[Array 15 (2022) 100219](https://doi.org/10.1016/j.array.2022.100219)



Contents lists available at [ScienceDirect](http://www.elsevier.com/locate/array)

Array

journal homepage: [www.elsevier.com/locate/array](http://www.elsevier.com/locate/array)

Remodelling correlation: A fault resilient technique of correlation sensitive stochastic designs

Shyamali Mitra [∗](#_bookmark0), Sayantan Banerjee, Mrinal Kanti Naskar

*Jadavpur University, Kolkata, India*

A R T I C L E I N F O A B S T R A C T

*Keywords:*

Stochastic logic circuits Soft errors

Reliability

Remodelling correlation Priority-based approach

Major sources of error in stochastic circuits are correlation errors, soft errors, and random fluctuation errors, which impact the circuit’s accuracy and reliability. The soft error has the effect of modifying the correlation status, which in turn modifies the probability of the output. This has serious implications for security and medical systems that require highly precise systems. To address this issue, we employ a fault-tolerant technique of correlation-sensitive stochastic logic circuits. To ensure reliable operation, we have developed a *Remodelling Correlation* (*ReCo*) framework for correlation-sensitive stochastic logic elements (SLEs). Using two intriguing case studies, we present two variants of *ReCo* model for combinational circuits with contradictory requirements. To achieve faster convergence to the desired Mean-Squared Error (MSE) value with less hardware area, the proposed method prioritizes the selection of logic elements and the placement of correction blocks. It is shown that the overall reliability of the circuit is unaffected by this method. To demonstrate the usefulness of the proposed framework, the contrast stretch operation on images of CEED2016, a standard contrast enhancement

output images using the proposed method is observed as 0*.*91 ± 0*.*02, which is significantly higher than the dataset, is investigated in a noisy setting. The average Multiscale structural similarity index (MS-SSIM) of the original images with error, i.e., 0*.*75 ± 0*.*12.

# Introduction

Computation on binary numbers using *Stochastic computing* [[1](#_bookmark54)] is gaining popularity nowadays because it offers several advantages [[2](#_bookmark55)] compared to conventional weighted-binary computation. It is a low power and low cost alternative to complex arithmetic functions. With a remarkable reduction in size, circuit complexity and power consump- tion, stochastic architecture has proved to be noise immune compared to the conventional implementation of binarization algorithms [[3](#_bookmark56)] and various other image processing tasks also [[4](#_bookmark57)]. Different types of errors, such as soft errors, correlation induced errors, and random fluctuation errors are identified that affect the accuracy and reliability of stochastic circuits [[5](#_bookmark58)]. Thus, to generate the desired function using unreliable components in the presence of errors has become a challenging task. Transient or soft errors are caused due to exposure to external radiation and are greatly increased by manufacturing defects in a chip. These introduce false logic at the output of the circuit [[6](#_bookmark59)]. If multiple faults occur at the nodes of a gate, the output may be erroneous. Soft errors are the cause of bit-flips in a stochastic bitstream and may lead to an undesirable correlation between two numbers.

There are several analytical approaches to assess the reliability of probabilistic circuits with unpredictable behaviours; Probabilistic

Gate Models (PGM), Probabilistic Transfer Matrices (*𝑃 𝑇 𝑀* ), Stochastic

Computational Model (SCM), Monte Carlo Simulation [[7](#_bookmark60)] etc. Never- theless to mention that majority of works are aligned towards reliability assessment and analysis [[8](#_bookmark61),[9](#_bookmark62)]. In [[10](#_bookmark63)], a multiple fault modelling methodology was proposed for efficient estimation of the circuit’s reli- ability. Gates that have the highest impact on circuit’s reliability were determined to design trade-off at early stages of circuit design. There are several approaches proposed in literature for modelling and analysis of transient faults in logic circuits [[11](#_bookmark64)], modelling and reduction [[12](#_bookmark65)], soft error analysis tool [[13](#_bookmark66)] for combinational logic circuits. In [[14](#_bookmark67)] a Bernoulli distribution based-model for reliability calculation was developed to decompose the evaluation objective of circuits, with single and double fault simulations. The proposed method is scalable with circuit size and is independent of the soft error rate. But the above works deal with the modelling and analysis of soft errors in weighted- binary logic circuits. Recently, Ting et al. [[15](#_bookmark68)] investigated the role of constant inputs in SC, and propose an algorithm to eliminate them by introducing sequential circuits. But they do not deal with handling the transient error scenarios in stochastic circuits. To our knowledge, no correction mechanism exists in the literature to generate reliable output under transient error scenarios for stochastic logic circuits.

∗ Corresponding author.

*E-mail address:* [shyamalimitra.iee@jadavpuruniversity.in](mailto:shyamalimitra.iee@jadavpuruniversity.in) (S. Mitra).

<https://doi.org/10.1016/j.array.2022.100219>

Received 15 April 2022; Received in revised form 21 June 2022; Accepted 2 July 2022

Available online 7 July 2022

2590-0056/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/) [nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

The proposed work focuses on the erroneous behaviour of stochastic circuits, which is primarily attributable to transient errors under noisy conditions, and its methodical correction using a correlation-based

*𝑙* outputs, a circuit *𝑃 𝑇 𝑀* is of size 2*𝑘* × 2*𝑙* . *𝐼 𝑇 𝑀* and *𝑃 𝑇 𝑀* for a two input AND gate are represented as matrices *𝐽* and *𝑀* .

*𝑦* = 0 *𝑦* = 1

framework. Correlation is one of the intriguing topics in SC and has become a powerful tool to analyze the difference in behaviour of the same logic circuit at different correlation status. This investigation is fo- cused on determining the behaviour of circuits subjected to soft errors. To reduce the impact of transient faults and ensure the circuit’s reli-

*𝐽* = ⎡⎢

⎢⎣

1 0

1 0 ⎥

⎤

0 1 ⎥⎦

1 0

ability, bitstreams are injected with desired correlations. Experiments are conducted on SLEs whose functional behaviour vary with changes in correlation. The objective is to develop a technology-independent framework for observing error-free output in complex circuits with minimal hardware. A study is also conducted to demonstrate that the reliability of a circuit is unaffected by subtle changes in correlation status. The contributions are highlighted as follows:

* We develop a correlation-based framework for correlation sensi- tive SLEs under transient error scenarios to model the error-free output.
* A priority-based approach in the selection of SLEs is explored to reduce the hardware complexity and improve the accuracy in computation for complex circuits.
* Two practical multi-input multi-level circuits with two distinct conditions are considered and treated utilizing two different al- gorithms.
* Evaluation of the proposed work on contrast stretch operations on images under high transient error rates.

In essence, this work not only contradicts the popular perception with regard to correlation, but firmly establishes that injecting a controlled degree of correlation can improve the error-resilient behaviour of the

introduced *𝑃 𝑇 𝑀* as a tool to assess reliability for correlation sensitive circuit. The rest of the work is organized as follows: In Section [2](#_bookmark2), we

designs. Section [3](#_bookmark7) discusses two major sources of error in stochastic circuits and introduces the proposed methodology in the noisy envi-

ronment for correlation-sensitive logic elements. With several initial

*𝑦* = 0 *𝑦* = 1

1 − *𝑝𝑒 𝑝𝑒*

⎡

⎤

*𝑀* = 1 − *𝑝𝑒 𝑝𝑒*

⎢

1 − *𝑝𝑒 𝑝𝑒*

⎢ ⎥

⎣ *𝑝𝑒* 1 − *𝑝𝑒*⎦

The rows in *𝐽* and *𝑀* correspond to input combinations 00*,* 01*,* 10, 11. Columns correspond to outputs *𝑦* = 0*, 𝑦* = 1. *𝑃 𝑇 𝑀* for a large circuit can be calculated from the *𝑃 𝑇 𝑀* of the individual gates and the way

they are interconnected, using two basic operations [[16](#_bookmark69)]:

* The overall *𝑃 𝑇 𝑀* of a circuit consisting of *𝑝* gates with *𝑃 𝑇 𝑀* s

*𝑀*1*, 𝑀*2*,* … *, 𝑀𝑝* connected in series, is obtained by multiplication of individual *𝑃 𝑇 𝑀𝑠*; *𝑀𝑠𝑒𝑟𝑖𝑒𝑠* = *𝑀*1*.𝑀*2*...𝑀𝑝*.

* The resultant *𝑃 𝑇 𝑀* of *𝑝* gates with *𝑃 𝑇 𝑀* s *𝑀*1*, 𝑀*2*,* … *, 𝑀𝑝* con-

multiplication of individual *𝑃 𝑇 𝑀* ; *𝑀𝑝𝑎𝑟𝑎𝑙𝑙𝑒𝑙* = *𝑀*1 *⊗ 𝑀*2 *⊗ 𝑀*3*... ⊗* nected in parallel is obtained by the Kronecker product or tensor

*𝑀𝑝*. Kronecker product or tensor multiplication multiplies each

element of a matrix A with size (*𝑚* × *𝑛*) with another matrix B of size (*𝑝* × *𝑞*) to give an output matrix C of size (*𝑚* × *𝑛*) × (*𝑝* × *𝑞*).

be represented via *𝑃 𝑇 𝑀* . We define an input vector of size 1 × 2*𝑘*, Input stochastic signals fed to a combinational circuit can also where *𝑘* is the total number of input signals which when multiplied by the overall circuit *𝑃 𝑇 𝑀* , *𝑀𝑐𝑘𝑡* gives the output *𝑃 𝑇 𝑀* . Let *𝑋* be

occur[rence of 1 g]iven as *𝑝𝑥*. *𝑋* is expressed as a two-element row vector a random variable with a Bernoulli distribution with a probability of

*𝑀* =

1 − *𝑝𝑥 𝑝𝑥* . For a combinational circuit with uncorrelated input

correlation assumptions, we have shown that a correct operating point of the circuit can be established with a suitably injected correlation that could generate an accurate result under the stated conditions. In Section [4](#_bookmark12), we extended the idea to simulate complex SLCs to show

bitstreams *𝑋* and *𝑌* having probabilities *𝑝𝑥* and *𝑝𝑦* and output signal *𝑍*

having a probability *𝑝𝑧*, we can represent the input vector *𝐼𝑖𝑛* using

*𝑃 𝑇 𝑀* as tensored product of two parallel identity signal matrices as,

*𝐼* = [(1 − *𝑝* ) *𝑝* ] *⊗* [(1 − *𝑝* ) *𝑝* ]

the effectiveness of the proposed model. Two approaches based on a

*𝑖𝑛*

*𝑥 𝑥*

*𝑦 𝑦*

priority-search model are demonstrated following two distinct condi- tions. The applicability of the proposed methodology in the context of an image processing task is discussed in Section [5](#_bookmark36). In Section [6](#_bookmark42),

= [(1 − *𝑝𝑥*)(1 − *𝑝𝑦*) (1 − *𝑝𝑥*)*𝑝𝑦 𝑝𝑥𝑝𝑦 𝑝𝑥𝑝𝑦*] which is a 4-element vec- tor. The output distribution *𝑍* is given by the matrix product of *𝐼𝑖𝑛* and the PTM of the circuit *𝑀𝑐𝑘𝑡* using Eq. ([1](#_bookmark1)).

highlights of the experimental results are jotted down with the pros and cons of the proposed algorithms.

*𝑍* = *𝐼*

*𝑖𝑛*

*.𝑀𝑐𝑘𝑡*

= [(1 − *𝑝𝑧*) *𝑝𝑧*] (1)

# Probability transfer matrices

The *𝑃 𝑇 𝑀* , which is used in analysing probabilistic logic circuits [[16](#_bookmark69)] has proved to be a convenient tool in error [[17](#_bookmark70)] and reliability anal- ysis [[18](#_bookmark71)] of small stochastic circuits. The PTM is derived from Ideal

Transfer Matrix (*𝐼 𝑇 𝑀* ) of a logic circuit. Each entry in *𝐼 𝑇 𝑀 𝐽* (*𝑖, 𝑗*)

of a logic circuit with *𝑖* = *𝑖*0*, 𝑖*1*,* … *, 𝑖𝑛*−1 and *𝑗* = *𝑗*0*, 𝑗*1*,* … *, 𝑗𝑚*−1 depicts the logic behaviour excited by a set of inputs. A 1 is observed in the matrix where a particular input combination generates a 1 in the truth

table. It can be observed as a conditional probability matrix so that,

*𝐽* (*𝑖, 𝑗*) = *𝑝*(*𝑜𝑢𝑡𝑝𝑢𝑡* = *𝑗*|*𝑖𝑛𝑝𝑢𝑡* = *𝑖*), where *𝑝* represents the conditional

*𝐼𝑖𝑛* can also be written in terms of bit overlaps in the bitstreams

*𝐼𝑖𝑛* = [*𝑖*0 *𝑖*1 *𝑖*2 *𝑖*3] = [*𝑛*00 *𝑛*01 *𝑛*10 *𝑛*11]

where, *𝑖*0*, 𝑖*1*, 𝑖*2*, 𝑖*3 represents the input probabilities *𝑝𝑥* and *𝑝𝑦* being 00*,* 01*,* 10 and 11 respectively. Thus, it can represent correlation between input signals as well. When *𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 1, maximum overlap of the bitstreams is guaranteed [[17](#_bookmark70)]. When *𝑝𝑥 > 𝑝𝑦*, the overlap 01 never occurs, so *𝑛*01 = 0. The probabilities of the other three overlaps 00, 10 and 11 are respectively given as (1 − *𝑝𝑥*), (*𝑝𝑦* − *𝑝𝑥*) and *𝑝𝑦*. We can

therefore express the input vector for two maximally correlated inputs as

*𝐼*+1 = [(1 − *𝑝𝑥*) 0 (*𝑝𝑥* − *𝑝𝑦*) *𝑝𝑦*] *, 𝑝𝑥 > 𝑝𝑦* (2)

probability of a particular output being true given a certain input

= [(1 − *𝑝* ) (*𝑝* − *𝑝* ) 0 *𝑝* ] *, 𝑝*

*> 𝑝*

(3)

combination. In Ideal Transfer Matrix (ITM), when the gate is assumed

*𝑦 𝑦 𝑥*

*𝑥 𝑥 𝑦*

to be error free, elements are either 0 or 1, representing exact binary values instead of probabilities.

Similarly, for two negatively correlated inputs the 11 overlap never occurs if *𝑝𝑥* +*𝑝𝑦* ≤ 1. Similarly, 00 overlap will never appear if *𝑝𝑥* +*𝑝𝑦* ≥ 1.

In *𝑃 𝑇 𝑀* (*𝑀* ) each entry contains a real value in the interval [0*,* 1]

*𝐼* = [1 − (*𝑝*

+ *𝑝* ) *𝑝 𝑝*

0] *, 𝑝*

+ *𝑝*

≤ 1 (4)

associated with output error probability information. For large circuits, −1

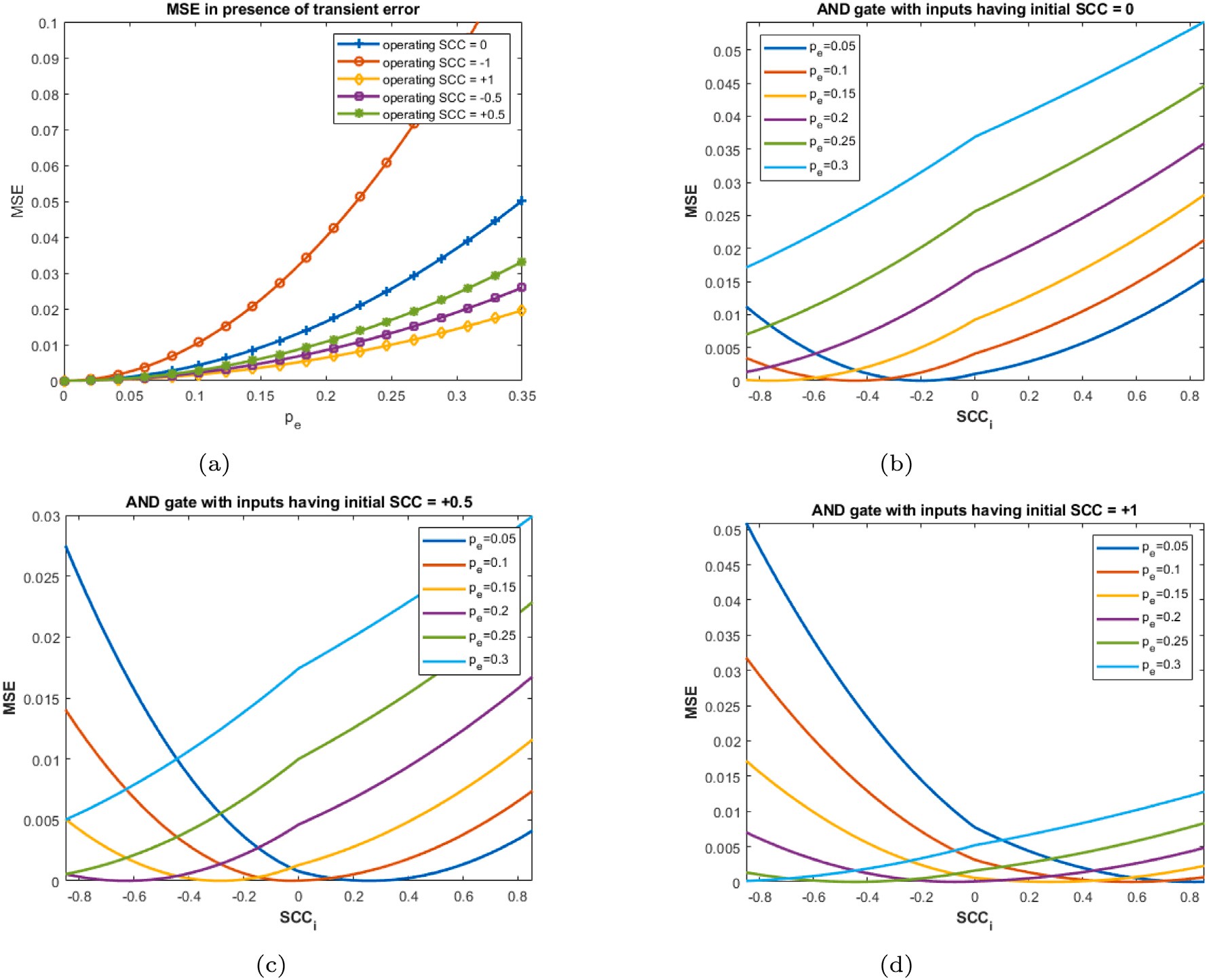
*𝑥 𝑦*

*𝑦 𝑥*

*𝑥 𝑦*

computation with *𝑃 𝑇 𝑀* is tedious. But for a circuit with *𝑘* inputs and

= [0 (1 − *𝑝𝑥*) (1 − *𝑝𝑦*) (*𝑝𝑥* + *𝑝𝑦*) − 1] *, 𝑝𝑥* + *𝑝𝑦* ≥ 1 (5)



**/ig. 1.** (a) MSE of AND gate with varying transient errors; Min. *𝑀𝑆𝐸* with *𝑅𝑒𝐶𝑜* for different initial *𝑆𝐶𝐶*s (b) *𝑆𝐶𝐶* = 0 (c) *𝑆𝐶𝐶* = 0*.*5 (d) *𝑆𝐶𝐶* = 1.

For two uncorrelated numbers the input vector is written using Eq. ([6](#_bookmark5))

where, *𝑝*(*𝑗 𝑖*) is the (*𝑖, 𝑗*)*𝑡ℎ* entry of *𝑃 𝑇 𝑀* . Using Eq. ([7](#_bookmark8)), reliability of the circuit at *𝑆𝐶𝐶* = 0 is obtained as (1 − *𝑝𝑒*). For AND gate, *𝑅𝑐𝑘𝑡* for

*𝐼* = [(1 − *𝑝* )(1 − *𝑝* ) (1 − *𝑝* )*𝑝*

|

0

*𝑥*

*𝑦*

*𝑥*

*𝑦*

*𝑝* (1 − *𝑝* ) *𝑝 𝑝* ] (6)

different ranges of *𝑆𝐶𝐶* can be obtained using Eq. ([8](#_bookmark6))

With the help of Eqs. ([2](#_bookmark3))–([6](#_bookmark5)), we can write the input vector matrix

*𝑥*

*𝑦*

*𝑥 𝑦*

*𝐼𝑆𝐶𝐶* for any intermediate *𝑆𝐶𝐶* value. These calculations are useful in

given inputs. *𝑃 𝑇 𝑀* s are used to study sequential circuits in the same deriving the stochastic functions given the correlation status between

way they are used to study and analyse combinational circuits [[18](#_bookmark71)].

* 1. *Reliability measure for circuits with varying correlation*

*𝑅𝑐𝑘𝑡*

⎧(*𝑝𝑒* − 1)(*𝑆𝐶𝐶*(1 − *𝑝𝑥* − *𝑝𝑦* + *𝑝𝑥.𝑝𝑦*) + *𝑝𝑥𝑝𝑦* − 1)+ (*𝑆𝐶𝐶*(*𝑝𝑥* + *𝑝𝑦* − 1) − *𝑝𝑥.𝑝𝑦*(*𝑆𝐶𝐶* + 1))(*𝑝𝑒* − 1)

= ⎪⎪= 1 − *𝑝𝑒,* ∀ *𝑆𝐶𝐶 <* 0

⎪

⎨⎪(*𝑝𝑒* − 1)(*𝑆𝐶𝐶.𝑝𝑥* + *𝑝𝑥.𝑝𝑦* − *𝑆𝐶𝐶.𝑝𝑥.𝑝𝑦* − 1)−

⎪(*𝑆𝐶𝐶.𝑝𝑥* − *𝑝𝑥.𝑝𝑦.*(*𝑆𝐶𝐶* − 1))(*𝑝𝑒* − 1)

⎪⎩= 1 − *𝑝𝑒,* ∀ *𝑆𝐶𝐶 >* 0

(8)

We are often concerned with the reliability of circuits under noisy conditions. The reliability of a circuit is defined as its ability to produce a correct output on a regular basis. For stochastic circuits, it can be

evaluated using the circuit’s *𝐼 𝑇 𝑀* (*𝐽* ) and *𝑃 𝑇 𝑀* (*𝑀* ). It can be shown

that the reliability of the circuit does not change with changes in correlation, but rather depends on the probabilistic error in the circuit.

**Definition 1.** The reliability of a circuit *𝑅𝑐𝑘𝑡* is invariant to change in correlation between inputs and depends on the probabilistic error rate

‘*𝑝𝑒*’.

Circuit reliability [[18](#_bookmark71)] is a measure of the similarity between its

*𝐼 𝑇 𝑀* and *𝑃 𝑇 𝑀* and is written as:

Eq. ([8](#_bookmark6)) demonstrates that the circuit’s reliability is independent of the correlation between the input numbers. Thus, error minimization by varying correlation has no effect on the circuit’s reliability. This

property aids subsequent analysis of *𝑆𝐿𝐸* in producing the corrected

output in the presence of transient error.

# Major error sources

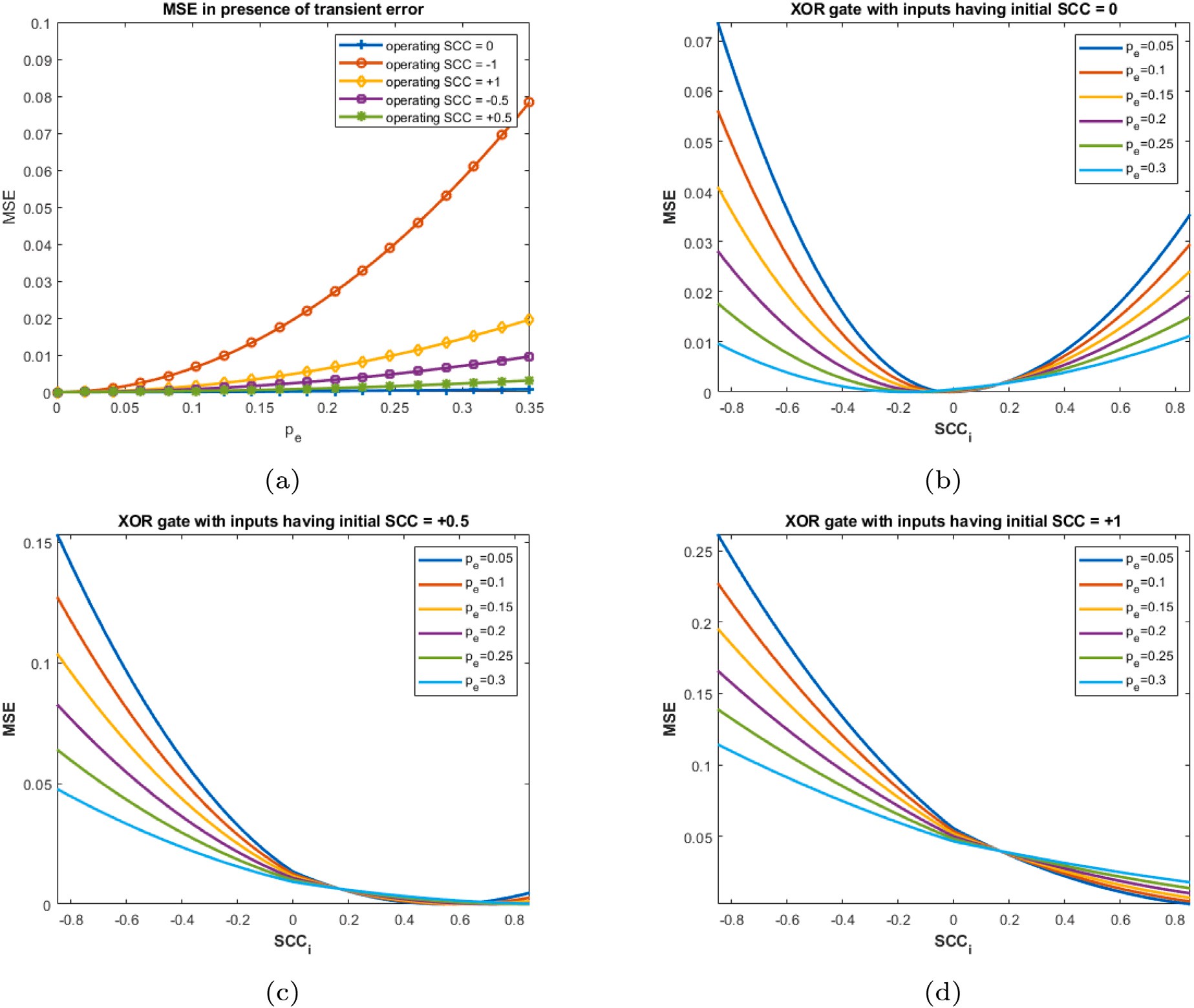
Circuits are becoming more vulnerable to transient faults as feature sizes, operating voltages, and design margins continue to shrink. Cos- mic rays, capacitive coupling, electromagnetic interference, and power transients are some of the leading physical phenomena. Transient faults induced by radiation have gained a lot of attention in recent years because they are seen as a potential roadblock to further technological advancement. In stochastic computing another major challenge is the

*𝑅𝑐𝑘𝑡* = ∑

*𝐽* (*𝑖,𝑗*)=1

*𝑝*(*𝑗*|*𝑖*)*.𝑝*(*𝑖*) (7)

correlation induced errors that are often associated with reordering of bits caused due to bit flips, which may or may not appear as a change



**/ig. 2.** (a) *𝑀𝑆𝐸* of XOR gate with varying transient errors; Min. *𝑀𝑆𝐸* with *𝑅𝑒𝐶𝑜* for different initial *𝑆𝐶𝐶*s (b) *𝑆𝐶𝐶* = 0 (c) *𝑆𝐶𝐶* = 0*.*5 (d) *𝑆𝐶𝐶* = 1.

in the probability of a given number when a single level circuit is considered. But for multi-level circuits this can effectively change the overall value. In the next section we focus on two major error sources and study their effect on the behaviour on logic circuits.

where, *𝑝𝑥*∧*𝑦* is obtained by bitwise AND operation between *𝑋* and *𝑌* .

Other generalized way of representing the SCC is:

*𝑛*11 *.𝑛*00 −*𝑛*01 *.𝑛*10

⎧⎪

*𝑛.𝑚𝑖𝑛*(*𝑛*11 +*𝑛*10 *,𝑛*11 +*𝑛*01 )−(*𝑛*11 +*𝑛*10 )(*𝑛*11 +*𝑛*01 )

*𝑆𝐶𝐶*(*𝑋, 𝑌* ) = ⎪

⎨

, *𝑛*11*.𝑛*00 *> 𝑛*01*.𝑛*10

* 1. *Correlation errors*

⎪

Correlation between two bitstreams has been identified as a ma- jor source of inaccuracy in certain stochastic circuits. Correlation in Stochastic computing indicates that the bitstreams generated by LSFR [[19](#_bookmark72)] or SNG [[20](#_bookmark73)] inherit some sort of dependence between them (cross-correlation) or between the bits of the same bitstream (auto-correlation). Earlier, correlation in stochastic circuits could only be vaguely identified as inaccurate output caused by a pair of bit- streams when passed through an AND gate. But, recently, correla- tion in stochastic computing has been quantified and identified with definiteness [[20](#_bookmark73)].

To quantify the correlation between input bitstreams *𝑋* and *𝑌* , *𝑆𝐶𝐶*

(Stochastic Correlation Coefficient) which is analogous to the similarity coefficient [[21](#_bookmark74)] is represented as

*𝑛*11 *.𝑛*00 −*𝑛*01 *.𝑛*10

(*𝑛*11 +*𝑛*10 )(*𝑛*11 +*𝑛*01 )−*𝑛.𝑚𝑎𝑥*(*𝑛*11 −*𝑛*00 *,*0)

⎪

⎪⎩ ,otherwise

where, *𝑛*11, *𝑛*10, *𝑛*01 and *𝑛*00 are the respective bit overlaps of *𝑋* and

*𝑌* . Thus, the measure of correlation is influenced only by the overlap of similar and dissimilar bits in the bitstreams. Let, *𝑋* = 110011110100, and *𝑌* = 010011110100, then, *𝑆𝐶𝐶* = +1. But, if *𝑋* = 101100010101, *𝑌* = 111111000101, *𝑆𝐶𝐶* = 0*.*5. In this case, not every 1 in Y is influenced by the presence of 1 in that position in *𝑋*. There is overlapping of 0′*𝑠* in *𝑋* and 1′*𝑠* in *𝑌* as well as 1′*𝑠* in *𝑋* and 0′*𝑠* in *𝑌* . Thus, the pair of bitstreams

is positively correlated to a certain degree. Correlation has also been found to have a positive effect on the circuit’s behaviour [[22](#_bookmark75),[23](#_bookmark76)]. XOR gate acts as an absolute subtractor when inputs are positively correlated, as shown in [Fig.](#_bookmark10) [3](#_bookmark10). Implementing the same function using binary inputs increases hardware complexity [[24](#_bookmark77)].

But for boundary values of probability, either, 0 or 1, the measure

⎧ *𝑝𝑋*∧*𝑌* −*𝑝𝑋* ⋅*𝑝𝑌* , *𝑝*

⎪

*𝑆𝐶𝐶*(*𝑋, 𝑌* ) = ⎨ *𝑚𝑖𝑛*(*𝑝𝑋 ,𝑝𝑌* )−*𝑝𝑋 𝑝𝑌*

*𝑋*∧*𝑌*

*> 𝑝𝑋*

⋅ *𝑝𝑌*

of *𝑆𝐶𝐶* becomes indeterminate. In both of these cases, it is impossible

to change the *𝑆𝐶𝐶* value with the help of any external circuit such as

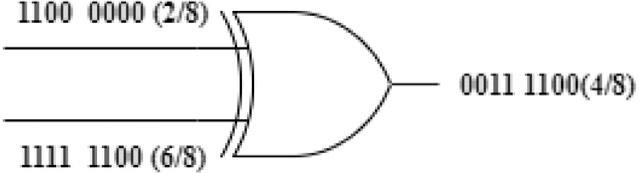
*𝑝𝑋*∧*𝑌* −*𝑝𝑋* ⋅*𝑝𝑌* , otherwise

⎪

⎩ *𝑝𝑋 𝑝𝑌* −*𝑚𝑎𝑥*(*𝑝𝑋* +*𝑝𝑌* −1*,*0)

a correlator. To relate to this, consider two SNs *𝑋*

= 00000000 and *𝑌* =



**/ig. 3.** XOR gate as absolute subtractor when inputs are positively correlated.

11111111. Logic operations on these numbers tend to produce output that will stick to the boundary values itself, either 0 or 1 depending

on the SLE. Attempts to change the correlation status will result in a change in probability value, which is undesired. Using a correlator circuit such as [[25](#_bookmark78)] will not be able to alter the degree of correlation

between *𝑋* and *𝑌* because only grouping of one kind of bit-pair (here

01) will be possible and we lose the leverage of pairing other three bit pairs i.e., 00*,* 10*,* 11. For any degree of correlation, we can write the output *𝑝𝑧* as a linear combination of its functions at *𝑆𝐶𝐶* = 0 and

*𝑆𝐶𝐶* = +1 or −1 [[20](#_bookmark73)] given as:

*𝑝𝑧* = *𝑓* (*𝑝𝑥, 𝑝𝑦*) = (1 + *𝑆𝐶𝐶*)*𝐹*0 − *𝑆𝐶𝐶.𝐹*−1*,* ∀ *𝑆𝐶𝐶 <* 0 (9)

*𝑝𝑧* = *𝑓* (*𝑝𝑥, 𝑝𝑦*) = (1 − *𝑆𝐶𝐶*)*𝐹*0 + *𝑆𝐶𝐶.𝐹*+1*,* ∀ *𝑆𝐶𝐶 >* 0 (10)

*𝐹*0*, 𝐹*−1*, 𝐹* +1 are the functions realized by the logic at *𝑆𝐶𝐶* = 0*,* −1*,* +1 respectively. Consider an AND gate with inputs *𝑝𝑥* and *𝑝𝑦*. For different correlation status between inputs, different functions are realized; *𝐹*0 =

*𝑝𝑥𝑝𝑦* (when inputs are uncorrelated), *𝐹*+1 = *𝑚𝑖𝑛*(*𝑝𝑥, 𝑝𝑦*) (positively

correlated) and *𝐹*−1 = *𝑚𝑎𝑥*(*𝑝𝑥* + *𝑝𝑦* − 1*,* 0) (negatively correlated).

Similarly, functions realized by XOR gate when the inputs are

uncorrelated, positively correlated, can be found out as,

*𝐹*0 = *𝑝𝑥*(1 − *𝑝𝑦*) + *𝑝𝑦*(1 − *𝑝𝑥*) = *𝑝𝑥* + *𝑝𝑦* − 2*𝑝𝑥𝑝𝑦* (11)

*𝐹*1 = |*𝑝𝑥* − *𝑝𝑦*| (12)

* 1. *Soft errors*

As semiconductor technology advances with reduced feature size and increased scalability, it is becoming more prone to soft errors [[6](#_bookmark59)]. The sources of such soft errors have been traced to mainly alpha particles and high energy cosmic rays [[26](#_bookmark79)]. Although soft errors are not unique to stochastic circuits, its properties make it more tolerant to soft errors than weighted-binary logic circuits. Soft errors do not affect the circuit physically, but they introduce behavioural changes in the circuit in the form of bit flips by introducing false logic [[27](#_bookmark80),[28](#_bookmark81)]. Transient or soft errors are caused due to exposure to external radiation and are greatly increased by manufacturing defects in a chip. These introduce false logic at the output of the circuit [[6](#_bookmark59)]. So, if multiple faults strike

modelled as bit flip error *𝑝𝑒* associated with each gate in the circuit as nodes of a gate, the output may be obtained erroneously. Bit flips are

a Bernoulli variable.

(BRV) represented by their probability of success *𝑝𝑥* to perform similar Stochastic numbers are analysed as Bernoulli random variables

using Mean Square Error (MSE) written as *𝐸𝑧* = *𝐸*[(*𝑝𝑧𝑒* − *𝑝𝑧*)2], where, operations as with other BRVs. Errors in BRVs are usually analysed

*𝑝𝑧𝑒* and *𝑝𝑧* represent the estimated and exact value respectively. A lot

of applications involving stochastic circuits are carried out in a noisy

environment where the circuit is prone to bit-flip errors. For nano-scale devices, transient or soft errors are growing prominence as the device features are downscaled to sub-micron ranges. The observed output might exceed the error threshold due to the change in the expected value of signals and also due to unwanted correlation introduced during bit flips. For larger circuits, this may be a major concern for accuracy [[29](#_bookmark82)].

The presence of soft errors coupled with other inherent error sources may cause model instability in multiple responses. Thus, to achieve the desired level of accuracy irrespective of the environment is a dire

need in this scenario. Soft errors can change the status of correlation between bitstreams that may or may not change the probability value. If an equal number of 1s and 0s are flipped in a bitstream on account of transient faults, then the probability value remains unchanged. However, if the number of bit-changes is unequal that may have an effect on changing the overall probability value. [Fig.](#_bookmark13) [4](#_bookmark13) shows the effect

logic elements (*𝑆𝐿𝐸*s). When fault-free, the AND gate implements of transient errors on the behaviour of a correlation sensitive stochastic

cases, where, *𝑝𝑒* at 0.125 hit the input nodes at different bit positions multiplication of two numbers. This condition is not true for two other

leading to shifted correlation status between two numbers as shown in [Fig.](#_bookmark13) [4](#_bookmark13)(b) and (c).

As we increase the transient error the *𝑀𝑆𝐸* increases exponentially.

In case of inputs operating in the negative range of correlation the error

surmounts with the incremental injection of soft error rates as shown

operating in the positive range of correlation will show reduced *𝑀𝑆𝐸* in [Fig.](#_bookmark4) [1](#_bookmark4)(a), [2](#_bookmark9)(a) and [6](#_bookmark19)(a)(red colour). Whereas the same bitstream

with the injection of soft errors (yellow colour). The responses of the correlation-sensitive logic elements like AND, OR and XOR gate with varying transient errors are captured in [Fig.](#_bookmark4) [1](#_bookmark4)(a),(b) and (c). These are discussed in detail in the next section, where the motto is to reduce the effect of the transient faults on probabilistic circuits by harnessing some of the unique properties of each of these correlation sensitive circuits.

# Handling errors in stochastic circuits using the proposed tech- nique

Minimizing errors is crucial since this distorts the output logic level of the circuit. We assume that transient faults at the gates introduced by external factors lead to change in the input as well as output probabilities thereby introducing uncertainty in correlation assumption of the circuit. It is observed that bit flips at different positions due to transient errors may lead to different correlation status between the same bitstreams. Undesired correlation can also lead to different stochastic functions being implemented by the same logic circuit, as shown in [Fig.](#_bookmark13) [4](#_bookmark13) and impedes the natural function to get implemented.

value if an unequal number of 0′*𝑠* and 1′*𝑠* are flipped. Thus, to realize Change in correlation status may also result in the change in probability

the accurate stochastic function and hence the output in this error scenario a correlation restoration scheme needs to be grabbed, that minimizes the effect of transient error in the circuit.

* 1. *The proposed Remodelling Correlation (ReCo) framework*

Our work suggests *Remodelling Correlation (ReCo)* technique to cater to the change in the probability assumption at the inputs owing to transient faults. We interpret techniques for correlation-sensitive ele-

ments to bring down the *𝑀𝑆𝐸* to a minimum level. While conducting

a study on correlation-sensitive SLEs we demonstrate that every design

error can be corrected by introducing correlation to a certain degree at the inputs. We deduce an operating point of the circuit in this

incorrect environment with a suitable injection of *𝑆𝐶𝐶* that reduces

*𝑀𝑆𝐸* to a minimum value. *𝐴𝑙𝑔𝑜𝑟𝑖𝑡ℎ𝑚*1 searches for a unique solution of the induced correlation within the range [−1,1] to find a minimum

error for input parameters. We begin our analysis by considering single

*𝑆𝐿𝐸*s. The flowchart of the proposed framework is shown in [Fig.](#_bookmark16) [5](#_bookmark16).

* + 1. *𝑅𝑒𝐶𝑜 analysis for correlation sensitive logic elements with zero correlation assumption*

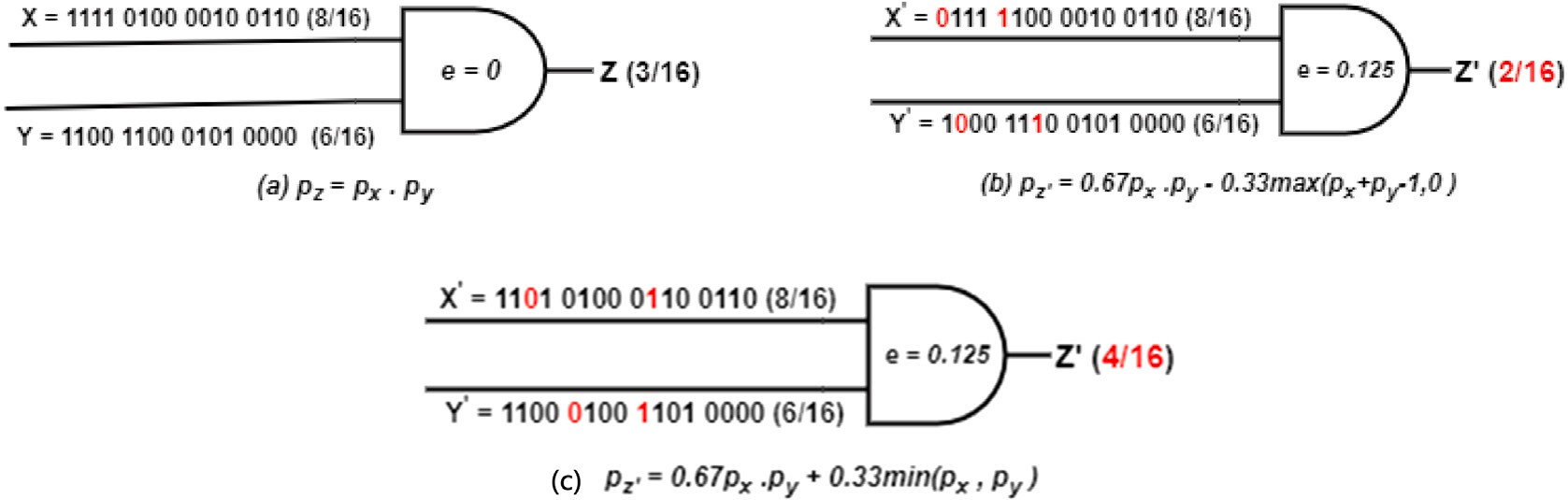
A stochastic circuit implements different real-valued functions when

function to be implemented at *𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 0 and any deviation is the correlation between input numbers is altered. We assume the target

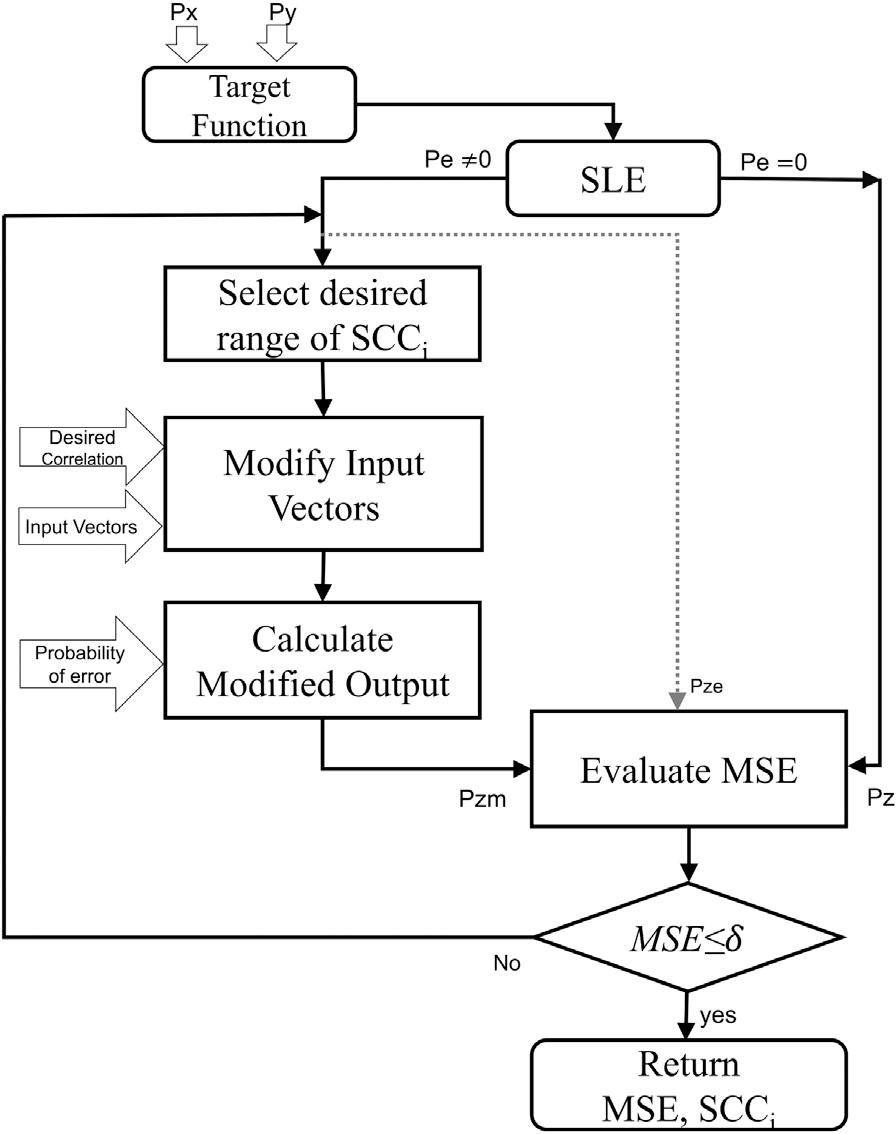
considered as faulty behaviour of the circuit.

noise. Since we have set *𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 0, we can evaluate *𝑝𝑧* = *𝑝𝑥𝑝𝑦* to *(i) AND gate:* Consider an AND gate that is inflicted by transient

be true. As we increase the probability of transient error, the observed



**/ig. 4.** Bit flips at different positions due to transient errors and its impact on correlation alteration resulting in different Stochastic functions.

= (1 + *𝑆𝐶𝐶𝑖*)(1 − *𝑝𝑦*)*𝑝𝑥* − *𝑆𝐶𝐶𝑖𝑝𝑥* (15)

*𝑖*11*𝑚* = (1 + *𝑆𝐶𝐶*)*𝑖*11(*𝐼*0 ) − *𝑆𝐶𝐶𝑖.𝑖*11(*𝐼*−1 )

= (1 + *𝑆𝐶𝐶𝑖*)*𝑝𝑥𝑝𝑦* (16)

where, *𝑖*00(*𝐼*0 ) , *𝑖*01(*𝐼*0 ) , *𝑖*10(*𝐼*0 ) , *𝑖*11(*𝐼*0 ) are four elements of input vector *𝐼*0. Similarly, *𝑖*00(*𝐼*−1 ) , *𝑖*01(*𝐼*−1 ) , *𝑖*10(*𝐼*−1 ) , and *𝑖*11(*𝐼*−1 ) are four elements of vector

*𝐼*−1. Now, the modified vectors of *𝐼𝑆𝐶𝐶𝑚* can be represented as *𝐼𝑆𝐶𝐶𝑚* =

[ ]

*𝑖*00*𝑚 𝑖*01*𝑚 𝑖*10*𝑚 𝑖*11*𝑚* . Reducing Eqs. ([13](#_bookmark17))–([16](#_bookmark14)) further we get,

⊺

⎡⎢ 1 − (*𝑝𝑥* + *𝑝𝑦*) + *𝑝𝑥𝑝𝑦*(1 + *𝑆𝐶𝐶𝑖*)⎤

*𝐼𝑆𝐶𝐶𝑚* = ⎢

⎥

⎢⎣

– *𝑝𝑦*(*𝑝𝑥* + *𝑝𝑥𝑆𝐶𝐶𝑖* − 1)

– *𝑝𝑥*(*𝑝𝑦* + *𝑝𝑦𝑆𝐶𝐶𝑖* − 1)

*𝑝𝑥𝑝𝑦*(*𝑆𝐶𝐶𝑖* + 1) ⎦

⎥

(17)

Similarly, for *𝑝𝑥* + *𝑝𝑦 >* 1,

⊺

⎡⎢ −(*𝑝𝑦* − 1)(*𝑝𝑥𝑆𝐶𝐶𝑖* − *𝑝𝑥* + 1) ⎤⎥

*𝐼* = ⎢

*𝑝𝑦*(*𝑆𝐶𝐶𝑖* − 1)(*𝑝𝑥* − 1) − *𝑆𝐶𝐶𝑖*(*𝑝𝑥* − *𝑝𝑦*)⎥

(18)

*𝑆𝐶𝐶𝑚* ⎢

⎢

⎥

⎣

*𝑝𝑥*(*𝑆𝐶𝐶𝑖* − 1)(*𝑝𝑦* − 1)

*𝑝𝑥𝑆𝐶𝐶𝑖* − *𝑝𝑥𝑝𝑦*(*𝑆𝐶𝐶𝑖* − 1) ⎦

⎥

**/ig. 5.** Flowchart of the proposed framework.

increase in *𝑀𝑆𝐸*, indicated in [Fig.](#_bookmark4) [1](#_bookmark4)(a) (blue). So we employ the output deviates more from the original output showing an exponential proposed method with an aim to reduce the *𝑀𝑆𝐸*.

**Algorithm 1:** *𝑅𝑒𝐶𝑜 analysis for a single gate*

1: **Input** *𝑝𝑥, 𝑝𝑦 , 𝑝𝑒 , 𝑖𝑛𝑝𝑢𝑡*\_*𝐺𝑎𝑡𝑒*; **Output** *𝑀𝑆𝐸𝑖, 𝑆𝐶𝐶𝑖*

2: ***ReCo***(*𝑖𝑛𝑝𝑢𝑡*\_*𝐺𝑎𝑡𝑒*)

3: [*𝑝𝑥, 𝑝𝑦* ]= input probabilities of *𝑖𝑛𝑝𝑢𝑡*\_*𝐺𝑎𝑡𝑒*;

4: *𝑝𝑒* = probabilities of transient errors of *𝑖𝑛𝑝𝑢𝑡*\_*𝐺𝑎𝑡𝑒*; 5: *𝑇 𝑟𝑢𝑒*\_*𝑂𝑢𝑡𝑝𝑢𝑡* = ***Eval***(*𝑝𝑥, 𝑝𝑦 , 𝑆𝐶𝐶*);

6: **for** *𝑆𝐶𝐶𝑖* = −1; *𝑆𝐶𝐶𝑖 <*= +1; *𝑆𝐶𝐶𝑖* + = 0*.*001 **do**

*𝑀𝑜𝑑𝑖𝑓 𝑖𝑒𝑑*\_*𝑂𝑢𝑡𝑝𝑢𝑡* = ***Eval***(*𝑝𝑥, 𝑝𝑦 , 𝑝𝑒 , 𝑆𝐶𝐶𝑖* );

*𝑀𝑆𝐸𝑖* = *𝑇 𝑟𝑢𝑒*\_*𝑂𝑢𝑡𝑝𝑢𝑡* − *𝑀𝑜𝑑𝑖𝑓 𝑖𝑒𝑑*\_*𝑂𝑢𝑡𝑝𝑢𝑡*

**if** *𝑀𝑆𝐸𝑖* ≤ *𝛿* **then**

return *𝑀𝑆𝐸𝑖* , *𝑆𝐶𝐶𝑖*

**end if**

**end for**

We modify each element of the input vector *𝑖*00*, 𝑖*01*, 𝑖*10, and *𝑖*11 be-

tween *𝑝𝑥* and *𝑝𝑦* to form a new vector consisting of modified elements.

7: return argmin

*𝑀𝑆𝐸𝑖*

{*𝑀𝑆𝐸𝑖* , *𝑆𝐶𝐶𝑖* }

For two independent inputs with a zero correlation assumption, we

express the input vector as,

*𝐼*0 = (1 − *𝑝𝑥*)(1 − *𝑝𝑦*) (1 − *𝑝𝑥*)*𝑝𝑦 𝑝𝑥*(1 − *𝑝𝑦*) *𝑝𝑥𝑝𝑦* , assuming *𝑝𝑥 < 𝑝𝑦*

[

]

For an AND gate with a given error rate *𝑝𝑒*, the modified output *𝑝𝑧𝑚*

as a function of *𝐼𝑆𝐶𝐶* can be written as,

*𝑚*

and *𝑝𝑥* + *𝑝𝑦* ≤ 1. Let the injected correlation *𝑆𝐶𝐶𝑖* be in the range of 0

to +1. Fo[r two negatively correlate]d inputs,

⎡1 − *𝑝𝑒 𝑝𝑒* ⎤

*𝐼*−1 = 1 − (*𝑝𝑥* + *𝑝𝑦*) *𝑝𝑦 𝑝𝑥* 0 . The modified input vectors are

*𝑝* = *𝐼*

× ⎢1 − *𝑝𝑒 𝑝𝑒* ⎥

(19)

calculated using Eq. ([9](#_bookmark11))

*𝑧𝑚*

*𝑆𝐶𝐶𝑚*

⎢1 − *𝑝𝑒 𝑝𝑒* ⎥

*𝑖*00

= (1 + *𝑆𝐶𝐶𝑖*)*𝑖*00(*𝐼* ) − *𝑆𝐶𝐶𝑖.𝑖*00(*𝐼* )

⎢⎣ *𝑝𝑒* 1 − *𝑝𝑒*⎥⎦

= (1 + *𝑆𝐶𝐶𝑖*)(1 − *𝑝𝑥*)(1 − *𝑝𝑦*) − *𝑆𝐶𝐶𝑖*{1 − (*𝑝𝑥* + *𝑝𝑦*)} (13)

*𝑚*

0

−1

*𝑖*01*𝑚* = (1 + *𝑆𝐶𝐶𝑖*)*𝑖*01(*𝐼*0 ) − *𝑆𝐶𝐶𝑖.𝑖*01(*𝐼*−1)

= (1 + *𝑆𝐶𝐶𝑖*)(1 − *𝑝𝑥*)*𝑝𝑦* − *𝑆𝐶𝐶𝑖𝑝𝑦* (14)

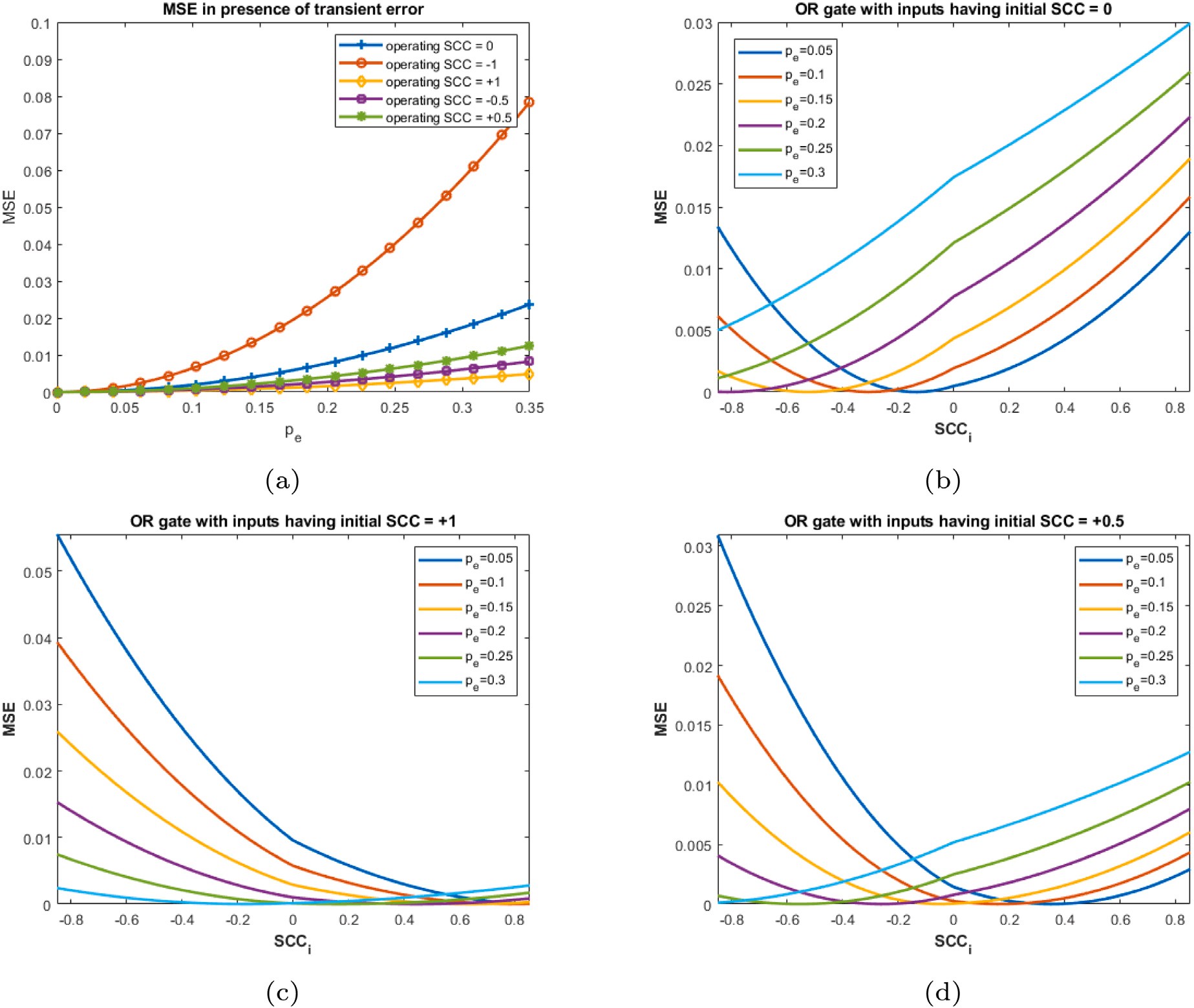
*𝑖*10*𝑚* = (1 + *𝑆𝐶𝐶𝑖*)*𝑖*10(*𝐼*0 ) − *𝑆𝐶𝐶𝑖.𝑖*10(*𝐼*−1)

Thus, *𝑝𝑧𝑚* is observed as *𝑓* (*𝑆𝐶𝐶𝑖*). We try to make *𝑝𝑧𝑚* close to *𝑝𝑧* to

reduce the observed error *𝑝𝑧𝑒*. Thus,

*𝑝𝑧𝑚* = *𝑝𝑒* + *𝑝𝑥𝑝𝑦*(1 + *𝑆𝐶𝐶𝑖*)(1 − 2*𝑝𝑒*) (20) The *𝑀𝑆𝐸* which is (*𝑝𝑧𝑚* − *𝑝𝑧*)2 is calculated as,

*𝑀𝑆𝐸𝑎𝑛𝑑* = {*𝑝𝑒* + *𝑝𝑥𝑝𝑦*(*𝑆𝐶𝐶𝑖* − 2*𝑝𝑒* − 2*𝑝𝑒𝑆𝐶𝐶𝑖*)}2 (21)



**/ig. 6.** (a) *𝑀𝑆𝐸* of OR gate with varying transient errors; Min. *𝑀𝑆𝐸* obtained using *𝑅𝑒𝐶𝑜* for different initial *𝑆𝐶𝐶*s (b) *𝑆𝐶𝐶* = 0 (c) *𝑆𝐶𝐶* = 1 (d) *𝑆𝐶𝐶* = 0*.*5.

The induced *𝑆𝐶𝐶𝑖* which reduces *𝑀𝑆𝐸* to a minimum possible value within the range [−1,+1] is obtained by differentiating Eq. ([21](#_bookmark18)) w.r.t

*𝑝𝑧𝑒* = 0*.*28 at *𝑝𝑒* = 0*.*15. Thus,

⊺

*𝑆𝐶𝐶* and equate it to 0.

2*𝑝𝑥𝑝𝑦*(1 − 2*𝑝𝑒*){*𝑝𝑒* + *𝑝𝑥𝑝𝑦*(*𝑆𝐶𝐶𝑖* − 2*𝑝𝑒* − 2*𝑝𝑒𝑆𝐶𝐶𝑖*)} = 0

*𝐼𝑆𝐶𝐶𝑚*

= ⎡⎢

⎢

⎢

⎢

⎣

(0*.*28 + 0*.*18*𝑆𝐶𝐶𝑖*)⎤

(0*.*42 − 0*.*18*𝑆𝐶𝐶𝑖*)⎥

(0*.*12 − 0*.*12*𝑆𝐶𝐶𝑖*)⎥⎥

(0*.*18 + 0*.*12*𝑆𝐶𝐶𝑖*)⎦

−(*𝑝𝑒* − 2*𝑝𝑒𝑝𝑥𝑝𝑦*)

Thus, *𝑝*

= 0*.*18*𝑆𝐶𝐶* + 0*.*64*𝑝*

– 0*.*36*𝑝 𝑆𝐶𝐶*

+ 0*.*18 and *𝑀𝑆𝐸* =

∴*𝑆𝐶𝐶𝑖* =

(22)

*𝑝 𝑝* (1 − 2*𝑝* )

(9*𝑆𝐶𝐶*

*𝑧𝑚* −18*𝑝 𝑆𝐶𝐶* )2 *𝑖 𝑒*

*𝑒 𝑖*

*𝑥 𝑦 𝑒*

*𝑖* +32*𝑝𝑒*

*𝑒 𝑖*

. Error reduces to 0 for *𝑆𝐶𝐶*

= −0*.*76. It is

Eq. ([22](#_bookmark20)) dictates the condition of reaching a minimum value of

*𝑀𝑆𝐸*

40000

observed that *𝑀𝑆𝐸* can be reduced to 0

if *𝑝𝑒* ≤

0*.*2

*𝑖*

(a considerate limit).

for *𝑆𝐶𝐶𝑖* in the range [−1,0]. Similarly, when *𝑝𝑥* + *𝑝𝑦 >* 1 with *𝑝𝑥 < 𝑝𝑦*,

*𝑆𝐶𝐶𝑖* can be evaluated as,

(*𝑝𝑒* − *𝑝𝑥* − *𝑝𝑦* + *𝑝𝑥𝑝𝑦* − 2*𝑝𝑒𝑝𝑥𝑝𝑦* + 1)

The results are confirmed graphically considering different values of *𝑝𝑥*

and *𝑝𝑦* at *𝑝𝑒* = 0*.*125 as shown in [Fig.](#_bookmark22) [7](#_bookmark22)(a).

*𝑆𝐶𝐶𝑖* =

(2*𝑝* − 1)(*𝑝*

– 1)(*𝑝*

– 1) (23)

1. *XOR gate:* At *𝑆𝐶𝐶* = 0, the XOR gate implements *𝑝𝑧* = *𝑝𝑥*(1 −

where, *𝑝 , 𝑝*

*𝑒*

* 0 and *𝑝*

*𝑥 𝑦*

*< 𝑝* . Thus, *𝑝 𝑝*

* 0. Expressions are derived

*𝑝𝑦*) + *𝑝𝑦*(1 − *𝑝𝑥*). Any deviation from the target function on account

*𝑥 𝑦*

*𝑥 𝑦*

*𝑥 𝑦*

of transient error is considered as a contribution to MSE. A similar

derivation prevents *𝑆𝐶𝐶𝑖* from being positive to achieve the minimum assuming negative induction of correlation. However, nothing in the

MSE.

To exploit the simplicity of equations and to achieve the maximum possible accuracy in calculations, parameters appearing in equations are verified graphically. [Fig.](#_bookmark4) [1](#_bookmark4)(b) shows different values of induced

operate at a minimum *𝑀𝑆𝐸* under different transient error rates. Now, foregoing approach is adopted in the analysis of the XOR gate to

*𝑝𝑧𝑚* for *𝑝𝑥 < 𝑝𝑦* is written as,

⎡⎢1 − *𝑝𝑒 𝑝𝑒* ⎤⎥

correlation to obtain zero error at the output at different error rates.

*𝑝* = *𝐼*

× ⎢ *𝑝𝑒* 1 − *𝑝𝑒*⎥

(24)

Note that, Eqs. ([22](#_bookmark20)), ([23](#_bookmark21)) always hold for *𝑝 <* 0*.*5. Using similar

*𝑧𝑚*

*𝑆𝐶𝐶𝑚*

⎢ *𝑝𝑒* 1 − *𝑝𝑒*⎥

*𝑒* ⎢1 − *𝑝*

*𝑝* ⎥

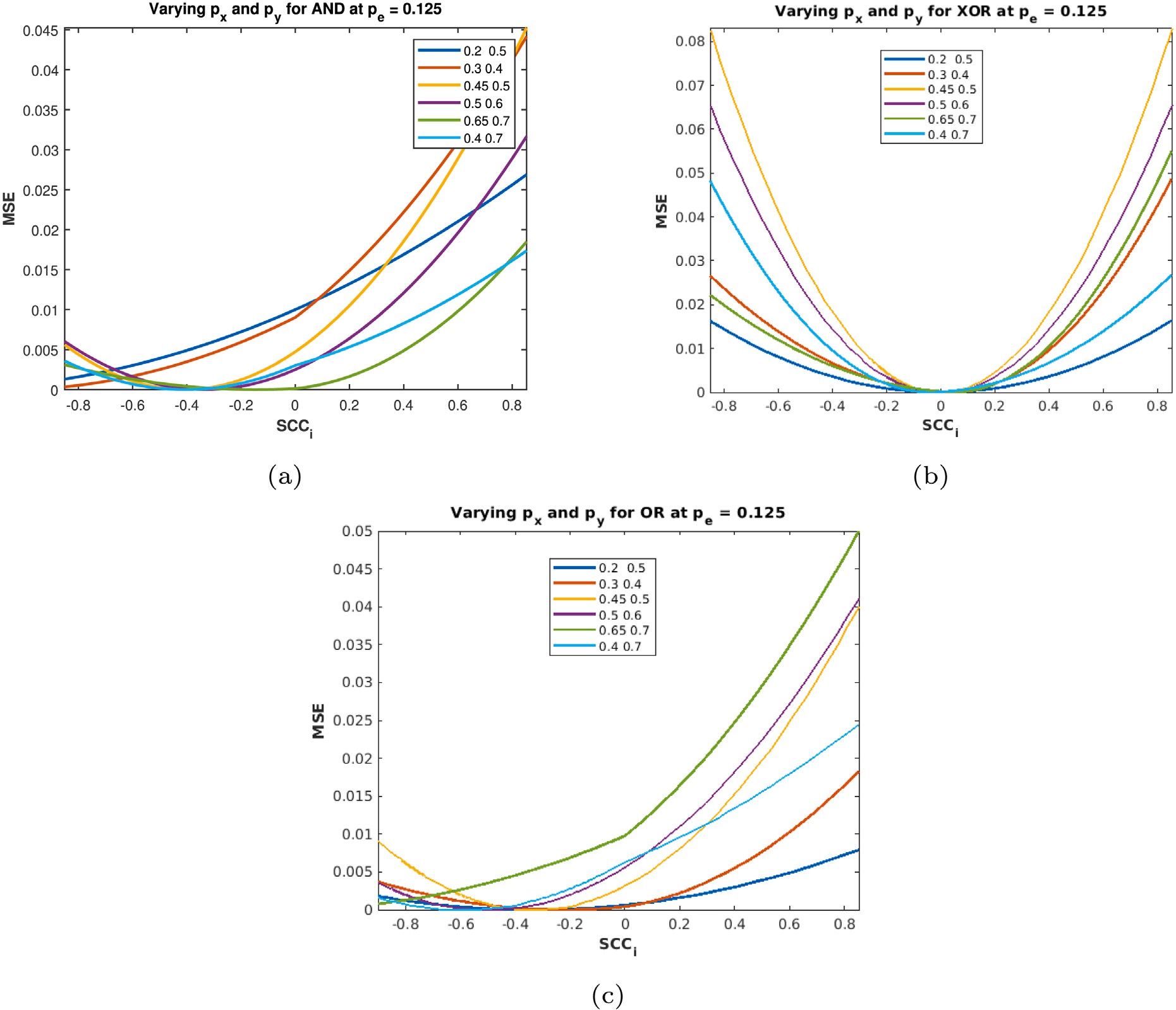
analysis, we arrive at different sets of equations for *𝑝𝑥 > 𝑝𝑦*.

**Example 1.** Consider an AND gate with *𝑝𝑥* = 0*.*3 and *𝑝𝑦* = 0*.*6. The

⎣ *𝑒 𝑒* ⎦

Substituting *𝐼𝑆𝐶𝐶𝑚* from Eq. ([17](#_bookmark15)),

error-free output is *𝑝𝑧* = 0*.*3 × 0*.*6 = 0*.*18. The observed output is *𝑝𝑧𝑚* = *𝑝𝑒* − *𝑝𝑥𝑝𝑦* − 2*𝑝𝑒*(*𝑝𝑥* + *𝑝𝑦* − 2*𝑝𝑥𝑝𝑦*) − 2*𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*(1 − 2*𝑝𝑒*)



**/ig. 7.** *𝑀𝑆𝐸* after *𝑅𝑒𝐶𝑜* analysis of correlation sensitive gates with *𝑝𝑒* = 0*.*125 subjected to transient error 0.125 for different input probabilities; (a) AND gate (b) XOR gate (c) OR gate.

*𝑀𝑆𝐸*

= {2*𝑝* (*𝑝* + *𝑝* ) − *𝑝* + 2*𝑝 𝑝* (*𝑆𝐶𝐶* − 2*𝑝* − 2*𝑝 𝑆𝐶𝐶* )}2 (25)

Thus for 0 ≤ *𝑝𝑒* ≤ 0*.*3, the error can be reduced to 0 in contrary to other [Fig.](#_bookmark9) [2](#_bookmark9) (b) that the XOR gate is most sensitive to changes in correlation.

*𝑥𝑜𝑟*

*𝑒 𝑥 𝑦 𝑒*

*𝑥 𝑦*

*𝑖 𝑒 𝑒 𝑖*

gates, AND and OR (see [Figs.](#_bookmark4) [1](#_bookmark4)(b) and [6](#_bookmark19)(b)). This is also validated using

We differentiate Eq. ([25](#_bookmark23)) w.r.t *𝑆𝐶𝐶𝑖* and equate it to 0.

– 2(2*𝑝 𝑝* − 4*𝑝 𝑝 𝑝* )(*𝑝* − 2*𝑝 𝑝* − 2*𝑝 𝑝* −

different values of *𝑝𝑥* and *𝑝𝑦* at *𝑝𝑒* = 0*.*125 which is shown in [Fig.](#_bookmark22) [7](#_bookmark22)(b).

1. *OR gate:* For uncorrelated numbers, OR gate implements *𝑝𝑧* =

*𝑥 𝑦*

*𝑒 𝑥 𝑦 𝑒*

*𝑒 𝑥*

*𝑒 𝑦*

*𝑝𝑥* + *𝑝𝑦* − *𝑝𝑥𝑝𝑦*. In presence of transient error let the function be *𝑝𝑧𝑒*. We

2*𝑝𝑥𝑝𝑦*(*𝑆𝐶𝐶𝑖* + 2*𝑝𝑒* + 2*𝑝𝑒𝑆𝐶𝐶𝑖*) = 0

∴*𝑆𝐶𝐶* = *𝑝𝑒* − 2*𝑝𝑒*(*𝑝𝑥* + *𝑝𝑦*) + 4*𝑝𝑒𝑝𝑥𝑝𝑦*

(26)

desired correlation. The modified output *𝑝𝑧𝑚* is then calculated using eliminate this error by introducing the *ReCo* block at inputs to inject the

Eq. ([17](#_bookmark15)).

*𝑖* 2*𝑝𝑥𝑝𝑦* − 4*𝑝𝑒𝑝𝑥𝑝𝑦*

⎡1 − *𝑝𝑒*

*𝑝𝑒* ⎤

Similarly, for *𝑝𝑥* + *𝑝𝑦 >* 1,

*𝑝𝑧𝑚*

= *𝐼𝑆𝐶𝐶* ×

⎢

*𝑚*

*𝑝𝑒* 1 − *𝑝𝑒*

*𝑝𝑒* 1 − *𝑝𝑒*⎥

⎥

*𝑝𝑒* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* + 4*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑆𝐶𝐶𝑖* = 4*𝑝* + 2(*𝑝* + *𝑝* )(1 − 2*𝑝* ) − 2*𝑝 𝑝* (1 + 2*𝑝* ) − 2 (27)

⎢⎣ *𝑝𝑒*

1 − *𝑝𝑒*⎥⎦

*𝑒 𝑥 𝑦*

*𝑒 𝑥 𝑦 𝑒*

∴*𝑝𝑧𝑚* = *𝑝𝑒* + *𝑝𝑥* + *𝑝𝑦* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* − *𝑝𝑥𝑝𝑦*−

[Fig.](#_bookmark9) [2](#_bookmark9)(b) shows different values of *𝑆𝐶𝐶𝑖* to reach zero *𝑀𝑆𝐸* at

different values of *𝑝**𝑒*, which is consistent with Eq. ([26](#_bookmark24)).

*𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖* + 2*𝑝𝑒𝑝𝑥𝑝𝑦* + 2*𝑝𝑒𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*

**Example 2.** Consider XOR gate with inputs *𝑝𝑥*

= 0*.*3 and *𝑝𝑦*

= 0*.*6.

∴*𝑀𝑆𝐸𝑜𝑟* = (*𝑝𝑒*{1 + *𝑝𝑥𝑝𝑦*} − 2*𝑝𝑒*{*𝑝𝑥* + *𝑝𝑦*} − *𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*{1 − 2*𝑝𝑒*})2 (28)

Thus *𝑝𝑧* = 0*.*54 and *𝑝𝑧𝑒* = 0*.*52 at *𝑝𝑒* = 0*.*15. By substituting *𝐼𝑆𝐶𝐶*

we find *𝑝𝑧𝑚* = 0*.*72*𝑝𝑒* − 0*.*08*𝑝𝑒* − 0*.*36*𝑆𝐶𝐶𝑖* + 0*.*54 and hence *𝑀𝑆𝐸* =

*𝑚*

We differentiate Eq. ([28](#_bookmark26)) w.r.t *𝑆𝐶𝐶𝑖* and equate it to 0.

−2(*𝑝 𝑝* − 2*𝑝 𝑝 𝑝* )(*𝑝* − 2*𝑝 𝑝* − 2*𝑝 𝑝*

(9*𝑆𝐶𝐶𝑖* +2*𝑝𝑒* −18*𝑝𝑒 𝑆𝐶𝐶𝑖* )2 . Thus, *𝑀𝑆𝐸* reduces to 0 at *𝑆𝐶𝐶* = 0.

*𝑥 𝑦*

*𝑒 𝑥 𝑦 𝑒*

*𝑒 𝑥*

*𝑒 𝑦*

625

*𝑖* −*𝑝 𝑝 𝑆𝐶𝐶* + 2*𝑝 𝑝 𝑝*

+ 2*𝑝 𝑝 𝑝 𝑆𝐶𝐶* ) = 0

From the experiment (see [Fig.](#_bookmark9) [2](#_bookmark9)(a)) it is inferred that the XOR

*𝑥 𝑦*

*𝑖 𝑒 𝑥 𝑦*

*𝑒 𝑥 𝑦 𝑖*

contributor to the overall *𝑀𝑆𝐸* in the circuit. Also, it is evident from gate is least responsive to probabilistic errors and hence the smallest

∴*𝑆𝐶𝐶* = *𝑝𝑒* − 2*𝑝𝑒*(*𝑝𝑥* + *𝑝𝑦*) + 2*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑖 𝑝𝑥𝑝𝑦* − 2*𝑝𝑒𝑝𝑥𝑝𝑦*

(29)

Similarly for *𝑝𝑥* + *𝑝𝑦 >* 1,

(*𝑝*

– *𝑝* + 2*𝑝 𝑝*

+ *𝑝 𝑝* )

*𝑥 𝑒*

*𝑒 𝑥*

*𝑥 𝑦*

*𝑆𝐶𝐶𝑖* =

*𝑒 𝑥 𝑦*

2*𝑝𝑒*(*𝑝𝑦* − *𝑝𝑥*) − *𝑝𝑒* + *𝑝𝑥𝑝𝑦*(1 − 2*𝑝𝑒*)

(30)

*𝑝 𝑝* (1 − 2*𝑝* )

*𝑆𝐶𝐶𝑖* =

*𝑥*

*𝑝*

– 2*𝑝𝑒𝑝𝑥*

– *𝑝𝑥𝑝𝑦*

+ 2*𝑝 𝑝 𝑝 , 𝑝𝑥* + *𝑝𝑦 >* 1 (35)

*𝑥 𝑦 𝑒*

error at the output of OR gate for different values of *𝑝𝑒*. [Fig.](#_bookmark19) [6](#_bookmark19)(b) shows different values of induced correlation to obtain zero

**Example 3.** Consider an OR gate with inputs *𝑝𝑥* = 0*.*3 and *𝑝𝑦* = 0*.*6

at *𝑝𝑒* = 0*.*15. The error free output *𝑝𝑧* = 0*.*72 and the observed output

*(b) Any positive intermediate correlation, 𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 0*.*5: Now

say +0*.*5. The function implemented by AND logic at *𝑆𝐶𝐶* = 0*.*5 is consider any intermediate positive correlation between the numbers,

*𝑝𝑧* = 0*.*5*𝑝𝑥*(1 + *𝑝𝑦*) for *𝑝𝑥 < 𝑝𝑦*. In presence of transient error in the circuit, we modify input vectors as *𝐼𝑆𝐶𝐶𝑚* ,

*𝑝𝑧𝑒* = 0*.*654. Using Eq. [2](#_bookmark25) *𝑝𝑧𝑚* = 0*.*36*𝑝𝑒𝑆𝐶𝐶𝑖* −0*.*44*𝑝𝑒* −0*.*18*𝑆𝐶𝐶𝑖* + 0*.*72 and

⎡ (1 − *𝑝* )(0*.*5*𝑝 𝑆𝐶𝐶* − *𝑝*

+ 1) ⎤

= (9*𝑆𝐶𝐶* +22*𝑝* −18*𝑝 𝑆𝐶𝐶* −6)2

⎢ 0 5

*𝑦*

(1 −

*𝑥 𝑖 𝑥*

(0 5 5 − )

⊺

⎥

*𝑀𝑆𝐸*

*𝑖 𝑒*

*𝑒 𝑖*

. Thus, *𝑀𝑆𝐸* can be reduced to zero at

⎢ *. 𝑝𝑦*

*𝑝𝑥*) − *𝑆𝐶𝐶𝑖*

*. 𝑝𝑥* + 0*. 𝑝𝑦*

*𝑝𝑥𝑝𝑦* ⎥

*𝑆𝐶𝐶* = −0*.*5238.

*𝑖*

2500

*𝐼𝑆𝐶𝐶𝑚* (+0*.*5) = ⎢

*𝑝* (*𝑆𝐶𝐶* − 1)(*𝑝*

– 1) ⎥

Thus, similar to the AND gate, the *𝑀𝑆𝐸* of the OR gate can be

*𝑥*

*𝑖*

*𝑦*

although the reduction is less than that of the *𝐴𝑁𝐷* gate. reduced to zero if the transient error is below a certain limit, say 0.2,

From the analysis, it is observed that the OR gate is least sensitive to changes in correlation, whereas XOR is the highest. AND gate is intermediate to them. It is also identified that the XOR gate is least affected by transient error, whereas AND gate is mostly influenced by

the presence of transient error. So *𝑀𝑆𝐸* increases immensely when

the error is imposed on an AND gate. These properties of the XOR

gate make it a suitable choice for the analysis of an error-resilient

⎢⎣ *𝑝𝑥.*(*𝑝𝑦* + 0*.*5*𝑆𝐶𝐶𝑖* − 0*.*5*𝑝𝑦𝑆𝐶𝐶𝑖*) ⎥⎦

*𝑝𝑧𝑚* = *𝐼𝑆𝐶𝐶𝑚* (+0*.*5) × *𝑀𝑎𝑛𝑑* = 0*.*5*𝑆𝐶𝐶𝑖𝑝𝑥*(*𝑝𝑦* − *𝑝𝑒* + *𝑝𝑒𝑝𝑦*)

+*𝑆𝐶𝐶𝑖𝑝𝑥*(0*.*5 − *𝑝𝑒* + *𝑝𝑒𝑝𝑦*) + *𝑝𝑥𝑝𝑦*(1 − 2*𝑝𝑒*) + *𝑝𝑒*

*𝑀𝑆𝐸𝑎𝑛𝑑* = *𝑝𝑒*(1 − 2*𝑝𝑥𝑝𝑦*) − *𝑝𝑥*(1 − *𝑝𝑦*) (0*.*5 + *𝑝𝑒𝑆𝐶𝐶𝑖*) + 0*.*5*𝑝𝑥*(1 − *𝑝𝑦*)(1 − *𝑝𝑒*)*𝑆𝐶𝐶𝑖*

Differentiating Eq. ([37](#_bookmark28)) w.r.t *𝑆𝐶𝐶𝑖* and putting it to 0, gives

0*.*5*𝑝𝑥*(1 − *𝑝𝑦*) + 2*𝑝𝑒𝑝𝑥𝑝𝑦* − *𝑝𝑒*

(36)

(37)

circuit design. In the next section, this idea is implemented on complex

*𝑆𝐶𝐶𝑖* =

0*.*5*𝑝* (1 − *𝑝* )(1 − 3*𝑝* ) (38)

*𝑥 𝑦 𝑒*

circuits that focus to minimize *𝑀𝑆𝐸* with minimum hardware using

the proposed methodology.

* + 1. *ReCo analysis for correlation-sensitive logic elements with non-zero*

*𝑆𝐶𝐶𝑖* =

2*𝑝𝑒𝑝𝑥* − *𝑝𝑥* − 2*𝑝𝑒* + *𝑝𝑥𝑝𝑦* + 2*𝑝𝑒𝑝𝑥𝑝𝑦*

2*𝑝𝑥* − 4*𝑝𝑒𝑝𝑥* − 2*𝑝𝑥𝑝𝑦* + 4*𝑝𝑒𝑝𝑥𝑝𝑦*

*, 𝑝𝑥* + *𝑝𝑦 >* 1*.* (39)

*correlation assumption*

Those SLEs which are sensitive to correlation implement an alto- gether different stochastic function. An example is shown with the help of an AND gate in [Fig.](#_bookmark13) [4](#_bookmark13)(b),(c). In this section, we reconsider SLEs with transient errors having an apriori correlation assumption. We

invoke ReCo analysis to suppress *𝑀𝑆𝐸* and formulate the underlying

assumption are discussed, i.e., *𝑆𝐶𝐶* = +0*.*5 and *𝑆𝐶𝐶* = +1 and perform conditions in support of that. Two distinct cases of initial correlation

a similar analysis to reduce errors at different degrees of transient faults. The target function is obtained considering an initial non-zero and positive value of correlation. It is observed that for an existing negative SCC between input variables the effect of transient errors in the circuit element is enhanced. Thus, such cases are excluded in our analysis.

initial correlation between *𝑝𝑥* and *𝑝𝑦*. The intersection of *𝑝𝑒* with the (i) *AND gate:* The analysis begins by setting a non-zero and positive

previously set positive value of correlation between inputs implicitly assumes that there is a shift in the value of correlation to arrive at the minimum MSE.

*(a) With existing 𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 1*:* For positively correlated numbers,

AND gate implements *𝑝𝑧* = *𝑚𝑖𝑛*(*𝑝𝑥, 𝑝𝑦*). We counter the effect of tran-

**Example 4.** Consider *𝑝𝑥* = 0*.*3, *𝑝𝑦* = 0*.*6 and *𝑝𝑒* = 0*.*15. From Eq. ([36](#_bookmark27)),

*𝑝𝑧𝑚* = 0*.*1*𝑆𝐶𝐶𝑖* +0*.*64*𝑝𝑒* −0*.*24*𝑝𝑒𝑆𝐶𝐶𝑖* +0*.*18. We invoke Eq. ([37](#_bookmark28)) to obtain

*𝑀𝑆𝐸* = 1*.*6 × 10−3(3*𝑆𝐶𝐶𝑖* + 16*𝑝𝑒* − 6*𝑝𝑒𝑆𝐶𝐶𝑖* − 3)2. Thus, for *𝑝𝑒* = 0*.*15,

*𝑀𝑆𝐸* can be reduced to zero by injecting *𝑆𝐶𝐶𝑖* = +0*.*2857.

With *𝑆𝐶𝐶* = +0*.*5 between inputs, *𝑝𝑧* and *𝑝𝑧𝑒* are 0.24 and 0.32. Using Eq. ([37](#_bookmark28)), *𝑝𝑧𝑚* = 0*.*1*𝑆𝐶𝐶𝑖* + 0*.*64*𝑝𝑒* − 0*.*24*𝑝𝑒𝑆𝐶𝐶𝑖* + 0*.*18. Thus,

*𝑀𝑆𝐸* = (3*𝑆𝐶𝐶𝑖* +32*𝑝𝑒* −9*𝑝𝑒 𝑆𝐶𝐶𝑖* −3)2 . Thus unlike the previous case, *𝑀𝑆𝐸*

can be reduced to zero only for *𝑝𝑒 <* 0*.*15 by injecting suitable positive

40000

*𝑆𝐶𝐶𝑖*.

1. *XOR gate:* We assume a positive definite correlation between the

inputs of an XOR gate and using similar analysis we derive from the condition for minimum MSE.

* 1. *With existing 𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 1*:* With positively correlated inputs,

the XOR gate implements *𝑝𝑧* = *𝐹*+1 = *𝑝𝑥* − *𝑝𝑦* . The deviation from this

| |

a suitable operating point of the circuit by defining *𝑆𝐶𝐶𝑖* using the assumption under the error scenarios can be encountered by finding

following derivations.

We modify the output by introducing the desired correlation such that,

sient error on the circuit by introducing a desired *𝑆𝐶𝐶𝑖* obtained using

following derivations.

⊺

*𝑝𝑧𝑚* = *𝑝𝑒* + *𝑝𝑥* + *𝑝𝑦* − 2*𝑆𝐶𝐶𝑖𝑝𝑥* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* − 2*𝑝𝑥𝑝𝑦*

+4*𝑝𝑒𝑝𝑥𝑆𝐶𝐶𝑖* + 2*𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖* + 4*𝑝𝑒𝑝𝑥𝑝𝑦* − 4*𝑝𝑒𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*

(40)

⎡⎢ −(*𝑝𝑦* − 1)(*𝑝𝑥𝑆𝐶𝐶𝑖* − *𝑝𝑥* + 1) ⎤⎥

*𝑀𝑆𝐸𝑥𝑜𝑟* = (*𝑝𝑒* + 2*𝑝𝑥* − 2*𝑝𝑒*(*𝑝𝑥* + *𝑝𝑦*)−

(41)

*𝐼* = ⎢

*𝑝𝑦*(*𝑆𝐶𝐶𝑖* − 1)(*𝑝𝑥* − 1) − *𝑆𝐶𝐶𝑖*(*𝑝𝑥* − *𝑝𝑦*)⎥

(31)

2*𝑝𝑥*(1 − 2*𝑝𝑒*){*𝑆𝐶𝐶𝑖*(1 + *𝑝𝑦*) + *𝑝𝑦*})2

*𝑆𝐶𝐶𝑚* (+1) ⎢

⎢

⎥

⎣

*𝑝𝑥*(*𝑆𝐶𝐶𝑖* − 1)(*𝑝𝑦* − 1)

*𝑝𝑥* − *𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*(*𝑆𝐶𝐶𝑖* − 1) ⎦

⎥

Differentiating Eq. ([41](#_bookmark29)) w.r.t *𝑆𝐶𝐶𝑖* and equating it to 0,

The modified output *𝑝*

is calculated as

*𝑝𝑒*(1 − 2*𝑝𝑥*)(1 − 2*𝑝𝑦*) + 2*𝑝𝑥*(1 − *𝑝𝑦*)

*𝑧𝑚*

*𝑆𝐶𝐶𝑖* =

2*𝑝* (2*𝑝* − 1)(*𝑝*

– 1) (42)

*𝑥 𝑒 𝑦*

*𝑝𝑧𝑚* = *𝐼𝑆𝐶𝐶𝑚*(+1) ×*𝑀𝑎𝑛𝑑* =*𝑝𝑒𝑝𝑥* − *𝑝𝑒*(*𝑝𝑥* + *𝑝𝑦* − 1) + *𝑝𝑒𝑝𝑦* (32)

*𝑝* − 2*𝑝*

+ 2*𝑝 .𝑝* − 2*𝑝 .𝑝*

+ 2*𝑝 .𝑝*

*𝑒 𝑥*

*𝑒 𝑥*

*𝑒 𝑦*

*𝑥 𝑦*

For *𝑝𝑥* + *𝑝𝑦* ≤ 1, modified *𝑀𝑆𝐸* is,

*𝑆𝐶𝐶𝑖* =

2*𝑝*

– 4*𝑝 .𝑝*

– 2*.𝑝 .𝑝*

+ 4*𝑝 .𝑝 .𝑝*

(43)

*𝑀𝑆𝐸𝑎𝑛𝑑* = (*𝑝𝑒* − *𝑝𝑥* + *𝑝𝑥*(1 − 2*𝑝𝑒*){*𝑆𝐶𝐶𝑖*(1 − *𝑝𝑦*) + *𝑝𝑦*})2 (33)

*𝑥 𝑒 𝑥*

*𝑥 𝑦*

*𝑒 𝑥 𝑦*

Differentiating Eq. ([33](#_bookmark30)) w.r.t *𝑆𝐶𝐶𝑖* and equate it to 0,

**Example 5.** Consider *𝑝𝑥* = 0.3, *𝑝𝑦* = 0.6, *𝑝𝑒* = 0*.*15. Thus, *𝑝𝑧* = 0*.*3 and

*𝑝𝑧𝑒* = 0*.*36 and *𝑝𝑧𝑚* = 0*.*48*𝑝𝑒𝑆𝐶𝐶𝑖* − 0*.*0800*𝑝𝑒* − 0*.*24*𝑆𝐶𝐶𝑖* + 0*.*54. Now,

*𝑝* − *𝑝* − *𝑝 𝑝* (1 − 2*𝑝* )

*𝑀𝑆𝐸* = (9*𝑆𝐶𝐶*+2*𝑝𝑒* −18*𝑝𝑒 𝑆𝐶𝐶*−6)2 , which can be reduced to zero when

*𝑆𝐶𝐶* = *𝑥 𝑒*

*𝑥 𝑦 𝑒*

(34)

625

*𝑖 𝑝𝑥*(2*𝑝𝑒* − 1)(*𝑝𝑦* − 1)

*𝑆𝐶𝐶𝑖* = 0*.*933.



**/ig. 8.** OR gate implementing different functions when inputs have different correlation status.

* 1. *Any positive intermediate correlation, 𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 0*.*5*:* In this case, the resultant function is *𝑝𝑧* = *𝐹*+0*.*5 = −*𝑝𝑦*(*𝑝𝑥* −1) when *𝑝𝑥 < 𝑝𝑦*. The

To determine the induced *𝑆𝐶𝐶𝑖* that minimizes *𝑀𝑆𝐸* under specified

error scenarios, the following derivations will suffice.

error-free output is obtained using modified output defined by input

*𝑝* = *𝑝* + *𝑝*

– *𝑝 𝑝*

– 2*𝑝 𝑝*

+ 0*.*5*𝑝 𝑝 𝑆𝐶𝐶* + *𝑝 𝑝 𝑝*

vectors in *𝐼* .

*𝑧𝑚*

*𝑒 𝑦*

*𝑒 𝑥*

*𝑒 𝑦*

*𝑒 𝑥*

*𝑖 𝑒 𝑥 𝑦*

(51)

*𝑆𝐶𝐶𝑚*(+0*.*5)

0*.*5*𝑝 𝑝 𝑝 𝑆𝐶𝐶* − *𝑝* (*𝑆𝐶𝐶* − 1)(*𝑝* − 1)(*𝑝*

– 1)

*𝑒 𝑥 𝑦*

*𝑖 𝑥 𝑖*

*𝑒 𝑦*

*𝑝𝑧𝑚* = *𝑝𝑒* + *𝑝𝑥* + *𝑝𝑦* − 5*𝑆𝐶𝐶𝑖𝑝𝑥* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* − 2*𝑝𝑥𝑝𝑦*

Now, *𝑀𝑆𝐸* in this case is calculated as;

+10*𝑝𝑒𝑝𝑥𝑆𝐶𝐶𝑖* + 5*𝑆𝐶𝐶𝑖𝑝𝑥𝑝𝑦* + 4*𝑝𝑒𝑝𝑥𝑝𝑦* − 10*𝑝𝑒𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*

*𝑀𝑆𝐸*

= 0*.*25(2*𝑝* + *𝑝* − 2*𝑝 𝑆𝐶𝐶* − 4*𝑝 𝑝*

– 4*𝑝 𝑝*

– *𝑝 𝑝*

*𝑜𝑟*

*𝑒 𝑥 𝑥*

*𝑖 𝑒 𝑥*

*𝑒 𝑦*

*𝑥 𝑦*

2 (52)

*𝑀𝑆𝐸𝑥𝑜𝑟* = 0*.*25(2*𝑝𝑒* + 2*𝑝𝑥* − 3*𝑝𝑥𝑆𝐶𝐶𝑖* − 4*𝑝𝑒𝑝𝑥* − 4*𝑝𝑒𝑝𝑦* − 2*𝑝𝑥𝑝𝑦*

+5*𝑝 𝑝 𝑆𝐶𝐶* + 3*𝑝 𝑝 𝑆𝐶𝐶* + 8*𝑝 𝑝 𝑝* − 5*𝑝 𝑝 𝑝 𝑆𝐶𝐶* )2 (44)

+ 3*𝑝𝑒𝑝𝑥𝑆𝐶𝐶𝑖* + 2*𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖* + 4*𝑝𝑒𝑝𝑥𝑝𝑦* − 3*𝑝𝑒𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*)

2*𝑝𝑒* + *𝑝𝑥* − 4*𝑝𝑒𝑝𝑥* − 4*𝑝𝑒𝑝𝑦* − *𝑝𝑥𝑝𝑦* + 4*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑆𝐶𝐶* =

(53)

*𝑒 𝑥*

*𝑖 𝑥 𝑦*

*𝑖 𝑒 𝑥 𝑦*

*𝑒 𝑥 𝑦 𝑖*

*𝑖* (2*𝑝𝑥*

– 3*𝑝𝑒𝑝𝑥*

– 2*𝑝𝑥𝑝𝑦*

+ 3*𝑝𝑒𝑝𝑥𝑝𝑦*)

Differentiating Eq. ([44](#_bookmark33)) w.r.t *𝑆𝐶𝐶𝑖* and equate it to 0,

2*𝑝* (1 − 2*𝑝* )(1 − 2*𝑝* ) + 2*𝑝* (1 − *𝑝* )

Similarly, for *𝑝𝑥*

+ *𝑝𝑦 >* 1

*𝑒 𝑦 𝑥 𝑥 𝑦*

*𝑆𝐶𝐶𝑖* =

*𝑝* (5*𝑝* − 3)(*𝑝*

– 1) *, 𝑝𝑥* + *𝑝𝑦* ≤ 1 (45)

2*𝑝𝑒* − *𝑝𝑥* − 2*𝑝𝑒𝑝𝑥* − 4*𝑝𝑒𝑝𝑦* + *𝑝𝑥𝑝𝑦* + 2*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑆𝐶𝐶* =

(54)

*𝑥 𝑒 𝑦*

*𝑖* 2*𝑝𝑥* − 4*𝑝𝑒𝑝𝑥* − 2*𝑝𝑥𝑝𝑦* + 4*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑝𝑒* − *𝑝𝑥* − 2*𝑝𝑒.𝑝𝑦* + *𝑝𝑥.𝑝𝑦* + 2*𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑆𝐶𝐶𝑖* =

2*𝑝𝑥* − 4*𝑝𝑒.𝑝𝑥* − 2*𝑝𝑥.𝑝𝑦* + 4*𝑝𝑒𝑝𝑥.𝑝𝑦*

*, 𝑝𝑥* + *𝑝𝑦 >* 1 (46)

**Example 8.** Consider *𝑝𝑥* = 0.3, *𝑝𝑦* = 0.6 and *𝑝𝑒* = 0*.*15. Thus, *𝑝𝑧* = 0*.*66

With positively correlated inputs the induced *𝑆𝐶𝐶𝑖* can generally be

written in the form,

*𝑝𝑒*(1 − 2*𝑝𝑥*)(1 − 2*𝑝𝑦*) + 2*𝑆𝐶𝐶𝑝𝑥*(1 − *𝑝𝑦*)

and *𝑝𝑧𝑒* = 0*.*612. *𝑝𝑧𝑚* = 0*.*24*𝑝𝑒𝑆𝐶𝐶* − 0*.*44*𝑝𝑒* − 0*.*12*𝑆𝐶𝐶* + 0*.*72. Using

Eq. ([52](#_bookmark32)), *𝑀𝑆𝐸* = (9*𝑆𝐶𝐶*+22*𝑝𝑒* −18*𝑝𝑒 𝑆𝐶𝐶*−3)2 . Thus, an error-free output can be obtained when the injected correlation is −0*.*05.

40000

*𝑆𝐶𝐶𝑖* = −*𝑝* (*𝑝*

*𝑥 𝑦*

*𝑒 𝑒*

– 1)(*𝑆𝐶𝐶* − *𝑝* − 3*𝑆𝐶𝐶𝑝* + 1) (47)

positive range when the initial correlation between the numbers is +1 Thus for an OR gate, induced correlation are mostly obtained in the

**Example 6.** Let *𝑝𝑥* = 0*.*3, *𝑝𝑦* = 0*.*6 and *𝑝𝑒* = 0*.*15. Thus, *𝑝𝑧* = 0*.*42 and

*𝑝𝑧𝑒* = 0*.*444. And *𝑝𝑧𝑚* = 1*.*12*𝑝𝑒𝑆𝐶𝐶* − 0*.*08*𝑝𝑒* − 0*.*56*𝑆𝐶𝐶* + 0*.*54. Thus,

*𝑀𝑆𝐸* = 0*.*0016(14*𝑆𝐶𝐶𝑖* + 2*𝑝𝑒* − 28*𝑝𝑒𝑆𝐶𝐶𝑖* − 3)2, which can be reduced to zero when *𝑆𝐶𝐶𝑖* = +0*.*275.

between numbers, the injected *𝑆𝐶𝐶𝑖* values are predominantly in the as shown in [Fig.](#_bookmark19) [6](#_bookmark19)(c), while for any intermediate correlation existing negative range except for *𝑝𝑒* ≤ 0*.*1. The simulation results are given in [Fig.](#_bookmark19) [6](#_bookmark19)(d). For any positive correlation *𝑆𝐶𝐶*, the expressions can be

generalized as:

1. *OR gate:* Consider an OR gate with inputs that are positively

*𝑝* + *𝑆𝐶𝐶𝑝* (1 − *𝑝* ) − 2*𝑝* (*𝑝*

+ *𝑝* ) + 2*𝑝 𝑝 𝑝*

correlated. Based on different correlation status, OR gate implements

*𝑆𝐶𝐶* = *𝑒*

*𝑥 𝑦*

*𝑒 𝑥 𝑦*

*𝑒 𝑥 𝑦*

(55)

attempt to derive the condition for attaining the smallest *𝑀𝑆𝐸* under different stochastic functions as shown in [Fig.](#_bookmark31) [8](#_bookmark31). Using *ReCo*, we

transient error scenarios.

* 1. *With existing 𝑆𝐶𝐶*(*𝑋, 𝑌* ) = 1*:* The OR gate implements *𝑝𝑧* =

*𝑚𝑎𝑥*(*𝑝𝑥, 𝑝𝑦*) when two numbers are positively correlated. We can sim- ilarly find the operating *𝑆𝐶𝐶𝑖* to obtain a minimum *𝑀𝑆𝐸* under error

scenarios.

∴*𝑝𝑧𝑚* = *𝑝𝑒* + *𝑝𝑦* − *𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* + *𝑝𝑒𝑝𝑥𝑆𝐶𝐶𝑖* + *𝑝𝑒𝑝𝑥𝑝𝑦*

*𝑖 𝑝𝑥*(1 − *𝑝𝑦*)(1 − *𝑝𝑒*) − *𝑆𝐶𝐶.𝑝𝑒.𝑝𝑥*(1 − *𝑝𝑦*)

* 1. *The proposed error detector circuit*

Consider an SLE operating in a noisy environment. The system level representation of the error correction mechanism for such an SLE is

shown in [Fig.](#_bookmark35) [10](#_bookmark35). The inputs to the unit are *𝑝𝑥* and *𝑝𝑦*, error *𝑝𝑒* and

output is the desired value *𝑝𝑧𝑚*. The control circuit or the error detector

−*𝑝 𝑝 𝑝 𝑆𝐶𝐶* − *𝑝* (*𝑆𝐶𝐶* − 1)(*𝑝* − 1)(*𝑝*

(48)

– 1)

circuit is used to determine the amount of deviation of an erroneous

*𝑒 𝑥 𝑦*

*𝑖 𝑥 𝑖*

*𝑒 𝑦*

output from an error-free output. It comprises a subtractor, a squarer

*𝑀𝑆𝐸𝑜𝑟* = (*𝑝𝑒* + *𝑝𝑥* − *𝑝𝑥𝑆𝐶𝐶𝑖* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* − *𝑝𝑥𝑝𝑦*

+2*𝑝𝑒𝑝𝑥𝑆𝐶𝐶𝑖* + *𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖* + 2*𝑝𝑒𝑝𝑥𝑝𝑦* − 2*𝑝𝑒𝑝𝑥𝑝𝑦𝑆𝐶𝐶𝑖*)2

*𝑝𝑒* + *𝑝𝑥* − 2*𝑝𝑒𝑝𝑥* − 2*𝑝𝑒𝑝𝑦* − *𝑝𝑥𝑝𝑦* + 2*𝑝𝑒𝑝𝑥𝑝𝑦*

(49)

and a comparator which determines the amount of error that is to be reduced. Auxiliary circuits like the shuffle buffer and the synchronizer are used to adjust the correlation between the input bitstreams. The output of the control unit is fed to the *ReCo* block consisting of a

*𝑆𝐶𝐶𝑖* =

*𝑝* (2*𝑝* − 1)(*𝑝*

– 1)

correlator circuit that generates the desired *𝑆𝐶𝐶*

to minimize MSE.

*𝑥 𝑒 𝑦*

*𝑖*

The control circuit is described below.

∴*𝑆𝐶𝐶* = *𝑝𝑒* − *𝑝𝑥* − 2*𝑝𝑒.𝑝𝑦* + *𝑝𝑥.𝑝𝑦*

(50)

*𝑖 𝑝𝑥* − 2*𝑝𝑒.𝑝𝑥* − *𝑝𝑥.𝑝𝑦* + 2*𝑝𝑒.𝑝𝑥.𝑝𝑦*

**Example 7.** Consider *𝑝𝑥* = 0.3 and *𝑝𝑦* = 0.6, *𝑝𝑧* = 0*.*6 and *𝑝𝑧𝑒* = 0*.*57 for

*𝑝𝑒* = 0*.*15. *𝑝𝑧𝑚* = 0*.*24*𝑝𝑒𝑆𝐶𝐶* − 0*.*44*𝑝𝑒* − 0*.*12*𝑆𝐶𝐶* + 0*.*72 with the help of

above equations (70) and (71), *𝑀𝑆𝐸* = (9*𝑆𝐶𝐶*+22*𝑝𝑒* −18*𝑝𝑒 𝑆𝐶𝐶*−6)2 . Thus,

* + 1. *Synchronizer*

maximum number of input bits to 00 or 11, restoring their respective The synchronizer [[25](#_bookmark78)] is a finite state machine that pairs up a

probabilities. This unit is placed between two uncorrelated sequences,

*𝑝𝑧𝑒* and *𝑝𝑧* to introduce positive correlation between sequences *𝑝𝑧𝑒* and

40000

*𝑝* . If the bits in *𝑝*

*𝑠*

and *𝑝*

MSE can be reduced to zero for *𝑝𝑒* = 0*.*15 when *𝑆𝐶𝐶𝑖* = +0*.*432.

*𝑧𝑠*

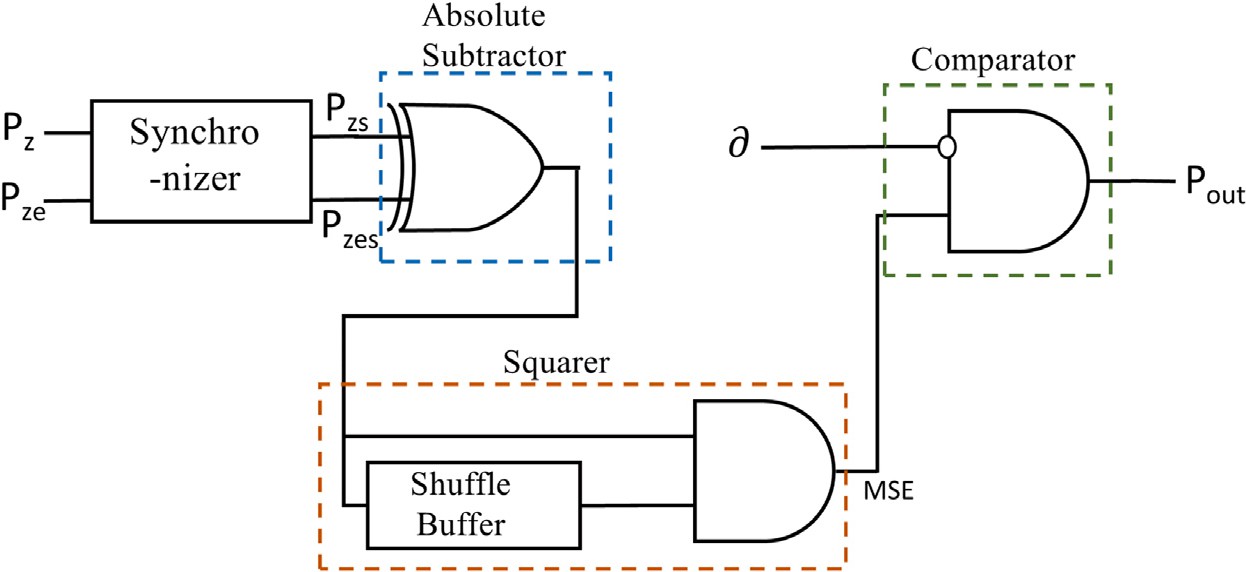
*𝑧𝑒*

*𝑧* are equal, then the corresponding bits are

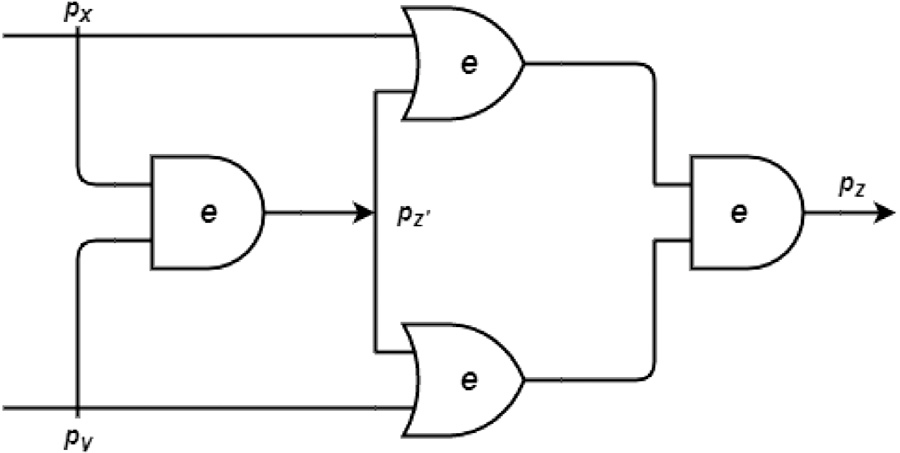
intermediate correlation, such as, *𝑆𝐶𝐶*(*𝑋, 𝑌* ) = +0*.*5, the stochastic (b) *Any positive intermediate correlation:* If we consider any positive function realized by OR gate is given by; *𝑝𝑧* = 0*.*5*𝑝𝑥* + *𝑝𝑦* − 0*.*5*𝑝𝑥𝑝𝑦*.

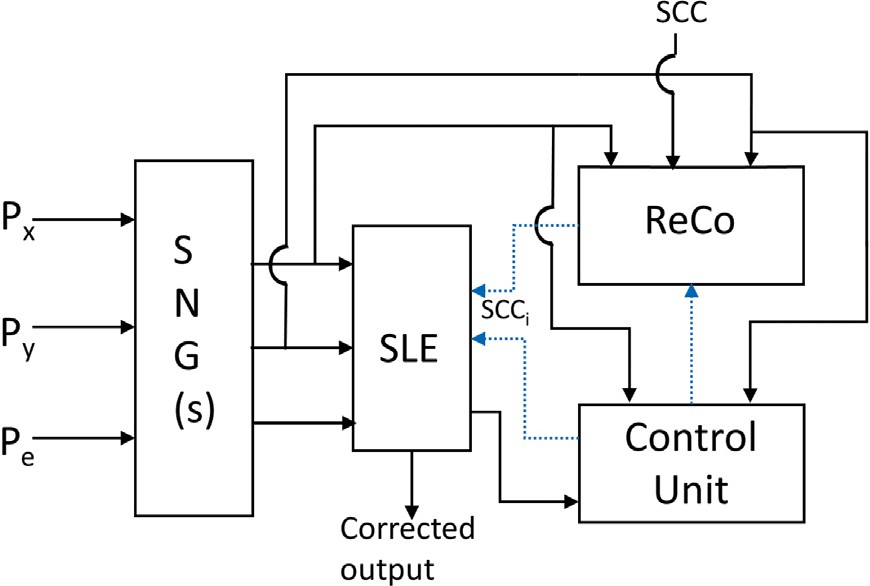
1 or 0, if they are in state *𝑆*1, are changed to *𝑆*0 and *𝑆*2, both 0 or both given as output. When bits are dissimilar, depending upon input values 1 are given as output. In this process, the probabilities of *𝑝𝑧𝑒* and *𝑝𝑧* are

kept unchanged.



**/ig. 9.** The error detector circuit..



**/ig. 10.** System-Level representation of Error Correction mechanism.

* + 1. *Subtractor*

The difference between the error value *𝑝𝑧𝑒* and the actual value *𝑝𝑧*

absolute subtraction i.e, *𝑝𝑑𝑖𝑓 𝑓* = *𝑝𝑧𝑠* − *𝑝𝑧𝑒𝑠* , when *𝑝𝑧𝑠* and *𝑝𝑧𝑒𝑠* are is calculated to check the amount of deviation. The XOR gate performs

| |

positively correlated [[20](#_bookmark73)] which is done using a synchronizer.

* + 1. *Squarer*

Multiplication of two uncorrelated SNs is performed by an AND gate, but fails to implement the squaring operation [[20](#_bookmark73)] when the same

input sequence is given. When *𝑝𝑑𝑖𝑓 𝑓* is squared to obtain the MSE, it is

necessary to minimize the correlation between SNs. A shuffle buffer

circuit is used [[25](#_bookmark78)] to reduce the correlation between inputs to obtain the accurate squaring operation. It includes a multiplexer, three D Flip- flops and a Random Number Generator to generate numbers between

0 and 1 [[25](#_bookmark78)].

* + 1. *Comparator*

obtained *𝑀𝑆𝐸* with a very small value, say, *𝛿* (0.0001). When two The stochastic comparator, which is shown in [Fig.](#_bookmark34) [9](#_bookmark34) compares the

inputs the stochastic function [[20](#_bookmark73)] is given by *𝑝𝑜𝑢𝑡* = *𝑚𝑎𝑥*((*𝑀𝑆𝐸* −*𝛿*)*,* 0). correlated inputs are given to an AND gate with an inverter to one of its When *𝑀𝑆𝐸* which is representative of the error in computation is lesser than the error-tolerance of the circuit *𝛿*, a bitstream of 0’s is

obtained at the output of the comparator which indicates that the output is obtained satisfactorily.

# /ormalization of *𝑹𝒆𝑪𝒐* technique for stochastic circuits

The next step is to formally apply the proposed technique in a combinational circuit. For a two-input multilevel circuit as shown in

[Fig.](#_bookmark37) [11](#_bookmark37), we use circuit *𝑃 𝑇 𝑀* obtained as, *𝑀𝑐𝑘𝑡* = (*𝐼 ⊗ 𝑀𝑎𝑛𝑑 ⊗ 𝐼* )*.*(*𝐼 ⊗*

*𝐹*2 *⊗ 𝐼* )*.*(*𝑀𝑜𝑟 ⊗ 𝑀𝑜𝑟*)*.𝑀𝑎𝑛𝑑* .

**/ig. 11.** A 2 input multi-level circuit.

Consider *𝑝𝑥* = 0*.*3 and *𝑝𝑦* = 0*.*6. The error free output is *𝑝𝑧* = 0.18. If

*𝑝𝑒* = 0*.*125, the observed output, *𝑝𝑧𝑒* = 0*.*33. Using the proposed method,

*𝑀𝑆𝐸* = (0*.*066*𝑆𝐶𝐶* + 0*.*0388)2 which shows that *𝑀𝑆𝐸* can be reduced to zero for *𝑆𝐶𝐶* ≈ −0*.*58.

We will now discuss two distinct cases of fault correction in multi- input multi-level circuits.

* 1. *𝑅𝑒𝐶𝑜 I: Error minimization for Multiple-input Single output (MISO) stochastic logic circuits*

Let us consider correlation-sensitive blocks interconnected in a fash- ion as shown in [Fig.](#_bookmark38) [12](#_bookmark38). The quantification of SCC is only available for two signals in literature, so we consider two input gates in the circuit

model. The block diagram consists of *𝑖* = 1*,* 2*,* … *, 𝑛* levels and each

level consists of multiple two input gates. Consider that one or more

an increased *𝑀𝑆𝐸* at the output. The *𝑀𝑆𝐸* is minimized by selecting gates at different levels are subject to transient faults which result in

suitable candidates for *ReCo* analysis using the proposed Algorithm 2 which is discussed below.

**Definition 2.** An observed error at the primary output(s) driven by one or more gates, can be minimized by suitably injecting correlation at the prioritized input gates defined by one or more faulty gates in the error path P.

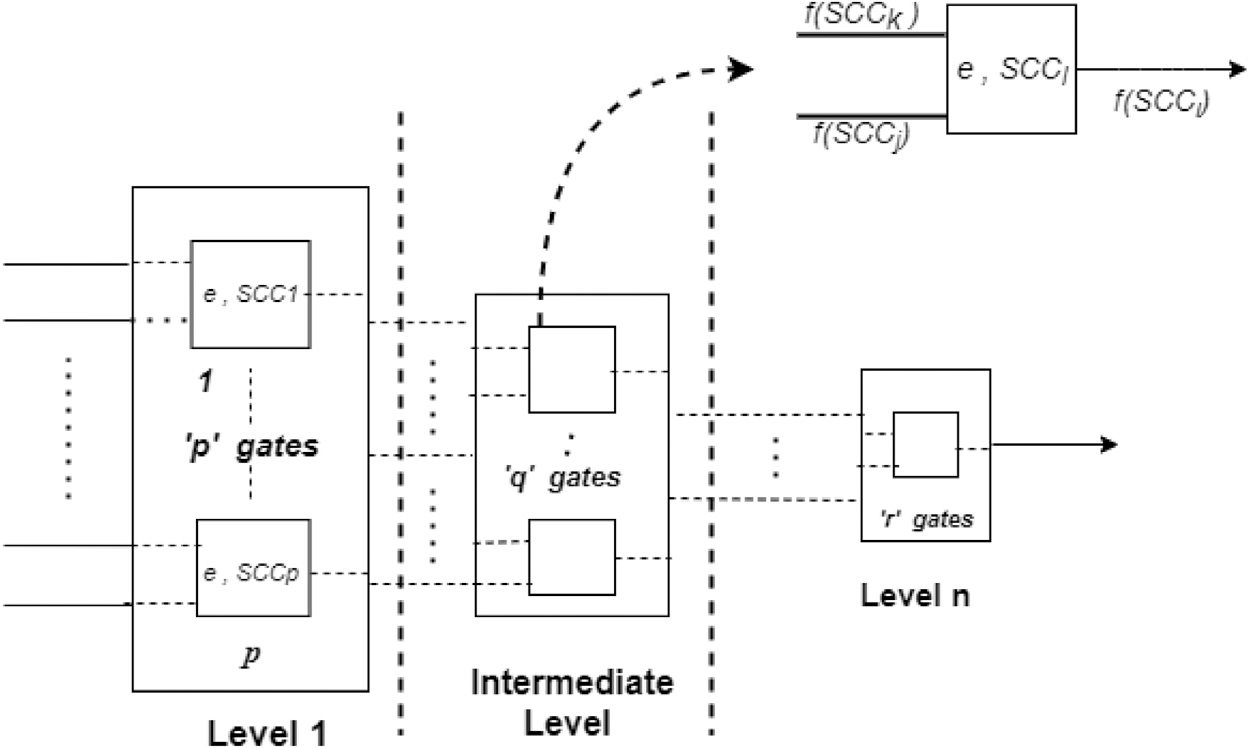
Assume that the correlation between input pairs for gates at level 1 as *𝑆𝐶𝐶*11, *𝑆𝐶𝐶*12, *𝑆𝐶𝐶*13,.., *𝑆𝐶𝐶*1*𝑝*, where *𝑝* is the number of 2-input gates at level 1. We can represent the output of each gate in level 1 in terms of *𝑆𝐶𝐶*. Thus, *𝑝𝑧*11 = *𝑓* (*𝑆𝐶𝐶*11), *𝑝𝑧*12 = *𝑓* (*𝑆𝐶𝐶*12), *𝑝𝑧*13 =

*𝑓* (*𝑆𝐶𝐶*13), . . . , *𝑝𝑧*1*𝑝* = *𝑓* (*𝑆𝐶𝐶*1*𝑝*) become functions of the corresponding

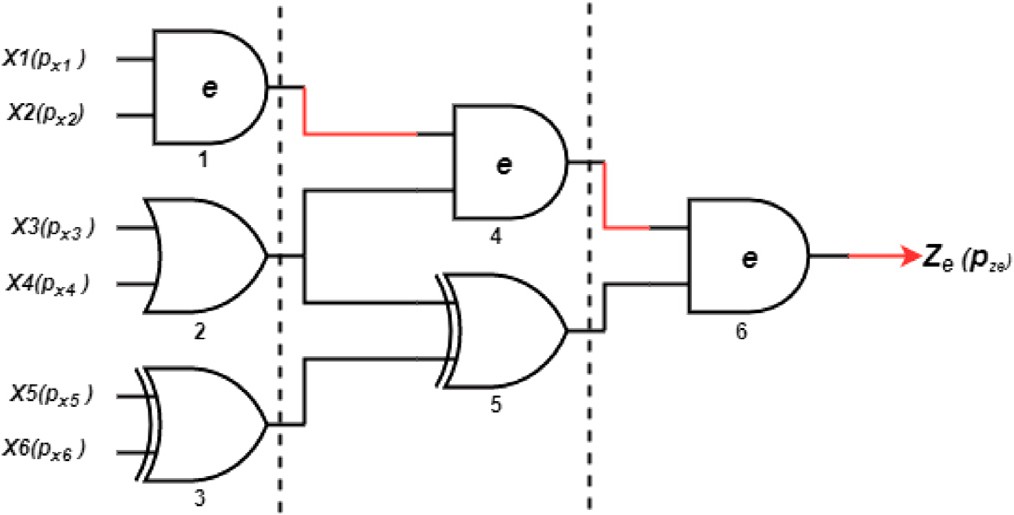
*𝑆𝐶𝐶*. These outputs are again inputs to certain gates in the next level

thus, in turn a function of *𝑆𝐶𝐶* of primary input pairs. of the circuit. The output from any gate in the intermediate level, is

Let the *𝑖*th intermediate level consists of *𝑞* interconnected gates. The output *𝑍𝑖*1 from gate 1 in *𝑖*th level is *𝑝𝑍𝑖*1 = *𝑓* (*𝑝𝑍*(*𝑖*−1)*𝑘 , 𝑝𝑍*(*𝑖*−1)*𝑗* ) where



**/ig. 12.** Block Diagram representation of Multi-level MISO SLC.

*𝑍*(*𝑖*−1)*𝑘* and *𝑍*(*𝑖*−1)*𝑗* are the outputs from *𝑘*th and *𝑗*th gate at the (*𝑖* − 1)*𝑡ℎ*

level. Thus, *𝑝𝑍*

*𝑖*1

= *𝑓* (*𝑝𝑍*(*𝑖*−2)*𝑙*

*, 𝑝𝑍*

(*𝑖*−2)*𝑚*

*, 𝑝𝑍*(*𝑖*−2)*𝑞*

*, 𝑝𝑍*

(*𝑖*−2)*𝑟*

), where *𝑝𝑍*

(*𝑖*−2)*𝑙*

and

*𝑝𝑍*(*𝑖*−2)*𝑚*

are the outputs of *𝑙*th and *𝑚*th gate of (*𝑖*−2)*𝑡ℎ* level that are inputs

to *𝑘*th gate in (*𝑖* − 1)*𝑡ℎ* level and *𝑝𝑍*(*𝑖*−2)*𝑞* and *𝑝𝑍*(*𝑖*−2)*𝑟* are the outputs from

*𝑞*th and *𝑟*th gate of (*𝑖*−2)*𝑡ℎ* level that are connected to *𝑗*th gate in (*𝑖*−1)*𝑡ℎ*

inputs, such that *𝑝𝑍𝑖*1 = *𝑓* (*𝑆𝐶𝐶*(*𝑖*−2)*𝑙, 𝑆𝐶𝐶*(*𝑖*−2)*𝑚, 𝑆𝐶𝐶*(*𝑖*−2)*𝑞 , 𝑆𝐶𝐶*(*𝑖*−2)*𝑟*). level. These outputs are again functions of SCCs of their corresponding

Thus, tracing the interconnected sub-networks, *𝑝𝑍*

*𝑖*1

is written as, *𝑝𝑍*

*𝑖*1 =

*𝑓* (*𝑆𝐶𝐶*11, *𝑆𝐶𝐶*12, *𝑆𝐶𝐶*13,.., *𝑆𝐶𝐶*1*𝑝*) of traced inputs at the primary

level.

initial assumption of *𝑆𝐶𝐶* at the inputs, that effect the succeeding Let transient faults at a certain level contribute to a shift in the

mediate correlation *𝑆𝐶𝐶𝑖*−1 present between *𝑍*(*𝑖*−1)*𝑘* and *𝑍*(*𝑖*−1)*𝑗* if the levels leading to erroneous results. We take into account the inter-

**/ig. 13.** A sample MISO circuit showing gates *𝐺*1*, 𝐺*4*, 𝐺*6 faulty.

circuit is non-faulty. Let, *𝑍*′

(*𝑖*−1)*𝑘*

and *𝑍*

′ (*𝑖*−1)*𝑗*

are the modified output

values on account of errors from *𝑘*th and *𝑗*th faulty gates at level

(*𝑖* − 1). These result in a change in the number of 1’s present in the

1. ***Determine input SLEs corresponding to faulty output node:***

bitstream. Let, probability of *𝑍*(*𝑖*−1)*𝑘* is changed from *𝑛*(*𝑖*−1)*𝑘* to *𝑛*′ and

*𝑍* is changed from *𝑛*

to *𝑛*′

. Eventually, *𝑆𝐶𝐶*

(*𝑖*−1)*𝑘*

Input gates that are connected to the faulty output node,

(*𝑖*−1)*𝑗*

′

(*𝑖*−1)*𝑗*

(*𝑖*−1)*𝑗*

*𝑖*1 is modified

denoted by ***IsConnected***(), are selected and are stored in

to *𝑆𝐶𝐶𝑖*1 which modifies the probability of *𝑍𝑖*1 from *𝑝𝑧*(*𝑖*−1)*𝑗* to *𝑝𝑧*′ −1)*𝑗* .

an array *𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*.

′ ′ ′

(*𝑖*

Thus, *𝑝* ′

*𝑧*

(*𝑖*−1)*𝑗*

= (1 + *𝑆𝐶𝐶*(*𝑖*−1)*𝑗* )*𝑝𝑧*′0(*𝑖*−1)*𝑗* − *𝑆𝐶𝐶*(*𝑖*−1)*𝑗 𝑝𝑧*′(+1)(*𝑖*−1)*𝑗* = *𝑛*(*𝑖*−1)*𝑗* ∕*𝑛*.

′

1. ***Register SLEs based on priority:*** The gates based on their

Similarly, *𝑝𝑧*′

can be written as *𝑝𝑧*′

= (1 + *𝑆𝐶𝐶*(*𝑖*−1)*𝑘*)*𝑝𝑧*′0(*𝑖*−1)*𝑘* −

priority values from left (highest) to right (lowest);

*𝑆𝐶𝐶*′

(*𝑖*−1)*𝑘*

*𝑝*

= *𝑛*′

∕*𝑛*.

(*𝑖*−1)*𝑘*

{*𝑋𝑂𝑅, 𝐴𝑁𝐷, 𝑂𝑅*} are sorted using ***𝑷 𝒓𝒊𝒐𝒓𝒊𝒕𝒚𝑺𝒐𝒓𝒕***() and are

(*𝑖*−1)*𝑘 𝑧*′(+1)(*𝑖*−1)*𝑘* (*𝑖*−1)*𝑘*

It is observed that the effect of transient error is reflected in an

of *𝑀𝑆𝐸* on *𝑆𝐶𝐶*. Thus, a suitable technique can be investigated that overall change in the probability of SNs. This suggests the dependence

adapts to the change in the assumption of correlation and tries to minimize the observed error at the output by using *ReCo* method.

It relocates *𝑆𝐶𝐶* at several levels to counterbalance the change in

initial assumption of *𝑆𝐶𝐶* due to transient faults. We trace faults and identify faulty gates in the circuit and remodel *𝑆𝐶𝐶𝑠* by modifying input vectors of the primary inputs of the sub network in level 1.

In noisy operating conditions, errors in output are assumed to be primarily contributed by one or more faulty gates in the circuit. For a multilevel MISO circuit shown in [Fig.](#_bookmark39) [13](#_bookmark39), the number of SLEs that undergo *ReCo* correction primarily depends on the number of faulty gates and the probability of error. Thus, it is important to identify faulty paths in the circuit. We generalize the procedure for fault correction in

*𝑀𝐼𝑆𝑂* circuits as follows:

* Check if the *𝑀𝑆𝐸* of the circuit is within the tolerable limit

*𝛿*(*<*= 10−3) or not. If not,

i ***Determine Faulty gates***: Generate *T* input test vectors *n* times and the output at each gate is observed. The error rate is estimated by the number of faulty outputs for *n*

inputs using ***FaultEvaluation***(*𝐶𝐼 𝑅𝐶𝑈 𝐼 𝑇* ).

stored in an array *𝑆*\_*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*.

and alter *𝑆𝐶𝐶* using ***𝑹𝒆𝑪𝒐***() (Algorithm 1). Calculate if, iv ***Selection of SLEs for ReCo:*** Pop gate from the priority list

*𝑀𝑆𝐸* ≤ *𝛿*, return corresponding *𝑆𝐶𝐶𝑖*. If not, invoke next

SLE from the sorted priority list.

From the previous discussion, it can be inferred that the XOR

the overall *𝑀𝑆𝐸* substantially compared to other correlation sensitive gate is most susceptible to changes in correlation and can reduce

SLEs. So XOR gate line up the highest in the correlation-sensitivity

tions from the list ***L\_MSE[]***. Suppose, ***𝑷 𝒓𝒊𝒐𝒓𝒊𝒕𝒚𝑺𝒐𝒓𝒕***() list consists of index. If a single gate does not suffice then we proceed for combina-

{*𝑋𝑂𝑅, 𝐴𝑁𝐷*1*, 𝐴𝑁𝐷*2} and *𝐴𝑁𝐷*2 generates less *𝑀𝑆𝐸* compared to

*𝐴𝑁𝐷*1, then we combine *𝑋𝑂𝑅* and *𝐴𝑁𝐷*2 to find minimum MSE. This

condition is often guided by the position of the gate(s) in the circuit.

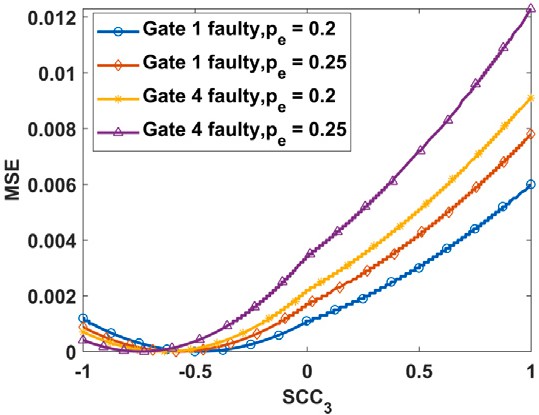
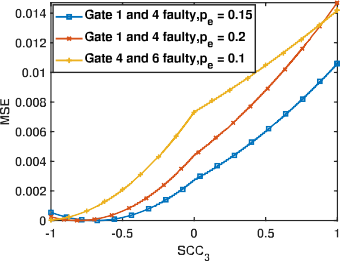
The whole analysis is carried out to improve the accuracy and reduce the number of correlators in the circuit.

It is observed that the *𝑀𝑆𝐸* is proportional to the number of faulty

gates and the error rate *𝑝𝑒*. For any error observed at the output, the

error can be due to transient error at the gate itself or due to the error

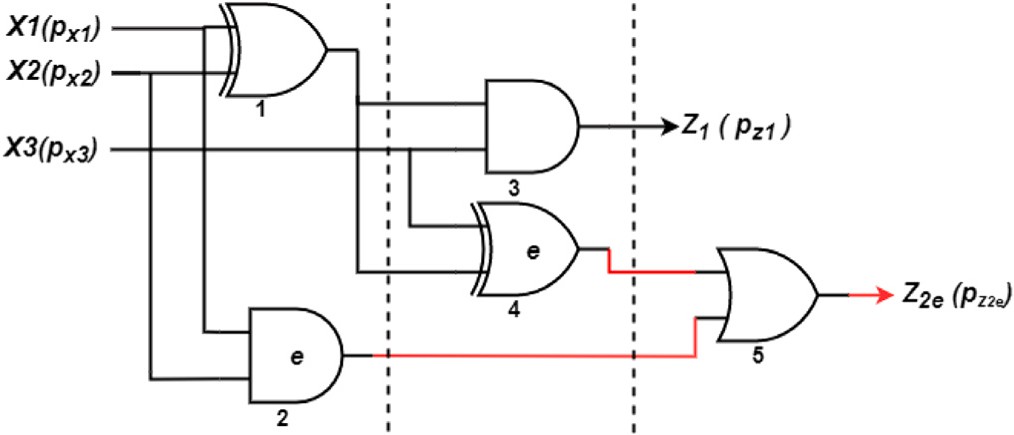
being propagated from the previous stage or the both. We consider different cases of fault propagation in [Fig.](#_bookmark39) [13](#_bookmark39) where the path P is indicated in red colour.





**/ig. 14.** Graphs showing different values of induced correlation to achieve minimum *𝑀𝑆𝐸* for different gates faulty at different error rates in [Fig.](#_bookmark39) [13](#_bookmark39).



**/ig. 15.** A sample MIMO circuit showing gates *𝐺*2*, 𝐺*4 faulty.

It is observed from [Fig.](#_bookmark40) [14](#_bookmark40)(a),(b) that for a single faulty gate, *𝐺*1/*𝐺*4 with *𝑝𝑒* ≤ 0*.*20, single gate *ReCo* analysis, *𝑆𝐶𝐶*1 or *𝑆𝐶𝐶*3 will suffice to minimize MSE. For multiple gates faulty, as for example, *𝐺*1*, 𝐺*4 faulty with *𝑝𝑒* ≤ 0*.*125 same logic can comply. Thus, in both cases, the error

of SLE (excluding OR). But *𝐺*1*, 𝐺*4 faulty at *𝑝𝑒* = 0*.*2 fault tolerance can can be minimized without being thoroughly guided by the priority-rule be achieved using prioritized SLE (*𝑆𝐶𝐶*3) only as shown in [Fig.](#_bookmark40) [14](#_bookmark40)(c)

(red).

of prioritized SLE is obligatory. Thus, *𝐺*1*, 𝐺*4*, 𝐺*6 faulty at *𝑝𝑒* ≤ 0*.*125, For a larger number of gates faulty even at low error rates, treatment

*𝑀𝑆𝐸* can be minimized using single gate *ReCo* (*𝑆𝐶𝐶*3). But the same

with *𝑝𝑒* = 0*.*20 we invoke dual gate *ReCo* (*𝑆𝐶𝐶*3*, 𝑆𝐶𝐶*1) as shown in [Fig.](#_bookmark47) [18](#_bookmark47)(d). When *𝐺*4 and *𝐺*6 are faulty with *𝑝𝑒* ≤ 0*.*25 we invoke dual *ReCo* (*𝑆𝐶𝐶*3*, 𝑆𝐶𝐶*1) as shown in [Fig.](#_bookmark47) [18](#_bookmark47)(a)–(c). It is observed that there is a finite error of 0.01 when *𝐺*1*, 𝐺*4*, 𝐺*6 are faulty at a rate of 0.25

even after dual *ReCo* analysis of prioritized SLEs. [Table](#_bookmark44) [1](#_bookmark44) is given to comprehend the nature of the analysis and the results obtained in the proposed work.

It is observed that the possibility of error reduction is motivated by several factors; the nature and number of faulty gates as well as the arrangement of gates in the circuit. As AND gate is the most susceptible to soft errors, an AND gate in place of the XOR gate would contribute to larger MSE. These factors coupled with the probability of error play a pivotal role in determining the circuits’ resilience towards soft errors.

There is also a slight dependence on input probability values i.e *𝑝𝑥*

and *𝑝𝑦* as indicated in [Fig.](#_bookmark22) [7](#_bookmark22)(a)–(c). The worst-case time complexity of Algorithm 2 is calculated as *𝑂*(*𝐺*) + *𝑂*(*𝐹* ) + *𝑂*{*𝐼 𝑙𝑜𝑔*(*𝐼* )} + *𝑂*(*𝐼* + *𝐼* 2), where, *𝐺* is the number of gates in the circuit and *𝐼* is the number of

input gates connected to these faulty gates.

* + 1. *ReCo II: Error minimization for Multi-input and Multi-output (MIMO) SLC*

down *𝑀𝑆𝐸* close to 0. We will see that in particular situations that can We have argued that *ReCo* analysis of the prioritized SLEs can bring

deviate from this initial assumption. When non-faulty gates converge to a different output node, *ReCo* analysis of primary SLEs may give

undesired results. The condition can be best described with the help of a Multiple-input–Multiple-Output circuit shown in [Fig.](#_bookmark41) [15](#_bookmark41). The circuit

has two distinct outputs *𝑍*1 and *𝑍*2 with probabilities *𝑝𝑧*1 and *𝑝𝑧*2 .

Consider two faulty gates *𝐺*2 and *𝐺*4 that converge to *𝑝𝑧*2 i.e., the output of gate 5 and the output is modified to *𝑝𝑧𝑒*2 . We assume that there are no faulty gates in the path that converges to *𝑝𝑧*1 . One way of suppressing

the propagation of faulty results to the non-faulty output node is to

perform *ReCo* analysis at the inputs of faulty gates only. This avoids analysis of primary SLEs that are explicitly connected to non-faulty outputs.

It is interesting to note that for *𝐺*2*, 𝐺*4 faulty with an error rate

≤ 0*.*2, the *𝑀𝑆𝐸* at the output *𝑍*2 can be reduced to 0 by doing *ReCo*

analysis at either of the faulty gates, *𝑆𝐶𝐶*2 or *𝑆𝐶𝐶*4, as shown in

of the precedence rule as shown in [Fig.](#_bookmark43) [16](#_bookmark43)(b). For *𝑝𝑒 >* 0*.*2, however, [Fig.](#_bookmark43) [16](#_bookmark43)(a). We may perform *ReCo* at any faulty gate location, regardless

*𝑀𝑆𝐸* can only be reduced to 0 using *𝐴𝑙𝑔𝑜𝑟𝑖𝑡ℎ𝑚*3. Modelling correlation between inputs of gate *𝐺*4 (which is the priority in this case), can reduce *𝑀𝑆𝐸* to 0 with a soft error margin of 0*.*3. At *𝑝𝑒* = 0*.*325, we invoke dual *𝑅𝑒𝐶𝑜* (*𝑆𝐶𝐶*4*, 𝑆𝐶𝐶*2) to observe the error free behaviour as

shown in [Fig.](#_bookmark47) [18](#_bookmark47)(e). *ReCo* is performed specifically at the faulty gate site to prevent the generation of erroneous output at a node that is

preceded by non-erroneous output nodes. The modelling of *𝑆𝐶𝐶*s at the

primary gates *𝐺*1 is avoided without affecting the output logic level at

*𝑍*1. This is contrary to the selection criteria in *𝐴𝑙𝑔𝑜𝑟𝑖𝑡ℎ𝑚*2. It attempts

(*𝑙* = 1). It is observed that *𝑀𝑆𝐸* can be reduced to zero even for very to minimize the number of correlator circuits for larger error rates 0.30 high error rates, i.e., *𝑝𝑒* = 0*.*325 using dual *ReCo* analysis. The results

Algorithm 3 is calculated as *𝑂*(*𝐺*) + *𝑂*{*𝐹 𝑙𝑜𝑔*(*𝐹* )} + *𝑂*(*𝐹* + *𝐹* 2), where, *𝐺* is of analysis are recorded in [Table](#_bookmark45) [2](#_bookmark45). The worst-case time complexity of the number of gates in the circuit and *𝐹* is the number of faulty gates.

# Experimental results and discussion

We can now identify the key factors on which the whole analysis

well as the number (*𝑙*) of *ReCo* blocks, depend on several underlying is hinged upon. The magnitude and polarity of induced correlation, as

circuit. As transient error increases, the *𝑀𝑆𝐸* increases exponentially factors. The predominant factor is the amount of transient error in the (see graphs in [Fig.](#_bookmark4) [1](#_bookmark4)(a)). With higher *𝑀𝑆𝐸* the value of *𝑙* tends to be larger. For *𝐺*1*, 𝐺*4*, 𝐺*6 faulty at *𝑝𝑒* ≤ 0*.*125 error can be reduced to 0 using a single *ReCo* block (*𝑆𝐶𝐶*3). The *𝑝𝑒* ≥ 0*.*20 error, however, can be reduced to 0 with *𝑙* = 2 as shown in [Table](#_bookmark44) [1](#_bookmark44). This is because *𝑀𝑆𝐸*

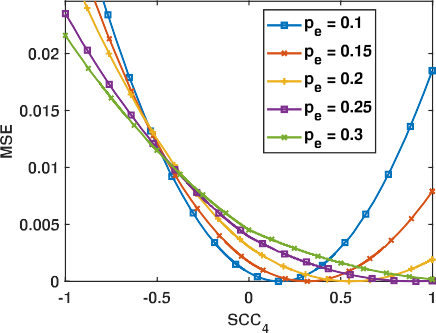
is higher in the second case.

creases, the value of *𝑙* increases to account for the increased MSE. When It is identified that as the number of faulty gates in the circuit in-

*𝐺*1 faulty and *𝐺*1*, 𝐺*4*, 𝐺*6 faulty with the same error rate, i.e., 0.20, the error-free output is obtained respectively with *𝑙* = 1 (*𝑆𝐶𝐶*3∕*𝑆𝐶𝐶*1) and *𝑙* = 2 (*𝑆𝐶𝐶*3*, 𝑆𝐶𝐶*1). In addition, the proliferation of the *𝑆𝐿𝐸*

reducing the error to zero. Therefore, if AND gate *𝐺*1 is replaced with with the highest priority in the input panel increases the possibility of

**/ig. 16.** Graphs showing different values of induced correlation to achieve minimum *𝑀𝑆𝐸* for *𝐺*2*, 𝐺*4 faulty at different error rates in [Fig.](#_bookmark41) [15](#_bookmark41).

**Table 1**

Comparison of *𝑀𝑆𝐸* with and without *ReCo* of the circuit in [Fig.](#_bookmark39) [13](#_bookmark39). *𝑀𝑆𝐸𝑚𝑖𝑛* represents minimum MSE.

|  |  |  |
| --- | --- | --- |
|  | Faulty gate(s) | MSE without ReCo |
| Gate 1 | 0.00042 |
| Error = 0.125 | Gate 4 | 0.00085 |
|  | Gate 1, 4 | 0.002 |
|  | Gate 1,4, 6 | 0.016 |
|  | Gate 1 | 0.0011 |
| Error = 0.20 | Gate 4 | 0.0022 |
|  | Gate 1, 4 | 0.0044 |
|  | Gate 1,4, 6 | 0.0375 |
|  | Gate 1 | 0.0017 |
| Error = 0.25 | Gate 4 | 0.0034 |
|  | Gate 1, 4 | 0.049 |
|  | Gate 1,4, 6 | 0.1317 |

with ReCo (Single/Dual gate)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *𝑆𝐶𝐶*1 | *𝑆𝐶𝐶*3 | *𝑆𝐶𝐶*1 , *𝑆𝐶𝐶*3 | *𝑆𝐶𝐶*2 , *𝑆𝐶𝐶*3 |
| *𝑆𝐶𝐶𝑖* | −0.36 | −0.61 | −0.19, −0.16 | −0.26, 0.33 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −0.51 | −0.45 | −0.27, −0.23 | −0.34, −0.38 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.61 | −0.4471, −0.4001 | −0.51, −0.82 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −1 | −0.9,−0.9 | −1,0 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.0095 | 0 | 0 | 0.001 |
| *𝑆𝐶𝐶𝑖* | −0.717 | −0.4896 | −0.36,−0.27 | −0.5,−0.49 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.6385 | −0.45,−0.41 | −1,−0.1 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.8078 | −0.546, −0.7 | −0.524, −0.68 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.0015 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −1 | −1,−1 | −1, −0.15 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.0314 | 0.0108 | 0 | 0.01 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.5813 | −0.44, −0.38 | −0.47,−0.83 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.0114 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.743 | −52,−0.55 | −1,−0.1 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.0004 | 0 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −1 | −0.8987 | −1,−1 | −1, −0.09 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.036 | 0 | 0 | 0.0045 |
| *𝑆𝐶𝐶𝑖* | −1 | −1 | −1,−1 | −1,0 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.041 | 0.023 | 0.01 | 0.02 |

**Table 2**

Comparison of *𝑀𝑆𝐸* with and without *ReCo* of the circuit in [Fig.](#_bookmark41) [15](#_bookmark41). *𝑀𝑆𝐸𝑚𝑖𝑛* represents minimum MSE.

|  |  |  |
| --- | --- | --- |
|  | Faulty gates | MSE  without ReCo |
| Error = 0.25 | Gates 2,4 | 0.003 |
| Error= 0.30 | Gates 2,4 | 0.004 |

with ReCo (single or dual gate(s))

|  |  |  |  |
| --- | --- | --- | --- |
|  | *𝑆𝐶𝐶*2 | *𝑆𝐶𝐶*4 | *𝑆𝐶𝐶*2 *, 𝑆𝐶𝐶*4 |
| *𝑆𝐶𝐶𝑖* | −0.9 | −0.33 | −0.2,−0.4 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.001 | 0 | 0 |
| *𝑆𝐶𝐶𝑖* | −0.93 | −0.87 | −0.15,−1 |
| *𝑀𝑆𝐸𝑚𝑖𝑛* | 0.002 | 0 | 0 |

an XOR gate, *𝑀𝑆𝐸* can be reduced to 0, even if, *𝐺*1*, 𝐺*4*, 𝐺*6 are faulty at *𝑝𝑒* = 0*.*25.

The observed *𝑀𝑆𝐸* is greatly dependent on the nature of faulty

gates. If AND gate *𝐺*4 is replaced in the circuit with an XOR gate, the overall *𝑀𝑆𝐸* is reduced from 0.0375 to 0.02 when *𝐺*1*, 𝐺*4*, 𝐺*6 are faulty at *𝑝𝑒* = 0*.*20. This is because XOR is least susceptible to transient

on the overall *𝑀𝑆𝐸*. A faulty gate distant from the periphery of the errors. The location of faulty gates within the circuit has also an impact

for example, *𝐺*1 faulty at an error rate *𝑝𝑒* = 0*.*2 gives *𝑀𝑆𝐸* as 0.0011, primary input and closer to the output contributes to a larger MSE. As whereas *𝐺*6 faulty at the same error rate gives larger *𝑀𝑆𝐸* (0.0237).

This is because *𝐺*6 is distant from the primary input side than *𝐺*1. It

is also implicit from [Fig.](#_bookmark4) [1](#_bookmark4)(c), (d) and [Fig.](#_bookmark9) [2](#_bookmark9)(c), (d) that the initial

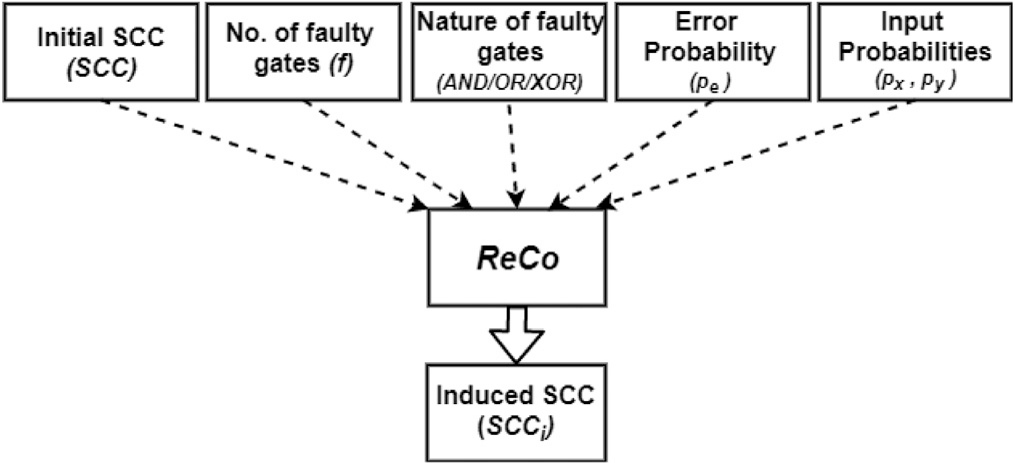
assumption of SCC also plays a significant role in determining the exact operating point of SCC for a given circuit. In the current experimental setup as in [Fig.](#_bookmark41) [15](#_bookmark41) the error rates up to 0.3 can be handled accurately and this is determined by the number of faulty gates in the circuit

(≤ 2) and location of faulty gates (closer to the input side) as shown

in [Fig.](#_bookmark48) [18(e)](#_bookmark48). A block diagram illustrating the interdependence of these

parameters is shown in [Fig.](#_bookmark46) [17](#_bookmark46).

For complex circuits with multiple faulty gates, we have introduced a priority-based selection scheme of SLEs showing promising results.



**/ig. 17.** Factors influencing the polarity and the magnitude of the induced SCC.

In our investigation, the XOR gate is a prime component because of its high correlation sensitivity. Assuming a priority-based strategy, the bare minimum number of *ReCo* blocks needed to achieve a minimum

*𝑀𝑆𝐸* is found as *𝑙* = 1. The only exception is *𝐺*1*, 𝐺*4*, 𝐺*6 faulty at *𝑝𝑒* =

0*.*25, which requires two *ReCo* blocks to achieve minimum *𝑀𝑆𝐸*. The graphs of [Fig.](#_bookmark49) [19](#_bookmark49) are obtained with *𝐺*1*, 𝐺*4*, 𝐺*6 faulty at different error

rates. The deviation in output from the actual value (without error) using the proposed scheme is much less compared to the non-priority- based approach. It has been found that this method can deal with high error rates while using less hardware. Furthermore, the desired value can be achieved with fewer iterations when using a priority-based approach. The deviation graph with the priority-based (red) approach is obtained with one *ReCo* block whereas the deviation with non-priority based approach (gray) is obtained with two *ReCo* blocks to model the output. The blue line represents to the observed error (without *ReCo*). This demonstrates the efficacy of the proposed priority-based approach in terms of hardware design. The efficacy of the proposed method relies heavily on the presence of correlation sensitive logic blocks in the circuit. If there are no correlation sensitive logic blocks in the circuit the method fails drastically. Also, care needs must be taken while placing the *ReCo* blocks at specific targets without which errors could be propagated to non-erroneous output node.

As there is no standard benchmark stochastic circuit available in lit-

erature, we have implemented a complex matrix multiplication circuit using stochastic logic blocks under the assumption that the erroneous behaviour of the circuit is due to the cumulative error caused by the

12 constants and some of the gates being noisy due to the presence

of transient error. A comparison of *𝑀𝑆𝐸* using CEASE method in

the presence of only constant error [[15](#_bookmark68)] and the proposed method in

presence of both the constant error and transient error is shown in [Table](#_bookmark53) [4](#_bookmark53). Due to presence of both the errors in the circuit the original

*𝑀𝑆𝐸* was bit higher in our case, but after correction using *𝑅𝑒𝐶𝑜*

method the *𝑀𝑆𝐸* can be lowered significantly, which shows that the

proposed method can potentially combat the effect of the duo.

In [Table](#_bookmark50) [3](#_bookmark50) estimates the number of *𝑅𝑒𝐶𝑜* blocks needed to minimize the *𝑀𝑆𝐸* at a given error rate to indicate the efficiency of the proposed

The cells marked in grey show the variations in *𝑀𝑆𝐸* at a particular method in making a trade-off between error metrics and overheads. error rate. In cases, where it is not possible to achieve the lowest *𝑀𝑆𝐸*

output. Increase in the number of *𝑅𝑒𝐶𝑜* blocks in the circuit increases using a single block, a second block is inserted to observe the error-free

hardware overhead, which can be minimized by using a priority-based approach. It is observed that even for very high degree of error rates

with multiple gates faulty, it is possible to bring down the *𝑀𝑆𝐸* within

a limit using only two *𝑅𝑒𝐶𝑜* blocks in the circuit.

# Case study: Contrast enhancement in images

In order to demonstrate the practicality and efficacy, we have implemented the proposed technique for contrast enhancement [[31](#_bookmark84)] to images of a standard publicly available contrast enhancement dataset,

**Algorithm 2:** *ReCo analysis for MISO Circuits*

1: **Input:** *𝐶𝐼𝑅𝐶𝑈 𝐼 𝑇 // Circuit with n number of inputs.*

**Output:** *𝑀𝑆𝐸, 𝑆𝐶𝐶𝑖*

2: **Variable Initialization:**

*𝑝*\_*𝑎𝑟𝑟*[ ] = {*𝑋𝑂𝑅, 𝑂𝑅, 𝐴𝑁𝐷*} *//Priority Sequence*

*𝐹* \_*𝑔𝑎𝑡𝑒*[ ] = {0} // number of faulty gates

*𝐼* \_*𝑔𝑎𝑡𝑒*[ ] = {*𝑛𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑖𝑛𝑝𝑢𝑡 𝑔𝑎𝑡𝑒𝑠*}

*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*[ ] = {0} *// Input gates* ∈ *𝐹 \_𝑔𝑎𝑡𝑒*[ ]

*𝑆*\_*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*[ ] = {0} *// Sorted 𝐹 𝐶\_𝐼 \_𝑔𝑎𝑡𝑒*[ ]

3: *𝐹* \_*𝑔𝑎𝑡𝑒*=***FaultEvaluation***(*𝐶𝐼 𝑅𝐶𝑈 𝐼 𝑇* )

*// Identify faulty gates in the circuit*

4: *𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*=***IsConnected***(*𝐼* \_*𝑔𝑎𝑡𝑒, 𝐹* \_*𝑔𝑎𝑡𝑒*)

5: *𝑆*\_*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*=***PrioritySort*** (*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*, *𝑝*\_*𝑎𝑟𝑟𝑟*)

6: **for** *𝑗* = 1 *to 𝑗 <*= ***maxElement*** *(𝑆\_𝐹 𝐶\_𝐼 \_𝑔𝑎𝑡𝑒*) **do**

[*𝐿*\_*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]] = ***ReCo***(*𝑆*\_*𝐹 𝐶*\_*𝐼* \_*𝑔𝑎𝑡𝑒*[*𝑗*])

**if** *(𝐿\_𝑀𝑆𝐸*[*𝑗*] *<*= *𝛿)* **then**

return *𝐿*\_*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]

**end if**

**end for**

7: [*𝑆*\_*𝐿𝑔𝑎𝑡𝑒, 𝐿*\_*𝑀𝑆𝐸*] = ***Sort*** (*𝐿*\_*𝑀𝑆𝐸*) //*Sort 𝑀𝑆𝐸 value along with their*

*gate number*

8: **for** *𝑗* = 1 *to 𝑗 <*= ***maxElement*** (*S\_FC\_I\_gate*) **do** [*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]] = *𝑁𝑒𝑤𝑅𝑒𝑐𝑜*(*𝑆*\_*𝐿*\_*𝑔𝑎𝑡𝑒, 𝑗* + 1) **if** *(𝑀𝑆𝐸*[*𝑗*] *<*= *𝛿)* **then**

return *𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]

**end if end for**

9: return argmin*𝑀𝑆𝐸*[*𝑗*] {*𝑀𝑆𝐸*[*𝑗*] , *𝑆𝐶𝐶𝑖* [*𝑗*]}

**Algorithm 3:** *ReCo analysis for MIMO Circuits* **Input:** *𝐶𝐼𝑅𝐶𝑈 𝐼 𝑇 // Circuit with n number of inputs* **Output:** *𝑀𝑆𝐸, 𝑆𝐶𝐶𝑖*

**Variable Initialization:**

*𝑝*\_*𝑎𝑟𝑟*[ ] = {*𝑋𝑂𝑅, 𝑂𝑅, 𝐴𝑁𝐷*} *//Priority Sequence*

*𝐹* \_*𝑔𝑎𝑡𝑒*[ ] = {0} // number of faulty gates

*𝑆*\_*𝐹* \_*𝑔𝑎𝑡𝑒*[ ] = {0} *// Sorted 𝐹 \_𝑔𝑎𝑡𝑒*[ ]

*𝐹* \_*𝑔𝑎𝑡𝑒*=***FaultEvaluation***(*𝐶𝐼 𝑅𝐶𝑈 𝐼 𝑇* )

*// Identify faulty gates in the circuit*

*𝑆*\_*𝐹* \_*𝑔𝑎𝑡𝑒*=***PrioritySort*** (*𝐹* \_*𝑔𝑎𝑡𝑒*, *𝑝*\_*𝑎𝑟𝑟𝑟*)

**for** *𝑗* = 1 *to 𝑗 <*= ***maxElement*** (*S\_F\_gate*) **do** [*𝐿*\_*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]] = ***ReCo***(*𝑆*\_*𝐹* \_*𝑔𝑎𝑡𝑒*[*𝑗*]) **if** *(𝐿\_𝑀𝑆𝐸*[*𝑗*] *<*= *𝛿)* **then**

return *𝐿*\_*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]

**end if end for**

[*𝑆*\_*𝐿𝑔𝑎𝑡𝑒, 𝐿*\_*𝑀𝑆𝐸*] = ***Sort*** (*𝐿*\_*𝑀𝑆𝐸*) //*Sort 𝑀𝑆𝐸 value along with their gate*

*number*

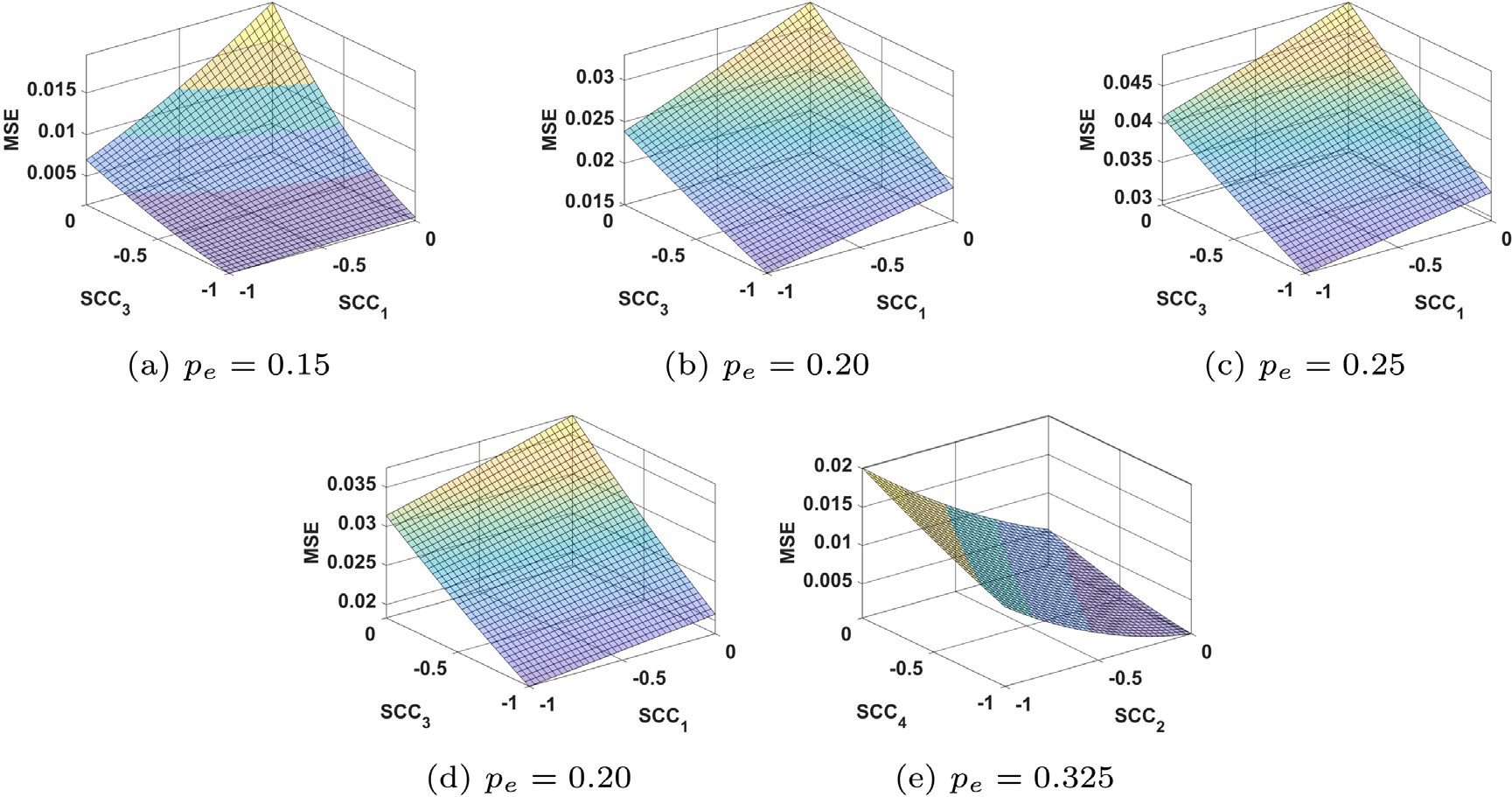
**for** *𝑗* = 1 *to 𝑗 <*= ***maxElement*** (*S\_F\_gate*) **do** [*𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]] = *𝑁𝑒𝑤𝑅𝑒𝑐𝑜*(*𝑆*\_*𝐿*\_*𝑔𝑎𝑡𝑒, 𝑗* + 1) **if** *(𝑀𝑆𝐸*[*𝑗*] *<*= *𝛿)* **then**

return *𝑀𝑆𝐸*[*𝑗*]*, 𝑆𝐶𝐶𝑖* [*𝑗*]

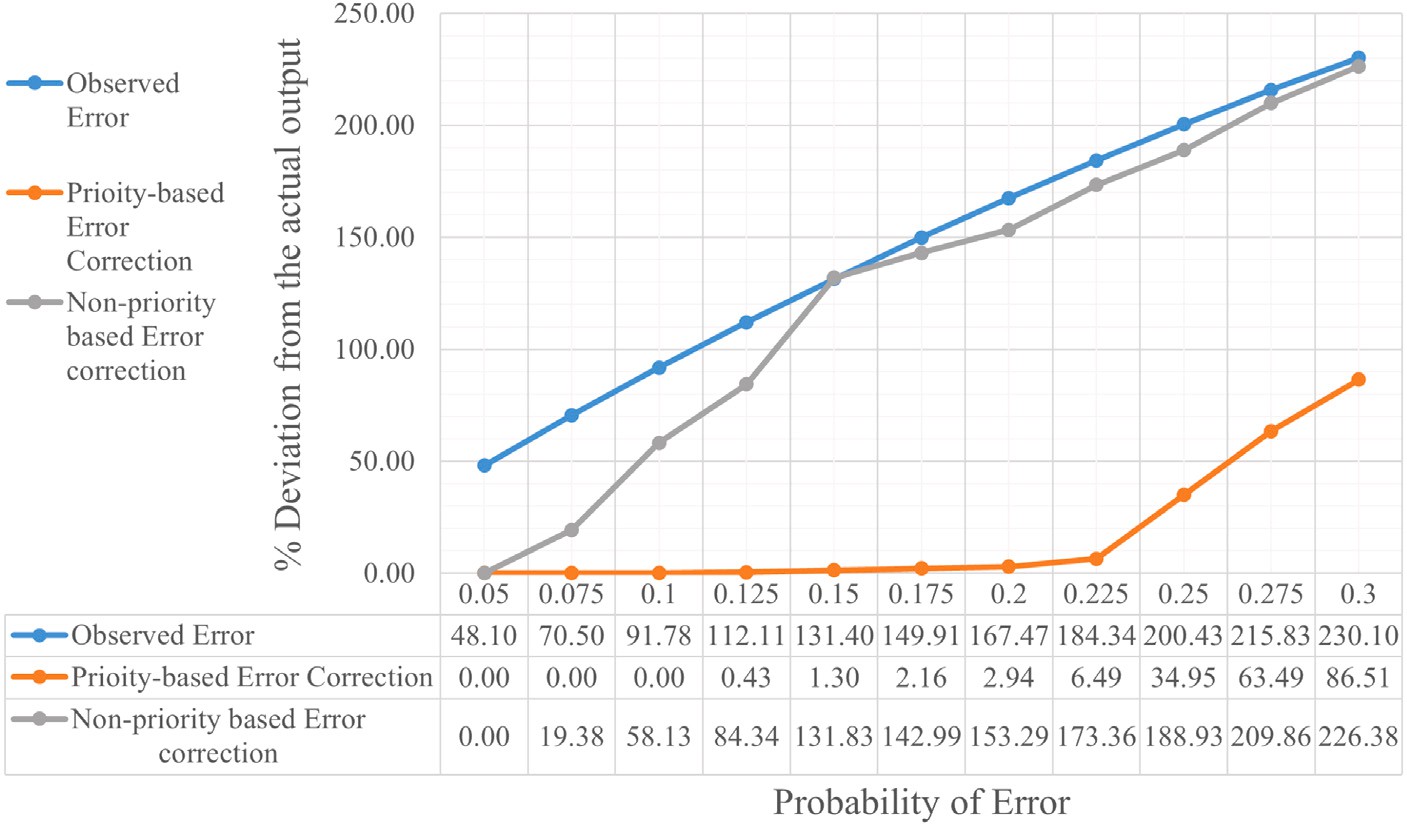
**end if end for**

return argmin*𝑀𝑆𝐸*[*𝑗*] {*𝑀𝑆𝐸*[*𝑗*] , *𝑆𝐶𝐶𝑖* [*𝑗*]}

CEED2016 [[30](#_bookmark83)] which contains 30 benchmark images. The output contrast enhanced images after introducing error and its subsequent revisal using the proposed methodology are shown in [Fig.](#_bookmark51) [20](#_bookmark51) using two randomly selected images from the CEED2016 dataset. Differ- ent evaluation metrics such as Multi Scale Structural Similarity Index (MS-SSIM) [[32](#_bookmark85)], Entropy [[33](#_bookmark86)], Peak signal-to-noise ratio (PSNR) [[34](#_bookmark87)] and Contrast-to-Noise Ratio (CNR) [[34](#_bookmark87)] are calculated on the entire CEED2016 dataset to measure the similarity between the enhanced images and the ground truth image using our methods. The results are shown in [Fig.](#_bookmark52) [21](#_bookmark52). It is observed that all the metrics of the enhanced images using the contrast stretching technique are improved after the correction of errors. A priority-based strategy is utilized for this



**/ig. 18.** Error minimization using dual ReCo analysis at different error rates when; (a), (b), (c) *𝐺*4*, 𝐺*6 faulty and (d) *𝐺*1*, 𝐺*4*, 𝐺*6 faulty in [Fig.](#_bookmark39) [13](#_bookmark39); (e) *𝐺*2*, 𝐺*4 faulty in [Fig.](#_bookmark41) [15](#_bookmark41).



**/ig. 19.** Deviation in output of two correction approaches from the actual value (without error) of the circuit in [Fig.](#_bookmark39) [13](#_bookmark39).

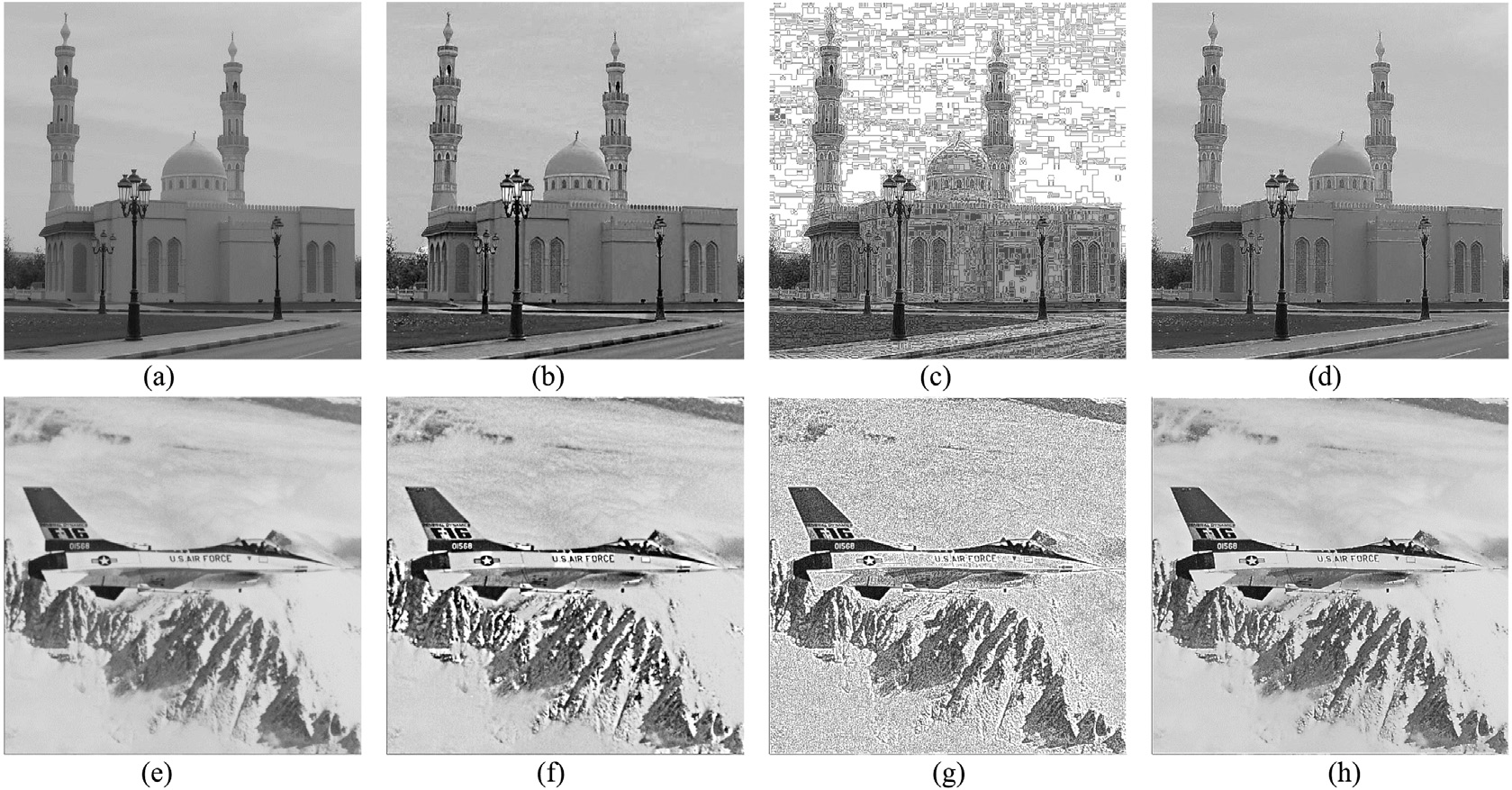
**Table 3**

Minimization of *𝑀𝑆𝐸* using varying number *𝑅𝑒𝐶𝑜* blocks at different error rates.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SL#** | zzzz **/ault**  z **Rate**  **/ault** zzzz  **Location** | *𝑝𝑒* = 0*.*1 | | *𝑝𝑒* = 0*.*2 | | *𝑝𝑒* = 0*.*3 | |
| **1** | **G1** | **#ReCo**  **blocks(l)** | **Min.**  **MSE** | **# ReCo**  **blocks(l)** | **Min.**  **MSE** | **# ReCo**  **blocks(l)** | **Min.**  **MSE** |
| l=1 | 0 | l=1 | 0 | l=1 | 0 |
| **2** | **G1,G4** | l=1 | 0 | l=1 | 0 | l=1 | 0.007 |
| l=2 | 0 |
| **3** | **G1,G4,G6** | l=1 | 0 | l=1 | 0.01 | l=1 | 0.05 |
| l=2 | 0 | l=2 | 0.027 |

purpose, where two SLEs are chosen to achieve an error-tolerant circuit behaviour. From [Fig.](#_bookmark52) [21](#_bookmark52)(a), it is observed that the MS-SSIM indices using the proposed *ReCo* method are considerably higher compared to the enhanced image with the error, indicating that the proposed

methodology provides faithful results even in transient error scenarios. The entropy of the different images after image enhancement are shown in [Fig.](#_bookmark52) [21](#_bookmark52)(b). The figure shows the entropy of the images (after correction) are close to the ground truth images compared to the images



**/ig. 20.** Samples images of CEED2016 dataset [[30](#_bookmark83)] before and after Contrast Stretching operations along with ground truth. (a,e) Original images, (b,f) Ground truth images, (c,g) Enhanced images with error, and (d,h) Corrected images using the proposed framework.





**/ig. 21.** Comparisons of different metrics calculated after Contrast Stretching operations on the images of CEED2016 dataset [[30](#_bookmark83)] with error and after the error correction using the proposed methodology.



**Table 4**

Performance comparison on Complex Matrix Multiplication using CEASE [[15](#_bookmark68)] and the proposed method.

Mean squared error Matrix Multiplication

CEASE

(Constant error)

ReCo

(Constant error + Transient error)

Original 8 × 10−3 10*.*75 × 10−3

After correction 2 × 10−3 0*.*44 × 10−3

obtained with error. The PSNR and CNR values of the images are also shown in [Fig.](#_bookmark52) [21](#_bookmark52)(c). The proposed method thus can be implemented with lower hardware cost in various image processing applications.

# Conclusion

Recent applications of stochastic computing involve noisy oper- ating conditions leading to incorrect results at times. The source of inaccuracy has been predominantly traced to transient errors. In this work, we have progressively varied the transient error probabilities

for single gates and observed their effect on the *𝑀𝑆𝐸* of these gates.

Attempts are made to formulate the process within a mathematical

framework. The study on the the effect of varying correlation on the

*𝑀𝑆𝐸* was extended to realistic multi-level circuits where single or

have modified the framework to minimize the overall *𝑀𝑆𝐸*. Algorithm multiple gates are subjected to transient errors. For such circuits, we 2 introduces a priority-based approach of choosing *𝑆𝐿𝐸* to reduce the

number of correlator circuits and to obtain the desired level of accuracy quickly under noisy operating conditions. Inevitably, there are conflicts in constraints in different applications, which are handled elegantly.

Algorithm 3 eliminates this issue for a MIMO circuit by introducing

correction blocks at fault specified nodes only. Both these algorithms have been observed to handle noise and yield accurate results, even at high transient error rates. In our future work, we will explore other variants of these algorithms to achieve even better accuracy using lesser number of *ReCo* blocks. Also, equivalent circuits can be developed with a given set of conditions to ease the task of error minimization at a lesser hardware cost.

# CRediT authorship contribution statement

**Shyamali Mitra:** Conceptualized the whole idea, Implemented the algorithms in the circuit level, Mathematical analysis of the proposed method. **Sayantan Banerjee:** Comparison with other algorithms, Find- ing algorithm complexity, Generated relevant graphs. **Mrinal Kanti Naskar:** Reviewed the paper, Providing suggestions.

# Declaration of competing interest

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

# References

1. [Gaines BR. Stochastic computing. In: Proceedings of the april 18-20, 1967, spring](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb1) [joint computer conference. 1967, p. 149–56.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb1)
2. [Alaghi A, Hayes JP. Survey of stochastic computing. ACM Trans Embed Comput](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb2) [Syst (TECS) 2013;12(2s):1–19.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb2)
3. [Najafi MH, Salehi ME. A fast fault-tolerant architecture for sauvola local image](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb3) [thresholding algorithm using stochastic computing. IEEE Trans Very Large Scale](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb3) [Integr (VLSI) Syst 2015;24(2):808–12.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb3)
4. [Li P, Lilja DJ. Using stochastic computing to implement digital image processing](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb4) [algorithms. In: 2011 IEEE 29th international conference on computer design.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb4) [IEEE; 2011, p. 154–61.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb4)
5. [Alaghi A, Qian W, Hayes JP. The promise and challenge of stochastic computing.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb5) [IEEE Trans Comput-Aided Des Integr Circuits Syst 2017;37(8):1515–31.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb5)
6. [Nicolaidis M. Design for soft error mitigation. IEEE Trans Device Mater Reliab](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb6) [2005;5(3):405–18.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb6)
7. Xiao R, Chen C. Gate-level circuit reliability analysis: A survey. VLSI Des 2014;2014. <http://dx.doi.org/10.1155/2014/529392>.
8. Vallero A, Savino A, Chatzidimitriou A, Kaliorakis M, Kooli M, Riera M, et al. SyRA: Early system reliability analysis for cross-layer soft errors re- silience in memory arrays of microprocessor systems. IEEE Trans Comput 2019;68(5):765–83. <http://dx.doi.org/10.1109/TC.2018.2887225>.
9. Xu X, Ban T, Li Y. SPLM: A flexible and accurate reliability assessment model for logic circuits. J Circuits Syst Comput 2019;28(02):1950032. [http://dx.doi.](http://dx.doi.org/10.1142/S0218126619500324) [org/10.1142/S0218126619500324](http://dx.doi.org/10.1142/S0218126619500324), [arXiv:10.1142/S0218126619500324](http://arxiv.org/abs/10.1142/S0218126619500324).
10. [Miskov-Zivanov N, Marculescu D. Multiple transient faults in combinational and](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb10) [sequential circuits: A systematic approach. IEEE Trans Comput-Aided Des Integr](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb10) [Circuits Syst 2010;29(10):1614–27.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb10)
11. [Miskov-Zivanov N, Marculescu D. A systematic approach to modeling and anal-](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb11) [ysis of transient faults in logic circuits. In: 2009 10th international symposium](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb11) [on quality electronic design. IEEE; 2009, p. 408–13.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb11)
12. [Miskov-Zivanov N, Marculescu D. MARS-C: modeling and reduction of soft errors](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb12) [in combinational circuits. In: Proceedings of the 43rd annual design automation](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb12) [conference. 2006, p. 767–72.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb12)
13. [Rajaraman R, Kim J, Vijaykrishnan N, Xie Y, Irwin MJ. SEAT-LA: A soft error](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb13) [analysis tool for combinational logic. In: 19th international conference on VLSI](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb13) [design held jointly with 5th international conference on embedded systems](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb13) [design. IEEE; 2006, p. 1–4.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb13)
14. [Cai S, He B, Wang W, Liu P, Yu F, Yin L, et al. Soft error reliability evaluation](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb14) [of nanoscale logic circuits in the presence of multiple transient faults. J Electron](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb14) [Test 2020;36(4):469–83.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb14)
15. Ting P, Hayes JP. Eliminating a hidden error source in stochastic circuits. In: 2017 IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems. 2017, p. 1–6. [http://dx.doi.org/10.1109/DFT.2017.](http://dx.doi.org/10.1109/DFT.2017.8244436) [8244436](http://dx.doi.org/10.1109/DFT.2017.8244436).
16. [Krishnaswamy S, Viamontes GF, Markov IL, Hayes JP. Probabilistic transfer](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb16) [matrices in symbolic reliability analysis of logic circuits. ACM Trans Des Autom](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb16) [Electron Syst (TODAES) 2008;13(1):1–35.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb16)
17. [Alaghi A, Ting P, Lee VT, Hayes JP. Accuracy and correlation in stochastic](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb17) [computing. In: Stochastic computing: techniques and applications. Springer;](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb17) [2019, p. 77–102.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb17)
18. [Krishnaswamy S, Viamontes GF, Markov IL, Hayes JP. Accurate reliability](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb18) [evaluation and enhancement via probabilistic transfer matrices. In: Design,](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb18) [automation and test in europe. IEEE; 2005, p. 282–7.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb18)
19. [Anderson JH, Hara-Azumi Y, Yamashita S. Effect of LFSR seeding, scrambling](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb19) [and feedback polynomial on stochastic computing accuracy. In: 2016 design,](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb19) [automation & test in europe conference & exhibition. IEEE; 2016, p. 1550–5.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb19)
20. [Alaghi A, P. Hayes J. Exploiting correlation in stochastic circuit design. In: 2013](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb20) [IEEE 31st international conference on computer design. IEEE; 2013, p. 39–46.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb20)
21. [Ahlgren P, Jarneving B, Rousseau R. Requirements for a cocitation similarity](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb21) [measure, with special reference to Pearson’s correlation coefficient. J Am Soc](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb21) [Inf Sci Technol 2003;54(6):550–60.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb21)
22. [Budhwani RK, Ragavan R, Sentieys O. Taking advantage of correlation in](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb22) [stochastic computing. In: 2017 IEEE international symposium on circuits and](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb22) [systems. IEEE; 2017, p. 1–4.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb22)
23. [Alaghi A, Hayes JP. On the functions realized by stochastic computing circuits.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb23) [In: Proceedings of the 25th edition on great lakes symposium on VLSI. 2015, p.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb23) [331–6.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb23)
24. [Kanoh T. Absolute value calculating circuit having a single adder. Google Patents;](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb24) [1990, US Patent 4, 953, 115.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb24)
25. [Lee VT, Alaghi A, Ceze L. Correlation manipulating circuits for stochastic](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb25) [computing. In: 2018 design, automation & test in europe conference & exhibition.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb25) [IEEE; 2018, p. 1417–22.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb25)
26. [Shivakumar P, Kistler M, Keckler SW, Burger D, Alvisi L. Modeling the effect of](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb26) [technology trends on the soft error rate of combinational logic. In: Proceedings](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb26) [international conference on dependable systems and networks. IEEE; 2002, p.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb26) [389–98.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb26)
27. [Alaghi A, Chan W-TJ, Hayes JP, Kahng AB, Li J. Trading accuracy for en-](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb27) [ergy in stochastic circuit design. ACM J Emerg Technol Comput Syst (JETC)](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb27) [2017;13(3):1–30.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb27)
28. [Chen T-H, Alaghi A, Hayes JP. Behavior of stochastic circuits under severe error](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb28) [conditions. It-Inform Technol 2014;56(4):182–91.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb28)
29. [Chen T-H, Hayes JP. Analyzing and controlling accuracy in stochastic circuits.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb29) [In: 2014 IEEE 32nd international conference on computer design. IEEE; 2014,](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb29)

[p. 367–73.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb29)

1. Qureshi MA, Sdiri B, Deriche M, Alaya-Cheikh F, Beghdadi A. Contrast enhance- ment evaluation database (CEED2016), vol. 3. 2017, [http://dx.doi.org/10.17632/](http://dx.doi.org/10.17632/3hfzp6vwkm.3) [3hfzp6vwkm.3](http://dx.doi.org/10.17632/3hfzp6vwkm.3), URL <https://data.mendeley.com/datasets/3hfzp6vwkm>.
2. Dhawan AP, Buelloni G, Gordon R. Enhancement of mammographic features by optimal adaptive neighborhood image processing. IEEE Trans Med Imaging 1986;5(1):8–15. <http://dx.doi.org/10.1109/tmi.1986.4307733>.
3. [Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb32) [quality assessment. In: The thrity-seventh asilomar conference on signals, systems](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb32) [& computers, 2003, Vol. 2. 2003, p. 1398–402.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb32)
4. [Gonzalez RC, Woods R, Eddins SL. Digital image processing using MATLAB (R).](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb33) [Upper Saddle River, NJ: Pearson; 2003.](http://refhub.elsevier.com/S2590-0056(22)00058-3/sb33)
5. Welvaert M, Rosseel Y. On the definition of signal-to-noise ratio and contrast-to- noise ratio for fMRI data. In: Yacoub E, editor. PLoS ONE 2013;8(11):e77089. <http://dx.doi.org/10.1371/journal.pone.0077089>.