}} - \sum_j \mathrm{b}[\mathrm{j}] * 2^{\mathrm{j}}\right|, \quad (24)$$

where i and j are the indices for the bits in a and b, respectively. In the previous example, the two erroneous values "100100" and "110101" have an ED of 1 and 16 to the golden "100101," respectively.

For a non-deterministic implementation, the output is probabilistic and usually follows a distribution for a given input a_i. In this case, the ED of the output (denoted by d_i) is defined as the weighted average of EDs of all possible outputs to the nominal output. Assume that for a given input, the output has a nominal value b, but it can take any value given in a set of vectors b_j $(1 \le j \le r)$; the ED of the output is then given by:

$$d_i = \sum_j \mathrm{ED}(\mathrm{b_j}, \mathrm{b}) * \mathrm{p_j}, \quad (25)$$

where p_j is the output probability of b_j $(1 \le j \le r)$.

4.2 Mean Error Distance (MED)

When the primary inputs to a circuit are non-deterministic and thus each input occurs at certain probability, the *mean error distance* (MED) of a circuit (denoted by d_m) is defined as the mean value of the EDs of all possible outputs for each input. Assume that the input is given by a set of vectors a_i $(1 \le i \le s)$ and that each vector occurs with a probability q_i $(1 \le i \le s)$. Then, the MED of the circuit is given by:

$$d_m = \sum_i \mathrm{d_i} * \mathrm{q_i}, \quad (26)$$

where d_i is the ED of the outputs for input a_i $(1 \le i \le s)$, which can be computed by (25).

For simplicity, uniformly-distributed random inputs are considered hereafter, i.e., each input occurs with the same probability. Consider as an example a 3-bit adder. In Figs. 3 and 4, let k = 3, m = 1 and n = 2. A 3-bit CFA is used to calculate the correct output value. A gate error rate of 0.028 is used for PFA; this value is selected such that the MED is close to that of the LOA (as shown in subsequent sections). Table 3 shows the results of three experiments, each of which consists of four consecutive clock cycles. As expected, the CFA has a MED of 0 throughout the four clock cycles in all experiments. However, the MED for the PFA is significantly greater than that of its LOA counterpart in most cases. Also shown is that the MEDs of the LOA and AMAs can be reduced to 0 at an intermediate time step. For example, the MED of the LOA has a value of 0 at both Clk1 and Clk4 in Experiment 3; this is due to the effect of error masking, as discussed next.

TABLE 3. MEAN ERROR DISTANCE FOR FOUR CLOCK CYCLES WITH RANDOM INPUTS.

Experiment	Architecture	Clk1	Clk2	Clk3	Clk4
No. 1	CFA	0	0	0	0
	PFA	0.7400	1.1920	1.4620	2.1900
	LOA	0	2	1	1
	AMA1	1	1	2	3
	AMA2	1	0	1	2
	AMA3	0	0	1	2
No. 2	CFA	0	0	0	0
	PFA	0.7220	1.0260	1.5940	1.9640
	LOA	0	1	1	1
	AMA1	0	1	1	2
	AMA2	0	1	1	1
	AMA3	1	2	2	3
No. 3	CFA	0	0	0	0
	PFA	0.7820	1.2100	1.5180	2.2340
	LOA	0	2	1	0
	AMA1	0	2	3	3
	AMA2	1	0	2	3
	AMA3	2	1	2	3

For uniformly distributed inputs, the 16 input values occur with the same probability of $1/16$ for this 3-bit adder. Based on the SPTM model, the transition matrix $\boldsymbol{\Phi_{ns}}$ for the lower two bits is given in Fig. 8 for each implementation (as the higher bits are accurate and are therefore the same). Given these transition matrices, the ED of an output can be computed for an input against the corresponding output of the CFA (taken as the golden value). Given $\boldsymbol{\Phi_{LOA}}$ in Fig. 8(b), for example, the MED of the LOA is:

$$d_{m,\mathrm{LOA}} = \frac{1}{16} * (0+0+0+0+0+1+0+1+0+0+2+2+0+1+2+1) = 0.625. \quad (27)$$

The MED of the PFA is calculated using $\boldsymbol{\Phi_{PFA}}$ in Fig. 8(f) as:

$$\mathrm{d_{m,PFA}} = \frac{1}{16} * \sum_i \mathrm{d_i} = 0.6167. \quad (28)$$

$$\Phi_{CFA}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & 1 & & & & & & & \\
0001 & & 1 & & & & & & \\
0010 & & & 1 & & & & & \\
0011 & & & & 1 & & & & \\
0100 & & 1 & & & & & & \\
0101 & & & 1 & & & & & \\
0110 & & & & 1 & & & & \\
0111 & & & & & 1 & & & \\
1000 & & & 1 & & & & & \\
1001 & & & & 1 & & & & \\
1010 & & & & & 1 & & & \\
1011 & & & & & & 1 & & \\
1100 & & & & 1 & & & & \\
1101 & & & & & 1 & & & \\
1110 & & & & & & 1 & & \\
1111 & & & & & & & 1 &
\end{array}$$

(a)

$$\Phi_{LOA}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & 1 & & & & & & & \\
0001 & & 1 & & & & & & \\
0010 & & & 1 & & & & & \\
0011 & & & & 1 & & & & \\
0100 & & 1 & & & & & & \\
0101 & & 1 & & & & & & \\
0110 & & & & 1 & & & & \\
0111 & & & & 1 & & & & \\
1000 & & & 1 & & & & & \\
1001 & & & & 1 & & & & \\
1010 & & & & & & & 1 & \\
1011 & & & & & & & & 1\\
1100 & & & & 1 & & & & \\
1101 & & & & 1 & & & & \\
1110 & & & & & & & & 1\\
1111 & & & & & & & & 1
\end{array}$$

(b)

$$\Phi_{AMA1}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & & & & 1 & & & & \\
0001 & & & 1 & & & & & \\
0010 & & & & & & 1 & & \\
0011 & & & & & 1 & & & \\
0100 & & & & 1 & & & & \\
0101 & & & 1 & & & & & \\
0110 & & & & & & 1 & & \\
0111 & & & & & 1 & & & \\
1000 & & & & 1 & & & & \\
1001 & & & & & 1 & & & \\
1010 & & & & & & 1 & & \\
1011 & & & & & 1 & & & \\
1100 & & & & 1 & & & & \\
1101 & & & & & 1 & & & \\
1110 & & & & & & 1 & & \\
1111 & & & & & 1 & & &
\end{array}$$

(c)

$$\Phi_{AMA2}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & 1 & & & & & & & \\
0001 & 1 & & & & & & & \\
0010 & 1 & & & & & & & \\
0011 & 1 & & & & & & & \\
0100 & & & 1 & & & & & \\
0101 & & & 1 & & & & & \\
0110 & & & 1 & & & & & \\
0111 & & & 1 & & & & & \\
1000 & & & & & 1 & & & \\
1001 & & & & & 1 & & & \\
1010 & & & & & 1 & & & \\
1011 & & & & & 1 & & & \\
1100 & & & & & 1 & & & \\
1101 & & & & & 1 & & & \\
1110 & & & & & & & 1 & \\
1111 & & & & & & & 1 &
\end{array}$$

(d)

$$\Phi_{AMA3}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & 1 & & & & & & & \\
0001 & & 1 & & & & & & \\
0010 & & & 1 & & & & & \\
0011 & & & & 1 & & & & \\
0100 & 1 & & & & & & & \\
0101 & & 1 & & & & & & \\
0110 & & & 1 & & & & & \\
0111 & & & & 1 & & & & \\
1000 & & & & & 1 & & & \\
1001 & & & & & & 1 & & \\
1010 & & & & & & & 1 & \\
1011 & & & & & & & & 1\\
1100 & & & & & 1 & & & \\
1101 & & & & & & 1 & & \\
1110 & & & & & & & 1 & \\
1111 & & & & & & & & 1
\end{array}$$

(e)

$$\Phi_{PFA}=\begin{array}{c|cccccccc}
 & 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111\\
\hline
0000 & 0.722 & 0.044 & 0.126 & 0.004 & 0.084 & 0.005 & 0.014 & 0.001\\
0001 & 0.044 & 0.722 & 0.004 & 0.126 & 0.005 & 0.084 & 0.001 & 0.014\\
0010 & 0.044 & 0.003 & 0.718 & 0.044 & 0.096 & 0.002 & 0.088 & 0.005\\
0011 & 0.003 & 0.044 & 0.044 & 0.718 & 0.002 & 0.096 & 0.005 & 0.088\\
0100 & 0.044 & 0.722 & 0.004 & 0.126 & 0.005 & 0.084 & 0.001 & 0.014\\
0101 & 0.083 & 0.004 & 0.765 & 0.044 & 0.012 & 0.001 & 0.086 & 0.005\\
0110 & 0.003 & 0.044 & 0.044 & 0.718 & 0.002 & 0.096 & 0.005 & 0.088\\
0111 & 0.045 & 0.002 & 0.044 & 0.001 & 0.806 & 0.047 & 0.051 & 0.004\\
1000 & 0.043 & 0.003 & 0.718 & 0.044 & 0.097 & 0.002 & 0.088 & 0.005\\
1001 & 0.003 & 0.043 & 0.044 & 0.718 & 0.002 & 0.097 & 0.005 & 0.088\\
1010 & 0.043 & 0.002 & 0.002 & 0 & 0.763 & 0.047 & 0.138 & 0.005\\
1011 & 0.002 & 0.043 & 0 & 0.002 & 0.047 & 0.763 & 0.005 & 0.138\\
1100 & 0.003 & 0.043 & 0.044 & 0.718 & 0.002 & 0.097 & 0.005 & 0.088\\
1101 & 0.047 & 0.003 & 0.044 & 0.001 & 0.804 & 0.046 & 0.051 & 0.004\\
1110 & 0.002 & 0.043 & 0 & 0.002 & 0.047 & 0.763 & 0.005 & 0.138\\
1111 & 0.004 & 0.001 & 0.025 & 0.002 & 0.091 & 0.004 & 0.826 & 0.047
\end{array}$$

(f)

Fig. 8. The SPTM $\mathbf{\Phi_{ns}}$ for the two lower bits of the 3-bit adder: (a) CFA (b) LOA (c) AMA1 (d) AMA2 (e) AMA3 and (f) PFA. Each row corresponds to an input consisting of two bits from the primary inputs and two bits from the feedbacks; each column corresponds to a 3-bit output of the 2-bit addition. An entry in a matrix indicates a transition (probability) from an input to an output.

Similarly, the MEDs of the AMAs are obtained as follows:

$$d_{m,\text{AMA1}} = \frac{1}{16} * (3+1+3+1+2+0+2+0+1+1+1+1+0+0+0+2) = 1.125, \quad (29)$$

$$d_{m,\text{AMA2}} = \frac{1}{16} * (0+1+2+3+1+0+1+2+2+1+0+1+1+0+1+0) = 1, \quad (30)$$

$$d_{m,\text{AMA3}} = \frac{1}{16} * (0+0+0+0+1+1+1+1+2+2+2+2+1+1+1+1) = 1. \quad (31)$$

This shows that the AMA1 has the largest MED, while the MED of the PFA (0.6167) is only slightly different from that of the LOA for an error rate of 0.028. However, the cumulated error distances are quite different in the sequential adders, as shown in Table 2. This is mainly due to a result of partial error masking [19]. For the LOA and AMA2, several so-called restoring inputs are found, as indicated by the neighboring 1's in the columns of $\boldsymbol{\Phi_{LOA}}$ and $\boldsymbol{\Phi_{AMA2}}$. By these inputs, errors are logically masked, which leads to the same next state from different present states. This can also be explained as follows. For certain primary inputs, the approximate logic masks the cumulative errors in the LOA and AMA2. However, this is not the case for the PFA (or CFA), because it is always affected by errors; therefore, errors accumulate rather than being masked.

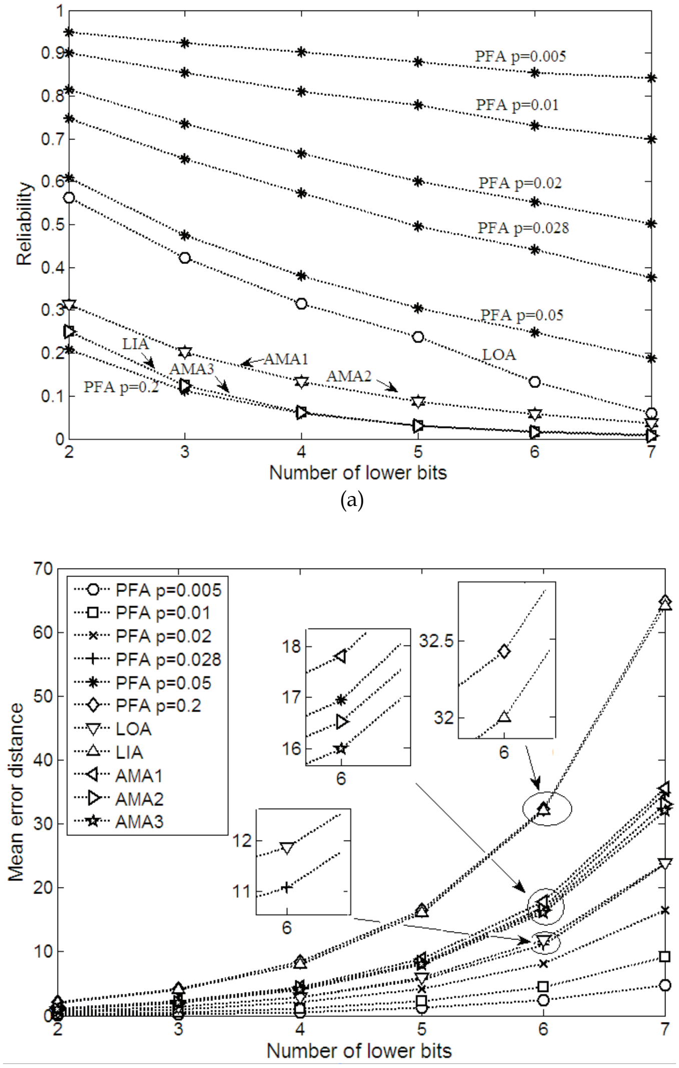
4.3 Mean Error Distance (MED) Evaluation

In this section, the MEDs of the various designs are evaluated against a baseline design of a reduced precision adder, i.e., the lower-bit ignored adder (LIA). In this implementation, rather than using n-bit unreliable adders for computation, the lower significant bits are ignored.

Consider the general case of a 32-bit adder (k=32). Fig. 9(a) shows the reliability of each adder design, while the MEDs are plotted in Fig. 9(b) by varying the number of lower bits, n. A comparison of the adder implementations shows that the interpretation of these metrics (i.e., reliability and MED) leads to a different assessment result; for example under the MED metric, the performance of LOA is similar to a PFA with a gate error rate of 0.028. However, under the reliability metric the reliability of LOA is significantly lower than the reliability of PFA for the same gate error rate.

As shown in Fig. 9(b), an n-bit ignored LIA performs slightly better in terms of the MED than an n-bit PFA at a gate error rate of 0.2 for the 32-bit adder. However, a PFA with an error rate significantly lower than 0.2 is expected to have a smaller MED. The AMAs have a similar performance as a PFA with an error rate of approximately 0.05. Moreover, as shown in Fig. 9(b), the expected MEDs of both the LOA and the PFA increase exponentially with the number of lower bits. For the LOA, error masking occurs when the carry bit is ignored - there is no carry for the lower bits. Therefore, errors in a lower bit are operationally isolated. In this way, error masking occurs in the combinational circuit. Therefore, errors in the lower bits cannot be propagated to the higher bits, thus preventing their cumulative effect.

The simulation results of Fig. 9 show the effectiveness of the proposed evaluation technique. It reveals that, although there is no cumulative error for the LOA, an error due to the approximate logic makes its MED not as optimistic as expected, especially for an increased number of lower bits. This is due to the fact that an error can occur in the most significant lower bit. The PFA shows a different scenario. Errors in the lower bits are likely to be propagated to the higher bits. So, its MED is rather large even for a small error rate, due to the cumulative errors from the lower bits (even though the error that results from the most significant bit is not that large).



(a)

(b)

Fig. 9. (a) Reliability vs. the number of lower bits in a 32-bit adder, and (b) MED vs. the number of lower bits in the 32-bit adder.

4.4 Normalized Error Distance (NED) and its Evaluation

It is shown that in Fig. 9(b), the MED increases exponentially with the number of lower bits in an adder. Therefore, MED is an unfair metric when a comparison is made between two adders with different lower bits, as the maximum value of error that can be effectively reached, has also changed. To overcome this limitation, a normalized MED, referred to as a *normalized error distance* (NED), is defined as follows:

$$d_n = \frac{d_m}{D}, \tag{32}$$

where d_m is the MED and D is the maximum value of error that an unreliable adder can have. This maximum value is usually 2^n for n lower bits, so we obtain

$$d_n = \frac{d_m}{2^n}. \tag{33}$$

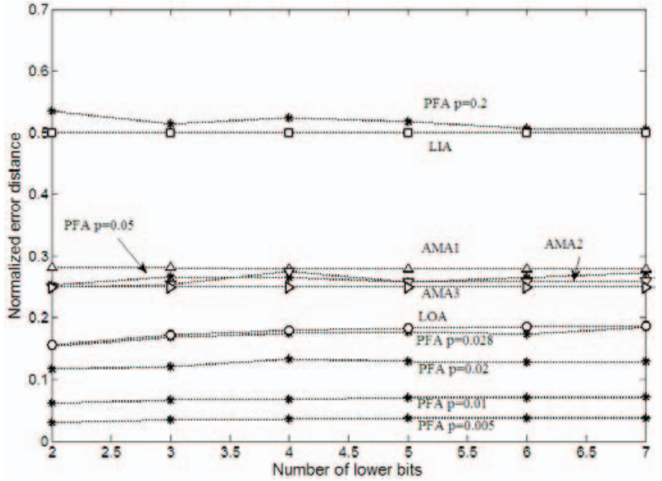


Fig. 10. Normalized Error Distance (NED) vs. the number of lower bits.

Fig. 10 shows the NEDs of the various adder implementations. The NEDs of the LIA are constant at 0.5 and thus serving as a baseline. For other implementations, as revealed in the figure, the NED shows little change and only fluctuates in a small interval when different lower bits are compared. This is consistent with the results in Fig. 9(b), i.e., the MED increases exponentially with the number of lower bits. Therefore, NED is a stable metric that is almost independent of the size of an implementation and is useful in assessing the reliability of a specific design. This feature also brings a new perspective for the evaluation and comparison of different adder designs for inexact computing.

However, it should be noted that the MED is very useful in evaluating adder implementations of different size. For example, a different number of lower bits can be implemented in approximate or probabilistic adders, as discussed previously. So, the MED is best suited for evaluating the effectiveness of an implementation of multiple-bit adders.

5 Power and Precision Trade-off

As discussed previously, inexact computing is confronted with many similarities and often contradictory features that can also be found in bio-inspired systems, i.e., systems made of a large number of unreliable modules. These systems utilize extensive networks to circumvent the unreliable nature of the computational modules while still retaining low power/energy consumption. The adder configurations presented previously draw significant resemblance to this type of system. The LOA and AMAs resort to approximate logic to target reliable modules and PFA resorts to characterizing probabilistic behavior of nanoscale modules. However, one of the issues that must be addressed for inexact computing is that probabilistic implementations may likely tradeoff too much accuracy for little saving in area and power. In this section, this tradeoff is evaluated using the product of power and NED of a design.

Consider first the power consumption of the adders. As suggested in [3], the energy (E) consumed by a probabilistic inverter increases exponentially with the probability of its correct functioning, *p*, if the noise magnitude remains constant. For simplicity, here we assume that the energy consumption of any binary gate increases exponentially with p. It is further assumed that the power consumption of any adder is normalized to that of a 1-bit CFA (given by a unit power), i.e.,

$$E_{1-\mathrm{CFA}} = 1. \tag{34}$$

Hence, the power consumption of a 1-bit PFA is then:

$$\mathrm{E}_{1-\mathrm{PFA}} = \mathrm{E}_{1-\mathrm{CFA}} * \frac{\mathrm{e}^{\mathrm{p}}}{\mathrm{e}^{1}} = \mathrm{e}^{\mathrm{p}-1}. \tag{35}$$

For a *k*-bit CFA, its power is then:

$$\mathrm{E}_{\mathrm{CFA}} = \mathrm{k} * \mathrm{E}_{1-\mathrm{CFA}} = \mathrm{k}. \tag{36}$$

For a *k*-bit PFA with *m* higher bits and *n* lower bits, its power consumption is given by:

$$\mathrm{E}_{\mathrm{PFA}} = \mathrm{m} * \mathrm{E}_{1-\mathrm{CFA}} + \mathrm{n} * \mathrm{E}_{1-\mathrm{PFA}} = \mathrm{m} + \mathrm{n} * \mathrm{e}^{\mathrm{p}-1}. \tag{37}$$

Since adders perform a similar function (i.e., an addition), we assume that the switching activities of the gates in an adder are similar. Therefore, the power consumption is considered to be proportional to the number of logic gates for an approximate implementation. In a single-bit LOA, there is only one OR gate instead of five gates as in a conventional adder; so as a first-order estimation, the power consumption of the lower bits in a LOA is 1/5 of that in a CFA, i.e.,

$$\mathrm{E}_{\mathrm{LOA}} = \mathrm{m} * \mathrm{E}_{1-\mathrm{CFA}} + \mathrm{n} * \mathrm{E}_{\mathrm{OR}} = \mathrm{m} + 0.2 * \mathrm{n}. \tag{38}$$

For the three AMAs, a reduction in the number of transistors allows for a lower operating voltage, which subsequently reduces power consumption. An application of the Inverse Discrete Cosine Transform (IDCT) is considered for evaluating the power consumption in [8]. Compared to an operating voltage of 1.13 V for the accurate IDCT operation (equivalent to an operation using the CFA), an AMA-based IDCT operation requires a voltage of 1.04 V, 1.1 V and 1.01 V, respectively for the three approximate implementations [8]. Therefore, the power can be estimated as

$$\mathrm{E}_{\mathrm{AMA}-1} = \mathrm{m} * \mathrm{E}_{1-\mathrm{CFA}} + \mathrm{n} * \mathrm{E}_{1-\mathrm{AMA}-1} = \mathrm{m} + \mathrm{n} * \frac{1.04^2}{1.13^2}. \tag{39}$$

$$\mathrm{E}_{\mathrm{AMA}-2} = \mathrm{m} * \mathrm{E}_{1-\mathrm{CFA}} + \mathrm{n} * \mathrm{E}_{1-\mathrm{AMA}-2} = \mathrm{m} + \mathrm{n} * \frac{1.1^2}{1.13^2}. \tag{40}$$

$$\mathrm{E}_{\mathrm{AMA}-3} = \mathrm{m} * \mathrm{E}_{1-\mathrm{CFA}} + \mathrm{n} * \mathrm{E}_{1-\mathrm{AMA}-3} = \mathrm{m} + \mathrm{n} * \frac{1.01^2}{1.13^2}. \tag{41}$$

For the LIA, its power consumption is simply:

$$\mathrm{E}_{\mathrm{LIA}} = \mathrm{m}. \tag{42}$$

The power consumption and saving of each lower bit compared to the CFA are reported in Table 4 for the various implementations. Given a fixed power budget, it has been shown that the MED of the PFA drastically increases with the number of lower unreliable bits, while the LIA has a significantly lower MED and the LOA performs the best with the smallest MED [11]. For a fixed MED, similarly, the power consumption of a 32-bit sequential adder can be considered.

TABLE 4. POWER AND SAVING PER LOWER BIT OF THE ADDER IMPLEMENTATIONS (COMPARED TO AND NORMALIZED TO THE POWER CONSUMPTION OF A 1-BIT CFA).

Implementation	Power per bit	Power saving per bit
PFA	p=0.05 : 0.9512	0.0488
	p=0.028: 0.9724	0.0276
	p=0.02: 0.9802	0.0198
	p=0.01: 0.9900	0.0100
	p=0.005: 0.9950	0.0050
LOA	0.2	0.8
LIA	0	1
AMA1	0.8471	0.1529
AMA2	0.9476	0.0524
AMA3	0.7989	0.2011

Table 5 shows the power consumption of the various implementations at MED=16. There is no substantial difference in the power consumption; however, the LOA has the best performance with the LIA as a close second. The PFAs consume more power than the LIA. For more unreliable bits, a lower error rate is required, so a higher power consumption results. So far, the comparison is constrained by either a fixed power budget or MED. Next, a different metric is used to evaluate the tradeoff between power consumption and precision (as represented by NED).

TABLE 5. POWER CONSUMPTION OF VARIOUS IMPLEMENTATIONS OF A 32-BIT ADDER WITH A LARGEST MED OF 16 (IN UNITS OF POWER CONSUMPTION OF A 1-BIT CFA).

Implementation	Features	Power consumption
PFA	m=27, n=5 : p=0.05	31.7561
	m=26, n= 6 : p=0.031	31.8169
	m=25, n= 7 : p=0.014	31.9027
LOA	m=26, n=6	27
LIA	m=28, n=4	28
AMA1	m=26, n=6	31.0823
AMA2	m=26, n=6	31.6856
AMA3	m=26, n=6	30.7933

This metric is given by the product of the normalized power per lower bit and NED, i.e., the power-NED product. As shown in Fig. 11, the power-NED product of a design has a rather constant value, which is nearly independent of the number of lower bits. For an implementation such as the PFA with a gate error rate of 0.005, its power consumption is considered high, while its NED is low. For a larger gate error rate, the power consumption decreases, while the NED increases. However, this synergetic effect of power and NED is captured by the power-NED product; the smaller the product, the better the design, in terms of a tradeoff between power consumption and precision. As shown in Fig. 11, the AMA2 has a similar power precision tradeoff as the PFA with a gate error rate of 0.05, while the AMA3 has a better power efficiency. Also shown is that the PFA with a gate error rate of 0.005 has a comparable power-NED product to that of the LOA, so a PFA with a lower gate error rate is preferred.

This tradeoff can further be explained by the ratio of the normalized power saving and NED, i.e., the power saving per lower bit (compared to the CFA) divided by the NED. This power saving-NED ratio provides a quantitative measure to assess the implications of the efficiency as related to the power saving per unit of the resulting error distance. Since it is normalized to one lower bit, as shown in Fig. 12, the power saving-NED ratio is also nearly independent of the number of lower bits. Moreover from Fig. 12, the LOA has the highest ratio; so when trading off precision for power saving, the LOA has a better efficiency than other designs considered in this manuscript. Note that in the analysis of the power-NED product and power saving-NED ratio, the CFA and LOA are not included, because the CFA has an NED of 0 and the LOA has a power of 0 (per lower bit). This shows a limitation of these measures, i.e., their inability to capture the effect of an individual metric of power or precision.

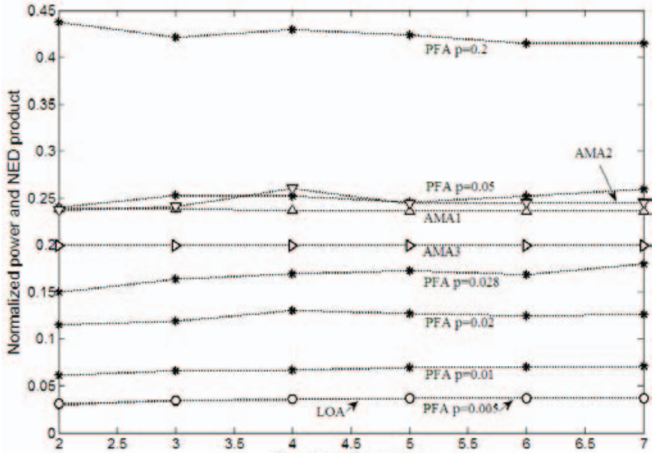


Fig. 11. The power-NED product vs. the number of lower bits for different adder implementations and gate error rates.

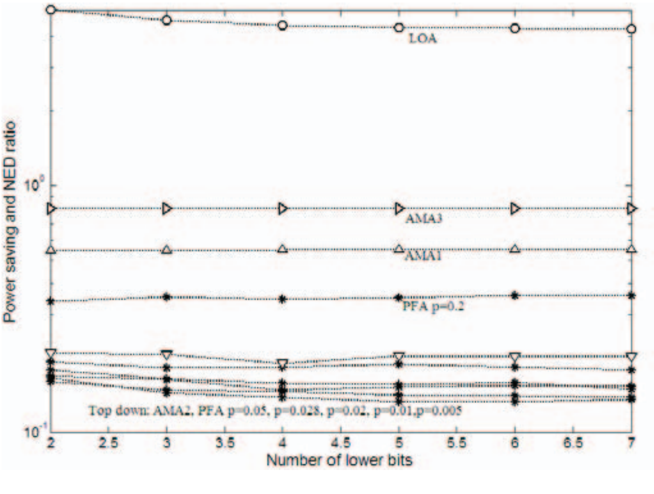


Fig. 12. The power saving-NED ratio vs. the number of lower bits for different adder implementations and gate error rates.

The relationship between power and precision, however, is further revealed in Fig. 13, in which the analysis of CFA and LIA is included. Power and precision are represented by the power consumption per bit and the NED of a design. While the LIA and CFA are the two extreme corner designs (with a power of 0 and an NED of 0, respectively), the other designs generally save power by allowing for some loss of precision. The LOA shows a better tradeoff between power saving and precision loss. Clearly, a design with a better power saving efficiency in terms of precision loss is desired (as indicated by the direction of the arrow in Fig. 13). As the product of normalized power and NED is used to investigate the precision/power relationship, this imposes an equal weight on the impact of power consumption and precision. In practice, a different measure that emphasizes the importance of a particular metric (such as the power or precision) can be used for a better assessment of a design according to the specific requirement of an application.

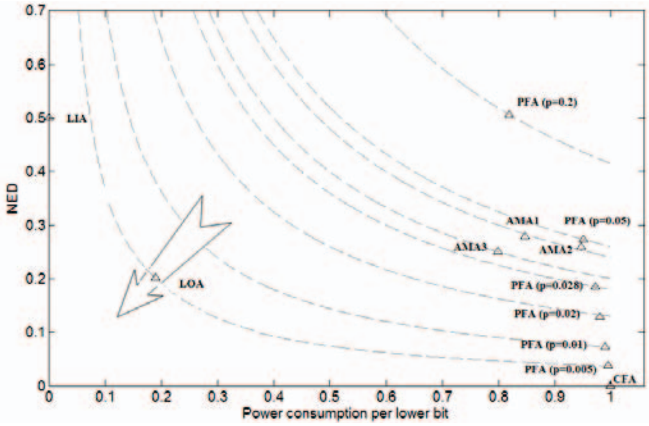


Fig. 13. Relationship between power and precision, given by the power consumption per bit and the NED of a design. Each dashed curve indicates a value of the product of power per bit and the NED. The arrow points to the direction for a better design with a more efficient power and precision tradeoff.

6 Conclusion

This paper has proposed several new metrics for evaluating approximate and probabilistic adders with respect to their reliability and power efficiency, as unified figures of merit for design assessment in inexact computing applications. The Error distance (ED) is defined as the arithmetic distance between an erroneous output and the correct one. The Mean error distance (MED) and normalized error distance (NED) have been proposed by considering the averaging effect of multiple inputs and the normalization of multiple-bit adders. While the MED is effective in evaluating the adder implementation of multiple bits, the NED is nearly invariant with the size of an implementation and is therefore useful in the reliability assessment of a specific design. To evaluate the tradeoff between power consumption and precision, the product of power and NED has been considered and the power efficiency against the precision loss is computed for gaining insights into the effectiveness of a design. The so-called sequential probability transition matrices (SPTMs) have been formulated for use in the computation of the proposed metrics.

Using the proposed metrics, several adder implementations, namely the LOA, AMAs and PFA, are compared against a baseline implementation, the LIA. Simulation results have indicated that, compared to probabilistic adders such as the PFA, approximate adders such as the LOA and AMAs are advantageous in terms of power saving, but with a relatively low precision (comparable to that of the PFA with a high gate error rate). Probabilistic adders such as the PFA, on the other hand, are able to provide a high precision, especially for a low gate error rate, but at the cost of a relatively high power consumption. This tradeoff in precision and power are quantitatively evaluated using the proposed metric of power-NED product. The evaluation results are further supported by the analysis of the power saving and NED ratio that indicates the efficiency of a design in trading off precision for power. Although not discussed and beyond the scope of this manuscript, the proposed metrics may also be useful in assessing other arithmetic circuits [20] for inexact computing and/or fault-tolerant designs [21] in nanocomputing applications.

References

[1] Y. Dote, and S.J. Ovaska, "Industrial Applications of Soft Computing: a Review," *Proc. IEEE*, Vol. 89, no. 9, pp. 1243-1265, 2001.

[2] R. Hegde and N.R. Shanbhag, "Soft digital signal processing," *IEEE Trans. VLSI Syst.*, vol. 9, no. 6, pp. 813–823, 2001.

[3] K.V. Palem, "Energy Aware Computing through Probabilistic Switching: A Study of Limits," *IEEE Trans. Computers*, 2005, pp. 1123-1137.

[4] V. Beiu, S. Aunet, J. Nyathi, R.R. Rydberg III and W. Ibrahim, "Serial Addition: Locally Connected Architectures," *IEEE Trans. Circ. and Sys. I*, vol. 54, no. 11, Nov. 2007, pp. 2564–2579.

[5] S. Cotofana, C. Lageweg, S. Vassiliadis, "Addition related arithmetic operations via controlled transport of charge," *IEEE Trans. Computers*, , vol. 54, no. 3, pp. 243- 256, March 2005.

[6] J. Huang and J. Lach, "Exploring the Fidelity-Efficiency Design Space using Imprecise Arithmetic," *ASPDAC*, 2011.

[7] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 4, pp. 850-862, April 2010.

[8] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, K. Roy," IMPACT: IMPrecise adders for low-power approximate computing" *Low Power Electronics and Design (ISLPED)* 2011 International Symposium on. 1-3 Aug. 2011.

[9] S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, "A probabilistic CMOS switch and its realization by exploiting noise," in *Proc. of In IFIP-VLSI SoC*, Perth, Western Australia, Oct. 2005.

[10] M. S. K. Lau, K. V. Ling, Y. C. Chu, A. Bhanu, "A General mathematical Model of Probabilistic Ripple-Carry Adders", in *Proc. Des. Autom. Test Eur.*, 2010, pp. 1100–1105.

[11] J. Liang, J. Han, F. Lombardi, "On the reliable performance of sequential adders in soft computing", in *2011 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, DFT* 2011, pp. 3-10.

[12] A. K. Verma, P. Brisk, P. Ienne, "Variable latency speculative addition: A new paradigm for arithmetic circuit design," *DATE*, pp. 1250-1255, 2008.

[13] N. Zhu, W.L. Goh, K.S. Yeo, "An enhanced low-power high-speed adder for error tolerant application," *ISIC*, pp. 69-72, 2009.

[14] J. Kim, S. Tiwari "Inexact computing for ultra low-power nanometer digital circuit design", in *IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, 2011, pp. 24-31.

[15] S. Krishnaswamy, G. F. Viamontes, I.L. Markov and J. P. Hayes: "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits." *ACM Trans. on Design Automation of Electronic Systems*, vol. 13, article 8, Jan. 2008.

[16] C-C Yu and J. P. Hayes, "Scalable and accurate estimation of probabilistic behavior in sequential circuits" *IEEE VLSI Test Symposium (VTS)*, 2010 28th. Santa Cruz, CA. pp. 165 – 170.

[17] J. Han, H. Chen, E. Boykin and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectronics Reliability*, vol. 51, no. 2, 2011, pp. 468-476.

[18] H. Chen and J. Han, "Stochastic computational models for accurate reliability evaluation of logic circuits," in *GLSVLSI'10, Proc. of the 20th Great Lakes Symposium on VLSI*, Providence, Rhode Island, USA, pp. 61–66, 2010.

[19] J. Liang, J. Han and F. Lombardi, "Reliable Operation and Error Masking in Sequential Circuits," *Internal Report, University of Alberta*, 2011.

[20] P. Kulkarni, P. Gupta and M. Ercegovac. "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *Proc., IEEE/ACM International Conference on VLSI Design*, Mar 31, 2011.

[21] J. Han, J. Gao, Y. Qi, P. Jonker, J. Fortes, "Toward hardware-redundant, fault-tolerant logic for nanoelectronics," *IEEE Design and Test of Computers*, vol. 22, no. 4, pp. 328–339, July–August 2005.

Jinghang Liang was born in Jilin, China, in 1986. He received the B.Eng. degree in Institute of Microelectronics from Tsinghua University, China, in 2009. Since 2010, he has been pursuing the M.Sc degree in Electrical and Computer Engineering, University of Alberta, Edmonton, Canada.

His research interests include nanoelectronic circuit and system design, fault-tolerant system design and genetic network modeling. Besides, he is interested in RF circuit designs especially high-speed PLLs.

Jie Han (S'02–M'05) received the B.Sc. degree in electronic engineering from Tsinghua University, Beijing, China, in 1999 and the Ph.D. degree from Delft University of Technology, The Netherlands, in 2004.

He is currently an assistant professor in the Department of Electrical and Computer Engineering at the University of Alberta, Edmonton, AB, Canada. He was a NASA INAC (Institute for Nanoelectronics and Computing) Postdoctoral Fellow in the Department of Electrical and Computer Engineering at the University of Florida from 2004 to 2007. From 2007 to 2009, he worked as a Research Scientist at the Advanced Medical Diagnostics S.A./B.V. in Belgium. His research interests include reliability and fault tolerance, nanoelectronic circuits and systems, novel computational models for nanoscale and biological applications.

Dr. Han was nominated for the 2006 Christiaan Huygens Prize of Science by the Royal Dutch Academy of Science (Koninklijke Nederlandse Akademie van Wetenschappen (KNAW) Christiaan Huygens Wetenschapsprijs). His work was recognized by the 125th anniversary issue of *Science*, for developing theory of fault-tolerant nanocircuits. He has published a book, two book chapters and about 25 refereed articles in journals and conferences. He served as a Technical Program Chair in IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2012) and as a Technical Committee Member in DFT 2011 and the International Workshop on Unique Chips and Systems, 2010 (UCAS-6) and 2012 (UCAS-7).

Fabrizio Lombardi (M'81–SM'02-F'09) graduated in 1977 from the University of Essex (UK) with a B.Sc. (Hons.) in Electronic Engineering. In 1977 he joined the Microwave Research Unit at University College London, where he received the Master in Microwaves and Modern Optics (1978), the Diploma in Microwave Engineering (1978) and the Ph.D. from the University of London (1982).

He is currently the holder of the International Test Conference (ITC) Endowed Chair Professorship at Northeastern University, Boston. At the same Institution during the period 1998-2004 he served as Chair of the Department of Electrical and Computer Engineering. Prior to Northeastern University, he was a faculty member at Texas Tech University, the University of Colorado-Boulder and Texas A&M University.

During 2007-2010 Dr. Lombardi was the Editor-In-Chief of the IEEE Transactions on Computers. He is also an Associate Editor of the IEEE Transactions on Nanotechnology and IEEE Transactions on CAD of ICAS. Dr. Lombardi serves as the Chair of the Committee on "Nanotechnology Devices and Systems" of the Test Technology Technical Council of the IEEE (2003 -). In the past, Dr. Lombardi was an Associate Editor (1996-2000) and Associate Editor-in-Chief (2000-2006) of the IEEE Transactions on Computers and twice a Distinguished Visitor of the IEEE-CS (1990-1993 and 2001-2004). Dr. Lombardi is a Golden Core Member of the IEEE Computer Society and Fellow of the IEEE for "contributions to testing and fault tolerance of digital systems". He is the recipient of the 2011 Meritorious Service Award by the IEEE Computer Society. Starting January 1, 2012 he is serving as an elected Member of the Board of Governors of the IEEE Computer Society.

Dr. Lombardi has received many professional awards: the Visiting Fellowship at the British Columbia Advanced System Institute, University of Victoria, Canada (1988), twice the Texas Experimental Engineering Station Research Fellowship (1991-1992, 1997-1998) the Halliburton Professorship(1995), the Outstanding Engineering Research Award at Northeastern University (2004) and an International Research Award from the Ministry of Science and Education of Japan (1993-1999). Dr. Lombardi was the recipient of the 1985/86 Research Initiation Award from the IEEE/Engineering Foundation and a Silver Quill Award from Motorola-Austin (1996). At the IEEE DFT07 Symposium, one of his papers won the best paper award.

Dr. Lombardi has been involved in organizing many international symposia, conferences and workshops sponsored by professional organizations as well as guest editor of Special Issues in archival journals and magazines such as IEEE Transactions on Computers, IEEE Transactions on Instrumentation and Measurement, IEEE Transactions on VLSI, the IEEE Micro Magazine and the IEEE Design & Test Magazine. He is the Founding General Chair of the IEEE Symposium on Network Computing and Applications and the IEEE International Workshop on Design and Test of Nano Devices, Circuits and Systems.

His research interests are bio-inspired and nano manufacturing/computing, VLSI design, testing, and fault/defect tolerance of digital systems. He has extensively published in these areas and coauthored/edited seven books.