

Susceptible Workload Evaluation and Protection using Selective Fault Tolerance

Mauricio D. Gutierrez¹  · Vasileios Tenentes¹ · Daniele Rossi² · Tom J. Kazmierski¹

Received: 30 January 2017 / Accepted: 5 June 2017 / Published online: 20 June 2017
© The Author(s) 2017. This article is an open access publication

Abstract Low power fault tolerance design techniques trade reliability to reduce the area cost and the power overhead of integrated circuits by protecting only a subset of their workload or their most vulnerable parts. However, in the presence of faults not all workloads are equally susceptible to errors. In this paper, we present a low power fault tolerance design technique that selects and protects the most susceptible workload. We propose to rank the workload susceptibility as the likelihood of any error to bypass the logic masking of the circuit and propagate to its outputs. The susceptible workload is protected by a partial Triple Modular Redundancy (TMR) scheme. We evaluate the proposed technique on timing-independent and timing-dependent errors induced by permanent and transient faults. In comparison with unranked selective fault tolerance approach, we demonstrate a) a similar error coverage with a 39.7% average reduction of the area overhead or b) a 86.9% average error coverage improvement for a similar area overhead. For

the same area overhead case, we observe an error coverage improvement of 53.1% and 53.5% against permanent stuck-at and transition faults, respectively, and an average error coverage improvement of 151.8% and 89.0% against timing-dependent and timing-independent transient faults, respectively. Compared to TMR, the proposed technique achieves an area and power overhead reduction of 145.8% to 182.0%.

Keywords Selective fault tolerance · Workload susceptibility analysis · Susceptible workload · Output deviations · Permanent faults · Transient Faults

1 Introduction

Reliability of devices has been affected by technology scaling despite its advantages. Devices manufactured using 32 nm technologies and below are more prone to errors produced by all sources of instability and noise [19, 20] due to the elevated cost of mitigating process variability [6] and the escalation of aging mechanisms [16]. As a result, transient and permanent faults can appear in general logic and generate errors in-the-field. Therefore techniques to make Integrated Circuits (ICs) fault tolerant are required. Fault tolerant IC design techniques are utilized for enhancing circuit reliability. These techniques often rely on redundancy of information, time or hardware [7]. Particularly, hardware redundancy consists of the complete or partial replication of a circuit in order to ensure correct functionality. By replicating the circuit, the reliability is increased as it is highly unlikely that an error would occur on every replica at the same time. Triple Modular Redundancy (TMR) utilizes two replicas of the original circuit, whose outputs are passed on to a majority voter [7]. TMR has been widely used

Responsible Editor: M. Goessel

✉ Mauricio D. Gutierrez
mdga1g11@soton.ac.uk

Vasileios Tenentes
v.tenentes@soton.ac.uk

Daniele Rossi
d.rossi@westminster.ac.uk

Tom Kazmierski
tjk@soton.ac.uk

¹ Electronics and Computer Science,
University of Southampton, Southampton, UK

² Applied DSP and VLSI Research Group,
University of Westminster, Westminster, UK

for safety-critical applications where robustness and data integrity are the top priority. Although TMR achieves a high level of reliability, it imposes a high area and power overhead, 200% of the original circuit plus the voter circuits, thus it is not viable for low power applications. Selective Fault Tolerance (SFT) and Selective hardening have been proposed to reduce area overhead and power consumption, by protecting only a subset of the workload of a circuit or its most vulnerable parts [3, 27].

Selective hardening aims to protect the most vulnerable parts of a circuit against soft errors [27]. This has been achieved in microprocessors [13, 14] through the identification of the architectural vulnerability factor of state elements which often requires stuck-at fault injection campaigns to calculate. Moreover, selective hardening has also been used in combinational logic to identify and protect vulnerable gates or nodes. This is achieved by propagating signal probabilities at the RT-Level to estimate the likelihood of an erroneous output caused by soft errors [8, 15, 17, 27]. A recent selective hardening technique uses a lightweight algorithm to rank the soft error susceptibility of logic cones in combinational logic according to their size, which is given by the sum of the fan-in and fan-out of its cells [23]. The highest ranked cones are duplicated and compared to detect if an error has occurred, in which case, shadow latches at the input enable a roll-back mechanism to recover from the error.

On the other hand, SFT as introduced by [3, 4], ensures functional protection of a pre-defined set of input patterns, which are referred to as workload, by using a partial TMR scheme. However, in the presence of a fault, not all input patterns are equally susceptible to it. Some patterns are less protected by the inherent logic masking of a circuit. When such patterns are executed in the presence of faults, the probability that the logic masking will be bypassed and an erroneous response will be generated is higher. Such patterns are defined as the susceptible workload. Previous works on SFT rely on randomly selecting the workload to protect without examining the susceptibility of that workload to faults.

Probabilistic fault models were developed for ranking test patterns according to their ability to sensitize the logic cones of a circuit that are more likely to propagate an erroneous response. Probabilistic fault models are known for improving both the modeled and the unmodeled defect coverage of tests, while not being biased towards any particular type of faults. Output deviations (OD) were introduced in [26] as an RT-Level fault model calibrated through technology failure information that stems from technology reliability characterization, such as inductive fault analysis [9]. This model is utilized for selecting the input patterns that maximize the probability of propagating an erroneous response to the primary outputs. The input patterns with the

highest output deviations have a greater ability to bypass the inherent logic masking of the circuit. In [25] is shown that selecting input patterns with high output deviations tends to provide more effective error detection capabilities than traditional fault models. In [24], a test set enrichment technique for the selection of test patterns is proposed. Output deviations have also been used for enriching the unmodeled defect coverage of tests during x-filling [5] and linear [11, 12, 21] and statistical [22] compression. The output deviation-based metric proposed in [12], was shown to increase the unmodeled defect coverage of test vectors by considering both timing-independent defects, such as stuck-at faults, and timing-dependent defects, such as transition faults which require two patterns.

In this paper, we present a novel low power fault tolerance design technique applicable at the register-transfer-level, that selects and protects the most susceptible workload on the most susceptible logic cones by targeting both timing-independent and timing-dependent errors. Preliminary results of this technique were presented in [10], where only the timing-independent errors induced by stuck-at faults and input bit-flips were considered. The workload susceptibility is ranked as the likelihood of any error to bypass the inherent logic masking of the circuit and propagate an erroneous response to its outputs when that workload is executed. The susceptible workload is protected by a partial Triple Modular Redundancy (TMR) scheme. To evaluate the fault-tolerance ability of the proposed technique, we consider as surrogate error models the timing-independent errors induced by stuck-at faults and transient input bit-flips. We also consider the timing-dependent errors induced by transition faults and temporary erroneous output transitions. We demonstrate that the proposed technique can achieve a similar error coverage with an average 39.7% area/power cost reduction. Furthermore, it can improve by 86.9% on average the achieved error coverage with a similar area/power cost. Particularly, when protecting only the 32 most susceptible patterns, an average error coverage improvement of 53.1% and 53.5% against errors induced by stuck-at and transition faults is achieved, respectively, compared to the case where the same number of patterns are protected without any ranking. Additionally, we observe an average error coverage improvement of 151.8% and 89.0% against temporary erroneous output transitions and errors induced by bit-flips, respectively. These error coverage improvements incur in an area/power cost in the range of 18.0–54.2%, which corresponds to a 145.8–182.0% reduction compared to TMR.

This paper is organized as follows. Section 2 presents an overview of previous works on Selective Fault Tolerance, reviews the probabilistic fault model of output deviations, introduces the concepts of uncorrelated and application-specific workloads and presents a motivational example.

Section 3 describes the proposed probabilistic selective fault tolerance design technique and output deviation-based ranking metrics used for pattern ranking. Simulation results from the application of the proposed technique to a set of the LGSynth’91 and ISCAS’85 benchmarks are discussed in Section 4. Finally, the concluding remarks are given in Section 5.

2 Motivation

In this section, the concept of Selective Fault Tolerance and the probabilistic fault model of output deviations (OD) are reviewed and the different types of workloads considered in this paper are briefly introduced. The capabilities of the OD model of detecting errors induced by multiple types of faults compared to the random selection of input patterns is presented as a motivational example.

2.1 Selective Fault Tolerance

Selective Fault Tolerance (SFT) was proposed as a modification of TMR. SFT reduces TMR cost by protecting the functionality of the circuit for only a subset of input patterns [3]. This input pattern subset X_1 is selected randomly by the designer. The input patterns within the subset are ensured to be protected with the same level of reliability of TMR, while the rest are not guaranteed protection.

Figure 1 depicts the existing technique of SFT design. For a circuit S_1 to be protected using SFT, two smaller circuits s_2 and s_3 are generated. The behaviour of circuits s_2 and s_3 with a protected set X_1 is described as follows:

$$S_1(x) = s_2(x) = s_3(x), \quad x \in X_1 \tag{1}$$

$$(S_1(x) = s_2(x)) \vee (S_1(x) = s_3(x)), \quad x \notin X_1 \tag{2}$$

According to the heuristic presented in [4], in order to determine if the input x falls within the protected set X_1 , the

characteristic function $\chi(x)$ must be specified. The output of this function is passed on to a modified majority voter as shown in Fig. 1, where the outputs of s_2 and s_3 are considered if and only if $\chi(x) = 1$, otherwise the output of the protected design is the output of circuit S_1 .

$$\chi(x) = \begin{cases} 1 & x \in X_1 \implies S = V_{out} \\ 0 & x \notin X_1 \implies S = S_1 \end{cases} \tag{3}$$

A simplified method for SFT was proposed in [2], where the circuits s_2 and s_3 are replaced by identical circuits. These circuits are the minimal combinational circuits that for an input pattern within the protected set X_1 , exhibit the same output as the original circuit S_1 . The protected design can be described in the following form: (*don't care* indicates undefined values).

$$S_1(x) = s_2(x) = s_3(x), \quad x \in X_1 \tag{4}$$

$$(s_2(x) = don't\ care) \wedge (s_3(x) = don't\ care), \quad x \notin X_1 \tag{5}$$

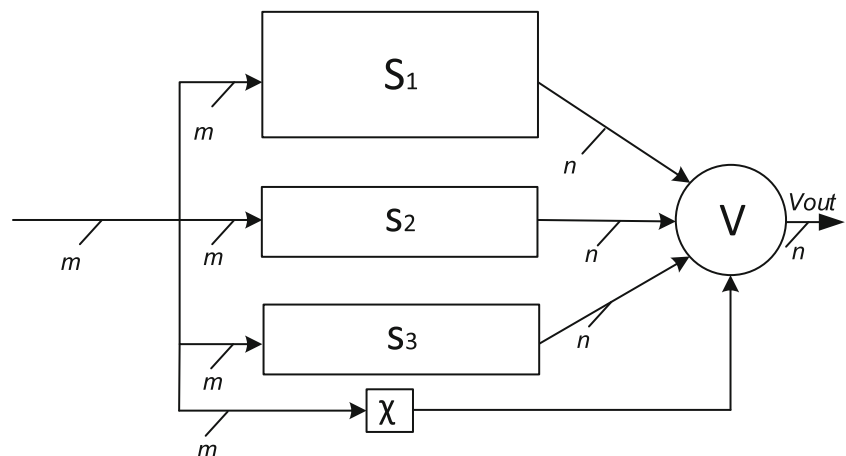
2.2 Probabilistic Fault Model: Output Deviations

The selection of input patterns with high output deviations tends to provide higher error detection capabilities than traditional fault models [25]. Output Deviations are used to rank patterns according to their likelihood of propagating a logic error. There are a few requirements to compute the output deviations of an input pattern. First, a confidence level vector is assigned to each gate in the circuit. The confidence level R_k of a gate G_k with N inputs and one output is a vector with 2^N elements such as:

$$R_k = (r_k^{00\dots00}, r_k^{00\dots01}, r_k^{00\dots10}, r_k^{00\dots11}, \dots, r_k^{11\dots11}) \tag{6}$$

where each $r_k^{xx\dots xx}$ denotes the probability that G_k 's output is correct for the corresponding input pattern. The actual probability values of the confidence level vectors can be generated from various sources, e.g., inductive fault

Fig. 1 Previous Selective Fault Tolerance design



analysis, layout information or transistor-level failure probabilities. In this paper, the probability values obtained by inductive fault analysis shown in [9] are used. However, as it is discussed in [26], deviation-based test patterns may be generated using various sets of confidence level vectors.

Propagation of signal probabilities in the circuit follows the principle shown in [18], with no consideration for signal correlation to reduce computation complexity. The signal probabilities $p_{k,0}$ and $p_{k,1}$ are associated to each net k in the circuit. In the case of the NOR gate G_2 with inputs c, d and output f , the propagation of signal probabilities with the confidence level vector r is as follows:

$$p_{z,0} = p_{c,0}p_{d,0} (1 - r_k^{00}) + p_{c,0}p_{d,1}r_k^{01} + p_{c,1}p_{d,0}r_k^{10} + p_{c,1}p_{d,1}r_k^{11} \tag{7}$$

$$p_{z,1} = p_{c,0}p_{d,0}r_k^{00} + p_{c,0}p_{d,1} (1 - r_k^{01}) + p_{c,1}p_{d,0} (1 - r_k^{10}) + p_{c,1}p_{d,1} (1 - r_k^{11}) \tag{8}$$

Example: considering $r = (0.8, 0.9, 0.9, 0.9)$, if input $c=0$ and $d=0$ ($p_{c,0}p_{d,0}=1$), then $p_{z,0}=0.2$ and $p_{z,1}=0.8$. Similarly, for $c=1$ and $d=1$ ($p_{c,1}p_{d,1}=1$), the propagated signal probabilities result in $p_{z,0}=0.9$ and $p_{z,1}=0.1$.

The same principle is applied to compute the signal probabilities for all gate types. For a gate G , let its fault-free output value for the input pattern t_j be $d, d \in (0, 1)$. The output deviation Δ_{Gj} of G for input pattern t_j is defined as $p_{G\bar{d}}$, where \bar{d} is the complement of d ($\bar{d} = 1 - d$). In other words, the output deviation of an input pattern is a measure, unbiased towards any fault model, of the likelihood that the output is incorrect due to a fault in the circuit [26].

Example Figure 2 shows a circuit with a confidence level vector associated with each gate. The table presents three input patterns and their output deviations. The first column shows the input pattern (a, b, c, d), along with the expected fault-free output value z . The next columns show the signal probabilities for both logic '0' and '1' of the two internal nets and the primary output (e, f, z). The output deviation of a pattern is the likelihood that an incorrect value is observed at the output z . Therefore the output deviations (the erroneous behaviour in G_3) for the presented input patterns are: $\Delta_{3,0000} = p_{z,1}, \Delta_{3,0101} = p_{z,1}, \Delta_{3,1111} =$

$p_{z,0}$. In this example, the input pattern 1111 has the greatest output deviation, with a probability of observing a 0 (the erroneous value) at z of $p_{z,0}=0.396$, thus offering the highest likelihood of detecting an error. •

2.3 Workload Types

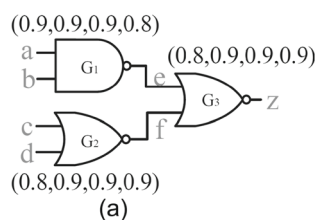
In the context of this paper, we consider two workload types based on whether the application is known during design:

- *Uncorrelated workload:* We consider that the application of the IC is not known during the design time, such as general purpose processors, and, hence, the in-the-field workload can only be considered uncorrelated. Only protection against the timing-independent errors, such as those induced by stuck-at faults or input bit-flips, can be targeted.
- *Application-specific workload:* The application of the IC is known during the design time, hence some information related to the in-the-field workload might also be available. Therefore, the workload can be protected against both timing-independent and timing-dependent errors, because the input patterns of the IC might be correlated allowing the identification of consecutive susceptible patterns within the workload.

2.4 Motivational Example

Table 1 presents the results of a motivational example that shows how different patterns in a workload may exhibit different susceptibility to errors. We select two sets of patterns. The first set is random patterns (rp) and the second set is selected based on the probabilistic fault model of output deviations (pp) [26] for the combinational circuit pdc from the LGSynth'91 benchmarks. For this experiment, we generate rp and pp sets by gradually increasing the size of the sets from 8 to 1024 patterns. The values of the random patterns presented (rp columns), are obtained using the average results of 30 different sets. The first column shows the size of the rp and pp sets. The next columns show the fault coverage of the rp and the pp, respectively, obtained by fault simulating the circuit against the permanent fault models of stuck-at (SA) and transition faults. These results represent the ratio of faults which affect the operation of the circuit for the examined input patterns. The pp set used to compute

Fig. 2 Output deviations example [26]: (a) simple circuit with confidence level vectors and (b) propagated output deviations



Inputs, z	$p_{e,0}$	$p_{e,1}$	$p_{f,0}$	$p_{f,1}$	$p_{z,0}$	$p_{z,1}$
0000, 0	0.1	0.9	0.2	0.8	0.886	0.114
0101, 0	0.1	0.9	0.9	0.1	0.837	0.163
1111, 1	0.8	0.2	0.9	0.1	0.396	0.604

(b)

Table 1 Measured SA fault coverage, transition fault coverage, susceptibility to bit-flips and temporary erroneous output transition between probabilistic and random patterns, for circuit *pd*c

patterns	SA fault coverage (%)		Impr. (%)	Tran. fault coverage (%)		Impr. (%)	Bit-flips suscept. (%)		Impr. (%)	EOT suscept. (%)		Impr. (%)
	rp	pp		rp	pp		rp	pp		rp	pp	
8	4.9	14.2	189.4	1.7	7	300.0	0.6	1.9	216.0	0.3	1.0	287.5
16	5.3	19.6	268.6	3.4	11.5	240.4	1.6	3.3	106.3	0.5	2.0	293.8
32	6.5	26.6	311.1	6.3	19.0	202.2	2.6	4.8	84.7	1.4	3.6	161.6
64	9.7	33.7	247.9	10.5	24.6	133.9	5.1	9.4	84.3	2.4	6.9	195.3
128	21.6	42.1	94.7	12.4	32.6	163.2	10.1	16.3	61.4	4.7	14.0	195.0
256	22.4	49.8	122.1	18.5	41.0	121.5	20.8	29.6	42.3	12.3	28.1	127.4
512	37.0	56.2	51.9	24.8	49.0	97.9	45.3	57.3	26.5	23.1	49.7	114.8
1024	39.0	61.8	58.5	33.9	57.3	69.1	83.3	93.5	12.0	53.0	82.2	55.2

the stuck-at fault coverage was obtained by considering an uncorrelated workload, while the pp set used to compute the transition fault coverage was obtained considering an application-specific workload consisting of 20000 patterns. Columns ‘Impr.’ show the improvement of the pp over the rp. The pp set presents a higher susceptibility than the rp set for stuck-at faults, which is in the range [51.9%, 311.1%], and for transition faults in the range of [69.1%, 300%]. This is due to the pp set containing the patterns with the highest output deviations which, by definition, have the highest likelihood to propagate an error to the output [26]. The next columns present results from the susceptibility evaluation of the rp and pp sets of patterns against errors induced by input bit-flips and temporary erroneous output transitions. The pp set used to evaluate the error coverage induced by bit-flips was obtained considering an uncorrelated workload, while the pp set used to compute the temporary erroneous output transitions was obtained by considering an application-specific workload. Particularly, input bit-flips are conducted by flipping a single bit at every input pattern of an uncorrelated workload. The values shown are the percentage of bit-flips at the primary inputs that bypassed the logic masking and propagated through the circuit to reach the output. For such errors, the pp set exhibit higher susceptibility [12.0%, 216.0%] compared to the rp set. In the case of temporary erroneous output transitions, which are errors that manifest as sporadic missing transitions at the outputs when applying consecutive pattern pairs within an application-specific workload, we observe that the pp set exhibits an error coverage improvement in the range of [55.2%, 293.8%], compared to the rp set.

3 Proposed Probabilistic Selective Fault Tolerance (PSFT) Design Technique

This section describes the proposed Probabilistic Selective Fault Tolerance (PSFT) design technique.

3.1 PSFT Design

The Probabilistic Selective Fault Tolerance (PSFT) design is presented in Fig. 3. The PSFT design consists of a partial TMR scheme of the original circuit *S*, two smaller redundant circuits *S_P*, and a *Z_P* characteristic function. The latter validates when the inputs of the *S_P* units belong to the protected input pattern set. Different from the previous SFT design, shown in Fig. 1, the original circuit *S* is connected to all the inputs nodes, while the *S_P* and *Z_P* units are only connected to the input nodes of the logic cones selected for protection (*I_p* < *I*, *O_p* < *O*). A majority voter *V_P* is used at the outputs of the circuit which operates only when the output is asserted (*Z* = 1). If *Z* = 0, then the voter propagates the outputs of the original circuit *S*. This functionality is described in Eqs. 9 and 10.

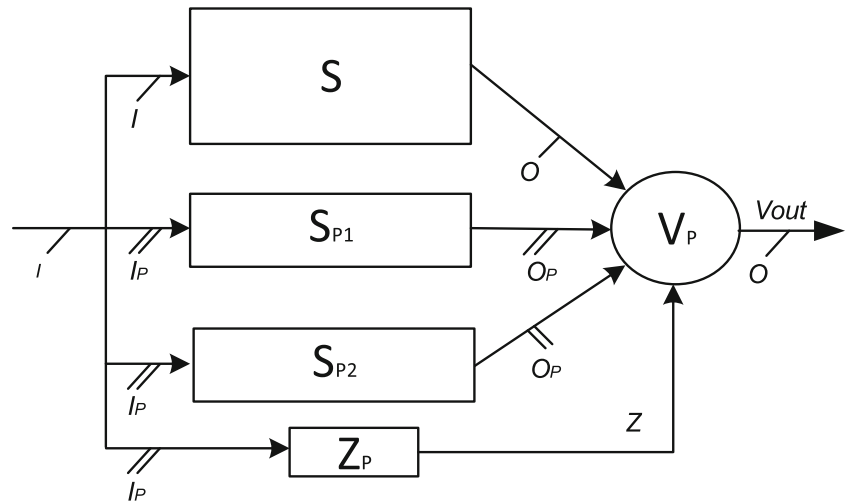
$$S_{p1}(x) = S_{p2}(x) = S(x) \quad x \in Z_p \tag{9}$$

$$S_p(x) = don't\ care \quad x \notin Z_p \tag{10}$$

3.2 Proposed PSFT Design Flow

Figure 4 presents the flow diagram for the proposed PSFT design technique. The number of patterns to protect (*N*) and the percentage of cones to protect *C_p* are considered as parameters of the proposed technique. The proposed technique consists of two processes. First, the *logic cone selection*, determines which logic cones of the circuit to protect given the *C_p* parameter, and produces the Selected Cones list *S_c*. The next process is the *pattern ranking and selection*, which consists of two different sub-processes depending on the workload type (uncorrelated or application-specific). For uncorrelated workload, a timing-independent ranking is performed on a large number of patterns and the *N* patterns who exhibit the highest output deviation metric are selected. For application-specific workload, a timing-dependent ranking of consecutive pattern pairs is deployed

Fig. 3 Proposed Probabilistic Selective Fault Tolerance design



to select the $N/2$ consecutive pairs that maximize the output deviation metric. Finally, the list of protected patterns is ready and may be synthesized.

3.3 Process 1 : Logic Cone Selection

The C_p (cone percentage) parameter defines the percentage of the cones to be protected. This parameter allows that the largest cones, in which errors are most likely to occur, are prioritized for protection, similarly to the cone selection technique presented in [23]. Initially, the cones are weighted according to their *exclusive size* $|C_{es}|$. The exclusive size

of a cone $|C_{es}|$ is the number of cells included in that cone that are not contained in any previously selected cone. The process begins by setting the percentage of selected cones C_{sp} to $1/(\# \text{ cones})$. The cone with the highest exclusive size $|C_{es}|$ is picked for protection by including it in the selected cones list S_c . The percentage of selected cones P_s is increased by $1/(\# \text{ cones})$ and the exclusive sizes of each cone are updated. The process repeats until P_s is higher or equal to the target C_p value. Finally, the selected cones list S_c is passed on to the pattern ranking and selection process.

Example The logic cone selection process for a small circuit is shown in Fig. 5. The three logic cones of the circuit have been marked and ranked according to their exclusive size $|C_{es}|$. The cone Z_1 has a $|C_{es}|$ of 8 cells, the cone Z_2 of 4 and the cone Z_3 of 1 cell. Note that the Z_2 cone has an actual size of 7 cells and an exclusive size of 4, because it shares 3 cells with cone Z_1 that have been discarded when calculating their exclusive sizes. The C_p parameter allows a trade-off between area overhead and error coverage. Particularly for this example, when $C_p = 0.3$, the Z_1 cone will be selected. When $C_p = 0.6$, both Z_1 and Z_2 cones are selected. Finally, with a $C_p = 1.0$, all three cones are selected. This trade-off is explored in Section 4. •

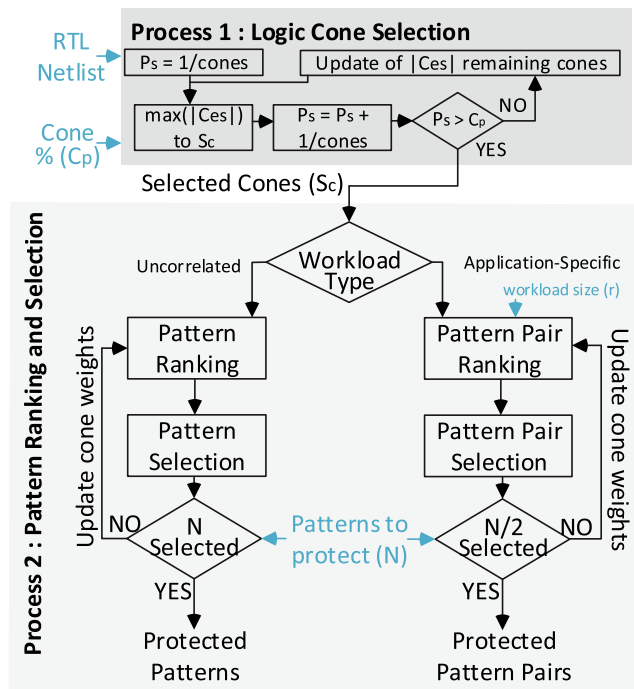


Fig. 4 Proposed design technique flow diagram.

3.4 Process 2 : Pattern Ranking and Pattern Selection

First, all cones in the selected cones list S_c are assigned an initial weight $W(c) = 1$. The weights are used by the pattern ranking process to ensure that the selected patterns to protect are not all biased towards a particular cone in the S_c . Next, according to the workload type, either uncorrelated or application specific, different pattern ranking processes are performed.

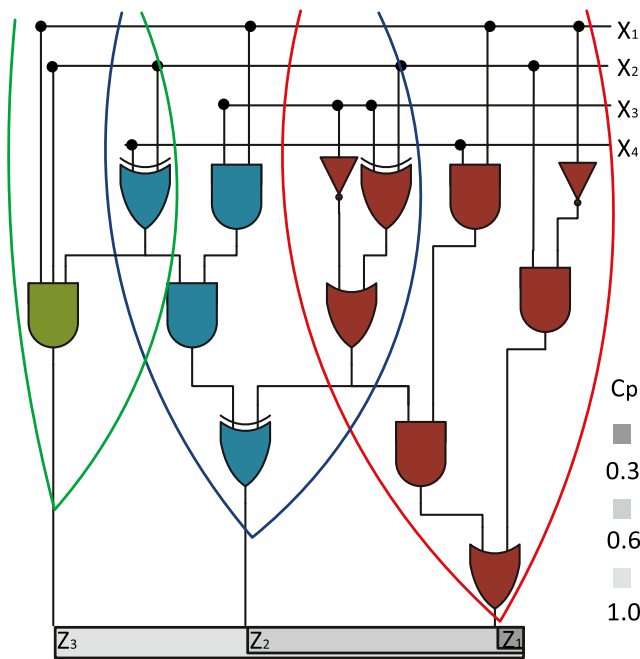


Fig. 5 Example of logic cone selection with different C_p

3.4.1 Timing-independent Pattern Ranking for Uncorrelated Workload

For an uncorrelated workload, where all input patterns are considered equally likely to occur without a known sequence, a ranking of a large number of patterns for protection of timing-independent faults is performed. We define Timing-Independent Deviation $TID(p, c)$ as the output deviation-based metric for ranking input patterns where no sequence of patterns is known.

Let $TID(p, c)$ be the deviation computed of pattern p for cone $c \in S_c$. Then the selected probabilistic pattern Pp is given by:

$$\max(W(c) \cdot TID(p, c)) = W(c) \cdot TID(p, c) \implies Pp = p \tag{11}$$

The selected probabilistic pattern Pp shown in Eq. 11 is the one in which the multiplication of the $TID(p, c)$ with the cone weights $W(c)$, is the maximum after the $TID(p, c)$ of a large number of patterns are computed. The weight of the selected cone c_s in which the maximum $TID(p, c)$ was observed, is updated according to:

$$W(c_s) = W(c_s) \cdot P \tag{12}$$

In order to avoid selecting patterns that target the same largest cones, the weight of the cones that have been already targeted by the selected probabilistic patterns are updated using Eq. 12 after each cone selection. The real-valued parameter P may be set to any value within $[0, 1]$, however, using small values would significantly reduce the weight of

the selected cone, preventing such cone from being selected in the next iteration even if the TID of a pattern for that cone is high. In order to prevent such scenario, we used $P = 0.9$ for our experiments allowing for cones where the TID is consistently high to be selected in adjacent iterations. The whole ranking process is repeated until all the required N probabilistic patterns have been selected.

Example Table 2 presents an example of a pattern selection using the TID metric. Consider the logic cones of Fig. 5 with weights $W(z_3) = 1$, $W(z_2) = 0.9$ and $W(z_1) = 0.81$, and two input patterns p_a and p_b . Applying the ranking based on the TID metric results in a selection of the pattern p_b . This is due to $W(z_2) \cdot TID(p_b, z_2)$ resulting in the highest value (0.63). After this selection, the weight of the cone z_2 is reduced according to Eq. 12.

3.4.2 Timing-dependent Pattern Ranking for Application-specific Workload

In the case of an application-specific workload, where a sequence of patterns of size r is expected, a pattern ranking for timing-dependent faults is deployed. Timing-Dependent Deviation $TDD([p_k, p_{k+1}], c)$ is the output deviation based-metric for ranking consecutive input patterns pairs in a known pattern sequence.

Let $TDD([p_k, p_{k+1}], c)$ be the deviation computed of consecutive pattern pairs $[p_k, p_{k+1}]$ for cone $c \in S_c$ where $k = (0, 1, 2, \dots, r)$ and r is the size of the application-specific workload. The selected probabilistic pattern pair $[P_p, P_{p+1}]$ is calculated as follows:

$$\begin{aligned} \max(W(c) \cdot TDD([p_k, p_{k+1}], c)) &= W(c) \cdot TID(p, c) \\ \implies [P_p, P_{p+1}] &= [p_k, p_{k+1}] \end{aligned} \tag{13}$$

The consecutive pattern pair $[P_p, P_{p+1}]$ shown in Eq. 13 is the one in which the multiplication of the $TDD([p_k, p_{k+1}], c)$ with the cone weights $W(c)$, is the maximum of all possible consecutive pattern pairs in the application-specific workload. Similarly to the timing-independent ranking, the weight of the cone c_s in which the

Table 2 TID pattern ranking and selection example

Cone	Weight $W(c)$	Pattern	$TID(p, c)$	$W(c) \cdot TID(p, c)$
z_3	1.00	p_a	0.25	0.25
		p_b	0.40	0.40
z_2	0.90	p_a	0.55	0.495
		p_b	0.70	0.63
z_1	0.81	p_a	0.75	0.607
		p_b	0.70	0.567

Table 3 Benchmark circuit Inputs/Outputs and Gates

Benchmark	I/O	Gates	Benchmark	I/O	Gates
pdcc	16/40	806	ex5	8/63	256
table3	14/14	445	t481	16/1	389
z9sym	9/1	291	c880	60/26	383
c3540	50/22	1699	c5315	178/123	2307
c6288	32/32	2406	c7552	207/108	3512

maximum $TDD([p_k, p_{k+1}], c)$ was observed, is reduced according to Eq. 12. This process is repeated until $N/2$ consecutive pattern pairs have been selected.

4 Experimental Validation

This section evaluates the proposed selective fault tolerance design technique by applying it on a subset of combinational circuits from the LGSynth’91 and ISCAS’85 benchmark suites (Table 3). The simulation setup for this evaluation is detailed. Comparison results against randomly protecting workload are presented with a discussion on the area cost of the proposed technique and a trade-off analysis for various number of patterns (N) and cone percentages (C_p). Finally, a comparison of computational effort between the proposed technique and statistical fault injection is performed.

4.1 Evaluation Setup

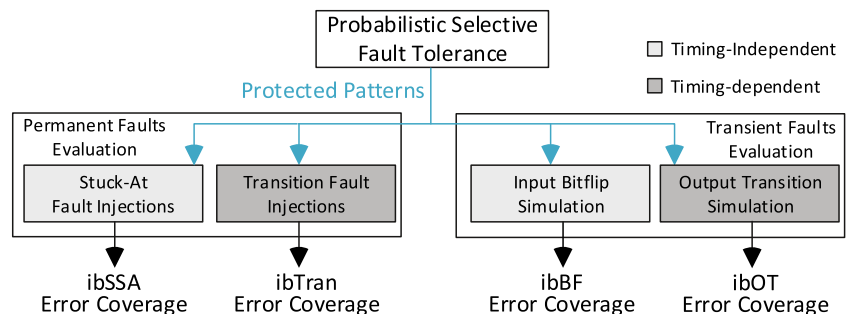
First, the most susceptible patterns and pattern pairs are synthesized using the ABC synthesis tool [1] into the proposed partial TMR scheme (Fig. 3) used by the PSFT technique. Figure 6 depicts the simulation setup used for the evaluation of the error coverage (EC) achieved by the proposed technique against errors induced by permanent and transient faults. For the permanent faults evaluation, single stuck-at (SSA) faults and transition faults are injected and fault simulated using commercial tools in order to obtain the EC of errors induced by SSA (**ibSSA EC**) and induced by transitions (**ibTran EC**). For the transient faults evaluation, bit-flips are injected at the inputs of the circuit to

Table 4 Improvement of Error Coverage (EC) of errors induced by Single Stuck-At faults (ibSSA)

N	ex5		Impr (%)	pdcc		Impr (%)
	ibSSA EC (%)	Impr (%)		ibSSA EC (%)	Impr (%)	
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	15.5	16.4	5.8	2.3	5.1	122.3
16	47.7	59.3	24.3	5.3	19.6	268.6
128	75.2	79.9	6.3	21.6	42.1	94.7
1024	84.0	91.0	8.4	39.0	61.8	58.5
	t481			table3		
N	ibSSA EC (%)	Impr (%)		ibSSA EC (%)	Impr (%)	
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	12.4	12.5	0.8	2.6	6.1	134.4
16	23.8	28.1	18.0	11.2	28.5	155.5
128	45.2	47.0	4.0	33.0	51.5	56.3
1024	61.7	62.0	0.5	74.5	76.9	3.3
	z9sym			c880		
N	ibSSA EC (%)	Impr (%)		ibSSA EC (%)	Impr (%)	
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	4.8	7.7	60.4	22.5	39.5	75.8
16	26.1	31.8	21.9	62.5	76.5	22.3
128	65.6	69.8	6.4	84.2	95.7	13.6
1024	91.7	94.0	2.5	95.1	99.8	5.1
	c3540			c5315		
N	ibSSA EC (%)	Impr (%)		ibSSA EC (%)	Impr (%)	
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	16.1	22.0	36.5	26.3	30.5	16.2
16	45.1	57.7	27.9	67.6	81.9	21.1
128	71.5	86.9	21.6	95.8	97.9	2.1
1024	93.7	97.4	3.95	99.6	99.7	0.1
	c6288			c7552		
N	ibSSA EC (%)	Impr (%)		ibSSA EC (%)	Impr (%)	
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	51.5	58.4	13.5	25.9	29.8	15.0
16	89.3	93.9	5.1	64.2	68.1	6.2
128	95.9	99.8	4.1	84.1	86.4	2.7
1024	98.7	100	1.3	89.8	92.1	2.6

obtain the EC of errors induced by bit-flips (**ibBF EC**). The **ibBF EC** is computed by injecting such random upsets and finding 50K events in which a bit-flip on an input pattern

Fig. 6 Evaluation Setup



propagates an error through the whole circuit and reaches the output of the unprotected circuit (S). The *ibBF* EC is the percentage of these upsets that are masked by the redundant circuits (S_p) of the PSFT design and therefore are not affecting the protected circuit. Furthermore, the EC of errors induced by temporary erroneous output transitions (**ibEOT EC**) is calculated for an application-specific workload of size $r = 20K$. The *ibEOT* EC is the percentage of transitions at the output occurring while executing such workload that are protected by the selected consecutive pattern pairs set. The experiments were performed on a Linux x64 desktop machine with an Intel Core i7-3770 CPU and 16GB of available RAM.

4.2 Simulation Results

Table 4 presents a comparison of the *ibSSA* EC in circuit S of a random patterns (S_{rp}) set and the ranked probabilistic patterns (S_{pp}) set for a large uncorrelated workload. As shown in Table 7, SFI execution times are particularly prohibiting for large workloads, therefore random patterns were used for comparison as they emulate an arbitrary selection of workload with no susceptibility information. The values shown in the S_{rp} column is the average of 30 different random patterns selections. The S_{pp} column shows the *ibSSA* EC of the ranked probabilistic patterns. The *Impr* (%) column shows the improvement of the S_{pp} over the S_{rp} calculated as: $Impr\% = (S_{pp} - S_{rp}) / S_{rp} \cdot 100$. Note that the S_{pp} consistently exhibit a higher *ibSSA* EC than the S_{rp} . This improvement saturates as the number of patterns N that is protected increases. This is attributed to the increased probability that the random patterns S_{rp} contain highly susceptible patterns.

Figure 7 presents the resulting *ibSSA* EC and area cost of the PSFT design for the circuit *c880*. The results for a

various number of protected patterns (N) are shown for a selected cone percentage $C_p = 0.1$. The left axis corresponds to the *ibSSA* EC and the right axis to the area cost of the proposed PSFT design. The area cost of the proposed PSFT design is the sum of the area costs of the three blocks (S , S_p & Z_p) (Fig. 3) divided by the size of the original circuit: $PSFT\ Area\ Cost = (2 \cdot S_p + Z_p) / S$. For the scope of this paper the cost of the voters will be ignored. The average area cost resulting after synthesizing many S_{rp} sets is similar to that of synthesizing the S_{pp} set when protecting the same number of patterns. It can be considered that power consumption is proportional to the area cost. Similar to the results shown in Table 4, the *ibSSA* EC of the *pp* is consistently higher than that of the *rp* for all examined N values.

The computation of the *ibSSA* EC of the PSFT design is calculated by adding the coverage in each of the blocks of the design. The EC in the original circuit S is obtained by the protected patterns (S_{pp}). The coverage of the S_p and Z_p blocks is 100%, as the protected patterns sensitize them fully. The *ibSSA* EC of the PSFT design is computed as:

$$EC_{PSFT} = \frac{(2 \cdot |S_p| + |Z_p|) + (S_{pp}) \cdot |S|}{2 \cdot |S_p| + |Z_p| + |S|} \tag{14}$$

where $|S|$, $|S_p|$, and $|Z_p|$ are the sizes of the blocks depicted in Fig. 3 and S_{pp} is the *ibSSA* EC of the ranked probabilistic patterns.

Table 5 shows results obtained for the transition faults evaluation. The comparison of the *ibTRAN* EC of the random patterns S_{rp} and the ranked probabilistic patterns S_{pp} is presented. The results are shown in the same format as Table 4. Note that the S_{pp} also exhibits a higher *ibTran* EC than the S_{rp} , despite transition faults not being targeted specifically by the pattern ranking using the output deviation-based metric.

Fig. 7 Area cost of Benchmark *c880* and the *ibSSA* EC for timing-independent errors for a selected cone percentage $C_p = 0.1$

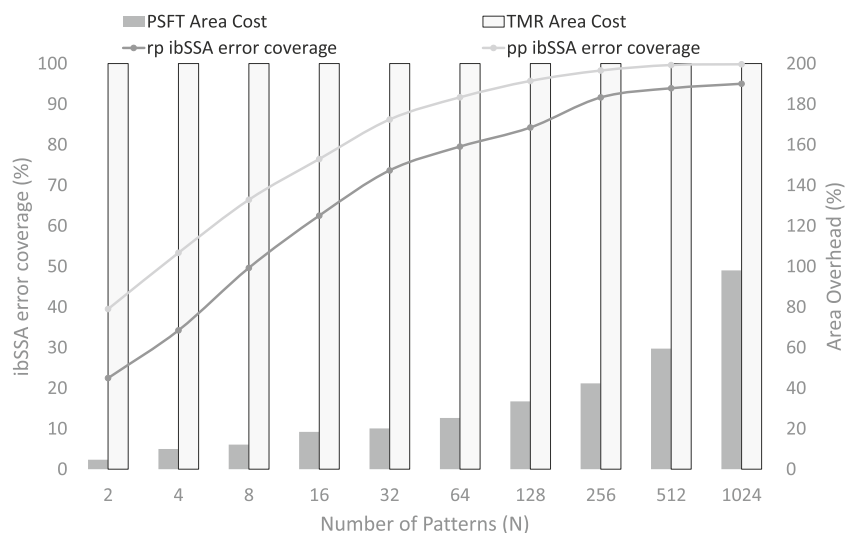


Table 5 Improvement of Error Coverage (EC) of errors induced by Transition faults (ibTran)

N	ex5		Impr (%)	pdc		Impr (%)
	ibTran EC(%)			ibTran EC(%)		
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	5.6	9.0	60.7	0.4	1.2	237.1
16	32.3	38.7	19.8	4.4	11.5	162.4
128	56.7	67.1	18.3	10.4	32.6	213.5
1024	81.3	93.5	15.0	33.9	57.3	69.0
	t481			table3		
N	ibTran EC(%)		Impr (%)	ibTran EC(%)		Impr (%)
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	0.1	0.4	300.0	0.8	2.5	219.5
16	1.7	5.6	233.9	3.8	13.5	255.3
128	15.2	18.7	23.2	21.3	24.5	15.0
1024	40.8	44.0	7.9	49.2	53.1	7.9
	z9sym			c880		
N	ibTran EC(%)		Impr (%)	ibTran EC(%)		Impr (%)
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	0.8	1.8	125.0	5.4	12.9	138.9
16	7.8	15.9	103.8	47.5	58.1	22.3
128	46.2	56.4	22.1	74.6	91.9	23.2
1024	77.3	89.1	15.3	92.9	99.6	7.2
	c3540			c5315		
N	ibTran EC(%)		Impr (%)	ibTran EC(%)		Impr (%)
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	4.8	10.2	112.3	7.1	13.2	85.9
16	31.6	35.9	13.6	44.8	52.1	16.3
128	62.4	68.8	10.3	86.8	91.2	5.1
1024	84.6	93.5	10.5	96.2	98.7	2.6
	c6288			c7552		
N	ibTran EC(%)		Impr (%)	ibTran EC(%)		Impr (%)
	S_{rp}	S_{pp}		S_{rp}	S_{pp}	
2	15.5	30.8	98.9	9.9	15.3	54.7
16	81.5	83.0	1.9	49.3	52.7	6.8
128	97.7	99.2	1.6	79.7	83.3	4.5
1024	99.8	100	0.2	89.1	89.3	0.2

The resulting ibTran EC and area cost of the PSFT design for the circuit *c880* are presented in Fig. 8. Similarly to Fig. 7, these results were obtained for a selected cone percentage $C_p = 0.1$, thus protecting only the 10% largest cones in the circuit. In both Figs. 7 and 8, the EC provided by the S_{pp} is consistently higher than for the S_{rp} , even though neither stuck-at faults nor transition faults were targeted when selecting the protected patterns.

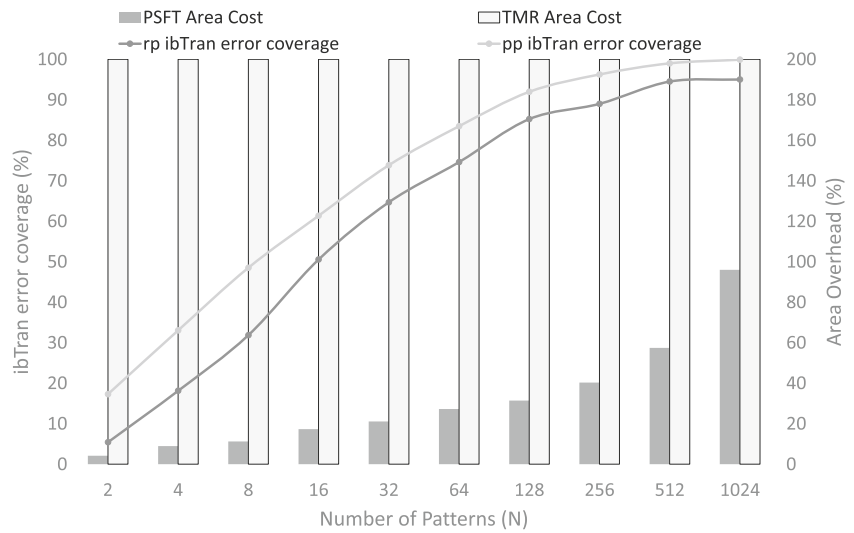
Table 6 shows the ibBF EC and the ibOT EC improvement of S_{pp} over S_{rp} , the ibSSA EC, ibTran EC, area

cost and TMR area improvement when only the top 10% largest logic cones are selected ($C_p = 0.1$) for 8 and 32 protected patterns. The second column shows the ibBF EC improvement calculated by the input bit-flip simulation applied on an uncorrelated workload, while the third column shows the ibEOT EC improvement of the ranked consecutive pattern pairs of an application-specific workload of size $r = 20K$. The ibSSA and ibTran EC of the S_{pp} and S_{rp} sets and the improvement of S_{pp} over S_{rp} are presented in the next columns. Following is the total ibSSA and ibTran EC of the whole PSFT design, as calculated by Equation 14. The area cost of S_P and Z_P blocks (Fig. 3) as well as the overall area cost of the proposed PSFT design are presented in the next three columns, respectively. The improvement in area cost over TMR is presented in the last column, which is calculated as $TMR_{impr} = 200 - \text{area cost of PSFT}$. Error coverage on the LGSynth'91 benchmarks is lower than on the ISCAS'85 benchmarks. This is due to the nature of the circuits, given that the ISCAS'85 circuits were created as a basis for comparing results of test generation, while the combinational LGSynth'91 benchmarks are mainly used in the logic synthesis and optimization field.

When 32 patterns are protected, Table 6 shows an average ibBF EC improvement of 89.0% and an average ibOT EC improvement of 151.8% of S_{pp} over S_{rp} , an average ibSSA EC of 70.5% with an average improvement of 53.1% and an average ibTran EC of 56.2% with an average improvement of 53.5%. This results in an overall average improvement of 86.9%. We observe an area cost in the range of 18.0–54.2% for all circuits, which corresponds to a 145.8–182.0% reduction compared to TMR. Note that for circuit *c880* using only 32 susceptible patterns selected with the output deviations-based metric, provides an ibSSA EC of 88.5% and an ibTran EC of 78.2% with an area cost of only 20.2%. The results of circuit *c880* exhibit on average a ibBF EC of 4.47% for S_{rp} and of 7.49% for S_{pp} , an improvement of 67.6%. The logic cones selected with a $C_p = 0.1$ have an input space of 2^{10} (10 inputs), therefore, with just 32 out of 2^{10} patterns ($32/2^{10} = 3.13\%$), the proposed technique can cover 7.49% of bit-flips at the inputs. Circuit *ex5* exhibits a large ibBF EC improvement compared to the other benchmarks due to the small input space (2^8), which allows for a simpler identification of the susceptible patterns in a workload.

When a specific error coverage constraint is set, the size of the ranked probabilistic patterns set (S_{pp}) is consistently smaller than the size of the S_{rp} set. For instance, when an ibSSA EC of 80% is required, the S_{pp} set is 12% to 63% smaller than the S_{rp} set. Similarly, for an ibTran EC of 70%, the S_{pp} is 13% to 61% smaller than the S_{rp} set. Considering that the same number of random patterns incurs in a

Fig. 8 Area cost of Benchmark *c880* and *ibTran* EC for timing-dependent errors for a selected cone percentage $C_p = 0.1$



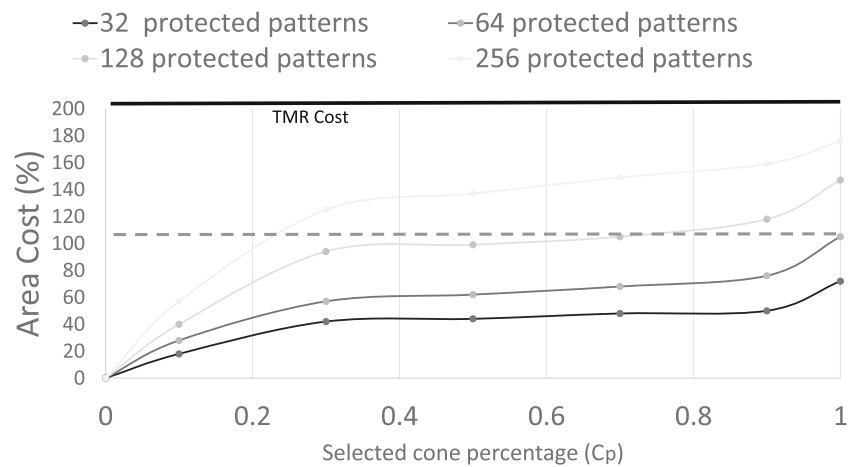
similar area cost, these smaller S_{pp} sets achieve a 39.7% average area reduction compared to the required S_{rp} set size to obtain the same error coverage.

The selective fault tolerance techniques presented in [2] and [4] rely on an arbitrary selection of its workload to protect without examining the susceptibility to either faults or

Table 6 Permanent (induced by single stuck-at and transition) and Transient (induced by erroneous output transitions and by bit-flips) EC, area cost and the improvement over TMR of the proposed technique with $C_p = 0.1$

$C_p = 0.1, N=8$ Benchmark	ibBF EC	ibEOT EC	ibSSA EC (%)				ibTran EC (%)				Area cost (%)			TMR
	Impr.(%)	Impr.(%)	S_{pp}	S_{rp}	Impr.(%)	EC_{PSFT}	S_{pp}	S_{rp}	Impr.(%)	EC_{PSFT}	S_P	Z_P	PSFT	Impr.(%)
ex5	231.6	133.3	44.5	33.2	34.0	47.6	26.0	21.5	21.2	33.8	5.8	0.9	13.4	186.6
pdcc	71.4	162.5	14.2	4.9	189.8	19.5	7.0	1.75	300.0	12.8	1.6	3.4	6.6	193.4
t481	47.2	180.0	21.4	18.1	18.2	28.5	2.9	0.4	663.2	11.7	4.7	0.6	10.1	189.9
table3	45.2	183.3	19.0	7.5	153.3	26.1	7.6	1.92	295.3	15.7	2.9	3.8	9.7	190.3
z9sym	189.7	151.5	19.6	12.4	58.1	25.5	10.7	7.2	48.6	17.2	3.4	1.1	7.9	192.1
c880	55.1	44.7	63.4	59.8	6.0	67.3	41.8	36.9	13.4	48.1	5.5	9.0	14.5	185.5
c3540	222.6	328.6	43.6	39.4	10.5	46.8	24.3	21.4	13.3	31.1	2.9	0.4	6.2	193.8
c5315	201.8	168.4	68.9	50.6	36.2	74.4	37.5	32.9	14.0	48.6	9.9	1.9	21.7	178.3
c6288	243.5	50.0	87.0	83.8	3.8	87.2	68.2	64.2	6.2	71.0	0.8	0.1	1.7	198.3
c7552	143.9	275.0	57.8	55.0	5.05	69.0	41.2	38.5	7.0	46.5	17.1	2.2	36.5	163.5
Average	145.2	167.7	43.9	36.5	51.5	49.2	26.7	22.7	138.0	33.7	5.5	2.3	12.8	187.2
$C_p = 0.1, N=32$ Benchmark	ibBF EC	ibEOT EC	ibSSA EC (%)				ibTran EC (%)				Area cost (%)			TMR
	Impr.(%)	Impr.(%)	S_{pp}	S_{rp}	Impr.(%)	EC_{PSFT}	S_{pp}	S_{rp}	Impr.(%)	EC_{PSFT}	S_P	Z_P	PSFT	Impr.(%)
ex5	266.8	230.8	71.0	60.8	16.8	75.9	53.1	50.6	5.0	61.0	10.1	3.8	24.0	176.0
pdcc	75.4	308.7	26.6	6.5	311.1	37.8	19.0	6.3	202.2	31.3	3.7	10.7	18.0	182.0
t481	24.4	158.3	34.2	30.4	12.5	48.1	7.3	5.2	41.5	15.7	12.7	1.7	27.1	172.9
table3	25.1	257.1	37.0	15.4	140.0	48.3	20.2	6.9	191.5	34.6	7.6	8.8	24.0	176.0
z9sym	31.5	165.4	44.8	37.6	19.1	55.3	28.8	17.6	63.6	42.3	10.5	2.4	23.4	176.6
c880	67.6	59.5	86.2	78.0	10.5	88.5	73.8	64.7	14.2	78.2	8.0	4.1	20.2	179.2
c3540	138.1	200.0	67.7	59.4	13.9	74.3	49.5	46.8	5.8	54.1	12.2	1.6	26.0	174.0
c5315	92.3	47.9	90.2	86.7	4.0	93.5	72.8	69.1	5.4	82.1	24.4	3.2	52.0	148.0
c6288	79.7	36.4	97.0	95.9	1.18	97.1	92.2	90.8	1.56	92.9	1.3	0.2	2.7	197.3
c7552	88.7	53.8	78.7	75.9	3.7	86.1	67.3	64.9	3.6	70.2	25.5	3.3	54.2	145.8
Average	89.0	151.8	63.3	54.7	53.1	70.5	48.4	42.3	53.5	56.2	11.6	4.0	27.2	172.8

Fig. 9 Area cost of different C_p for benchmark *pdc*



errors. They protect a subset of all the possible input patterns of a combinational circuit. These works present results for a group of small circuits of the LGSynth'91 benchmark suite, which include the single-cone circuits *t481* and *Z9sym*. In the case of circuit *t481*, results in Tables 4, 5 and 6 show that the proposed PSFT technique (S_{pp}) offers various error coverage improvements for both timing-independent and timing-dependent errors compared to an unranked selection of patterns (S_{rp}). Applying PSFT to circuit *Z9sym*, Table 4 shows that the S_{pp} set consistently exhibits higher error coverage than the S_{rp} set. In particular, with $N = 128$, which covers 25% of all possible inputs, the proposed technique offers an error coverage improvement of the S_{pp} over the S_{rp} of 6.38%. Furthermore, with $N = 256$, which covers 50% of the input patterns, the error coverage improvement results in 5.27%. The resulting area cost of the PSFT technique for circuit *Z9sym* with $N = 128$ and $N = 256$ is 99% (TMR impr = 101%) and 139% (TMR impr = 61%), respectively, which is similar to the area cost reported in [2, 4]. The area cost of the two techniques is similar in all cases.

Figure 9 presents the trade-off between area cost of the PSFT design and different cone percentage C_p values for the circuit *pdc*. The area cost of $N = (32, 64, 128, 256)$ protected patterns is shown for all C_p values. With a $C_p = 1$, all the logic cones are selected whereas with $C_p = 0.1$, only the largest 10% of the logic cones in the circuit are chosen. When $C_p = 1$, the PSFT design is synthesized for all cones, which yields a high area cost. This is due to the intrinsic logic sharing present in most circuits which the synthesis tool is unable to simplify. It can be seen as expected, that the area cost decreases until reaching a $C_p = 0$. Note that the area cost of the PSFT design for $C_p = 1$ ranges from 176% for $N = 256$ to 72% for $N = 32$, which decreases to 57% and 18% with $C_p = 0.1$.

4.3 Computational Effort Comparison

We compare the CPU time that is required by the proposed technique (pattern-ranking presented in Section 3.4) in order to compute the most susceptible patterns in an application-specific workload of 2000 patterns (Section 2.3) with that required by a statistical fault injection (SFI) simulation. The SFI simulation consists of five different stuck-at fault injection campaigns on 50% of all possible fault sites while executing the same uncorrelated workload. The circuit outputs are compared against the error-free case on each cycle to determine if an error has occurred. The N patterns that exhibited the highest number of errors across all fault injection campaigns were deemed as the most susceptible for the SFI simulation.

Table 7 shows the CPU runtime of both the SFI simulation and the pattern ranking of the proposed PSFT technique. The second column shows the number of selected patterns N . The third column presents the CPU runtime in seconds required by the SFI to run the five different fault injection campaigns. Note that the execution time is the same for all N as the simulation must run all the 2000 patterns of the workload to then select the N patterns that propagate the most errors. Column 4 shows the time required to find the N most susceptible patterns using the proposed technique. Finally, column 5 presents the speed up achieved by the proposed technique compared to the SFI simulation. The time required by the proposed technique increases as the number of patterns increase, although even for 1024 patterns, the proposed technique is several orders of magnitude faster for selecting the most susceptible patterns in a workload. Additionally, the patterns selected by the proposed technique are not biased towards any particular fault model while those selected by the SFI simulation are biased towards the stuck-at fault model.

Table 7 CPU runtime of SFI and the proposed PSFT technique

	N	SFI (s)	PSFT (s)	x Speed up
ex5	4	16293.5	1.1	14560.2
	16		3.3	4915.9
	64		12.1	1344.7
	256		48.6	335.2
	1024		190.6	85.5
pdc	4	78470.0	1.5	51985.0
	16		3.8	20902.3
	64		12.8	6138.1
	256		48.8	1606.8
	1024		194.1	404.2
t481	4	16469.2	0.7	24267.9
	16		1.6	10028.5
	64		5.4	3029.9
	256		21.4	771.0
	1024		82.4	199.9
table3	4	98964.5	1.1	86972.1
	16		3.4	29645.4
	64		12.2	8082.2
	256		47.7	2072.7
	1024		189.4	522.5
z9sym	4	16387	0.9	18207.8
	16		1.4	11705.0
	64		4.7	3486.6
	256		20.1	815.3
	1024		78.9	207.7
c880	4	4140.9	0.7	6098.6
	16		1.9	2094.0
	64		7.1	578.2
	256		27.8	149.1
	1024		111.3	37.2
c3540	4	36796.5	1.8	20751.4
	16		4.8	7626.7
	64		17.5	2108.1
	256		68.6	536.5
	1024		269.7	136.5
c5315	4	105786.7	2.7	39180.3
	16		7.6	13919.3
	64		30.1	3514.5
	256		107.6	983.1
	1024		475.3	222.6
c6288	4	134628.3	3.0	45000.4
	16		8.6	15575.6
	64		31.5	4273.7
	256		131.2	1026.0
	1024		510.3	263.8

Table 7 (continued)

	N	SFI (s)	PSFT (s)	x Speed up
c7552	4	83125.7	3.4	24210.7
	16		10.2	8185.8
	64		36.6	2270.5
	256		147.2	564.8
	1024		572.4	145.2

5 Conclusion

We showed that not every workload is equally susceptible to errors induced by permanent or transient faults, which results in some input patterns being less protected by the inherent logic masking of the circuit (Table 1). We proposed to rank this susceptibility to errors in order to protect those patterns that have the most likelihood of propagating errors to the output. By combining the technique of Selective Fault Tolerance (Fig. 1) and a probabilistic fault model based on the theory of output deviations (Fig. 2), we proposed a low power selective fault tolerance design technique (Figs. 3 and 4). The proposed technique protects the most susceptible workload of the most susceptible logic cones using a partial TMR scheme. We showed that the proposed technique is up to almost 5 orders of magnitude faster when finding the most susceptible workload than a fault-injection-based methodology. We evaluated the proposed technique by considering as surrogate error models the timing-independent errors induced by permanent stuck-at faults and transient bit-flips and the timing-dependent errors induced by permanent transition faults and temporary erroneous output transitions on a set of benchmarks (Table 3). Trade-offs between achieved tolerance against permanent (Tables 4 and 5) and transient (Table 6) errors, together with area cost (Fig. 9) are also presented. We conclude that the protection of the most susceptible workload through a probabilistic fault model that is unbiased towards any type of fault, ensures that the fault tolerance against any type of errors is enriched. Therefore, the usage of output deviations to determine the most susceptible workload in an application may assist in enhancing circuit reliability beyond the scope of a partial TMR scheme.

Acknowledgments This work has been supported by the Mexican CONACYT and by the EPSRC Grant EP/K034448/1 (the PRiME Programme www.primeproject.org).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. ABC: A system for sequential synthesis and verification. <http://www.eecs.berkeley.edu/alanmi/abc/>
2. Agnola L, Vladutiu M, Udrescu M, Prodan L Simplified selective fault tolerance technique for protection of selected inputs via triple modular redundancy systems. In: 2012 7th IEEE international symposium on applied computational intelligence and informatics (SACI)
3. Augustin M, Goessel M, Kraemer R (2010) Reducing the area overhead of TMR-systems by protecting specific signals. In: 2010 IEEE 16th international on-line testing symposium, pp 268–273
4. Augustin M, Goessel M, Kraemer R (2011) Implementation of selective fault tolerance with conventional synthesis tools. In: 2011 IEEE 14th international symposium on design and diagnostics of electronic circuits systems (DDECS), pp 213–218
5. Balatsouka S, Tenentes V, Kavousianos X, Chakrabarty K (2010) Defect aware x-filling for low-power scan testing. In: Design, automation test in Europe conference exhibition (DATE 2010), p 2010
6. Bhunia S, Mukhopadhyay S, Roy K (2007) Process variations and process-tolerant design. In: 20th international conference on VLSI design held jointly with 6th international conference on embedded systems (VLSID'07), pp 699–704
7. Cazeaux JM, Rossi D, Metra C (2004) New high speed CMOS self-checking voter. In: 10 IEEE international symposium on on-line testing 2004 (IOLTS), pp 58–63
8. Fazeli M, Ahmadian S, Miremadi S, Asadi H, Tahoori M (2011) Soft error rate estimation of digital circuits in the presence of Multiple Event Transients (METs). In: 2011 design, automation & test in Europe, pp 1–6
9. Ferguson FJ, Shen JP (1988) A CMOS fault extractor for inductive fault analysis. *IEEE Trans Comput-Aided Des Integr Circ Syst* 7(11):1181–1194
10. Gutierrez M, Tenentes V, Kazmierski T (2016) Susceptible workload driven selective fault tolerance using a probabilistic fault model. In: 22nd IEEE international symposium on on-line testing and robust system design 2016 (IOLTS)
11. Kavousianos X, Chakrabarty K, Kalligeros E, Tenentes V (2010) Defect coverage-driven window-based test compression. In: 2010 19th IEEE Asian test symposium, pp 141–146
12. Kavousianos X, Tenentes V, Chakrabarty K, Kalligeros E (2011) Defect-oriented LFSR reseeding to target unmodeled defects using stuck-at test sets. *IEEE Trans Very Large Scale Integr (VLSI) Syst* 19(12):2330–2335
13. Maniatakos M, Makris Y (2010) Workload-driven selective hardening of control state elements in modern microprocessors. In: 2010 28th VLSI test symposium (VTS), pp 159–164
14. Maniatakos M, Michael M, Tirumurti C, Makris Y (2015) Revisiting vulnerability analysis in modern microprocessors. *IEEE Trans Comput* 64(9):2664–2674
15. Mohanram K, Touban N (2003) Cost-effective approach for reducing soft error failure rate in logic circuits. In: International test conference, 2003. Proc. ITC 2003
16. Omana M, Rossi D, Bosio N, Metra C (2013) Low cost NBTI degradation detection and masking approaches. *IEEE Trans Comput* 62(3):496–509
17. Pagliarini S, Naviner L, Naviner JF (2012) Selective hardening methodology for combinational logic. In: 2012 13th latin American test workshop (LATW), pp 1–6
18. Parker KP, McCluskey EJ (1975) Probabilistic treatment of general combinational networks. *IEEE Trans Comput C-24*(6):668–670
19. Semiconductor industry association, international technology roadmap for semiconductors (ITRS) 2013. <http://www.itrs2.net/2013-itrs.html>
20. Srinivasan J, Adve S, Bose P, Rivers J (2005) Lifetime reliability: toward an architectural solution. *IEEE Micro* 25(3):70–80
21. Tenentes V, Kavousianos X (2011) Low power test-compression for high test-quality and low test-data volume. In: 2011 20th Asian test symposium (ATS), pp 46–53
22. Tenentes V, Kavousianos X (2013) High-quality statistical test compression with narrow ate interface. *IEEE Trans Comput-Aided Des Integr Circ Syst* 32(9):1369–1382
23. Wali I, Virazel A, Deveautour B, Bosio A, Girard P, Sonza Reorda M (2017) A low-cost reliability cost trade-off methodology to selectively harden logic circuits. *Journal of 91 Electronic Testing: Theory and Applications (JETTA)*
24. Wang Z, Chakrabarty K (2006) An efficient test pattern selection method for improving defect coverage with reduced test data volume and test application time. In: 2006 15th Asian test symposium
25. Wang Z, Chakrabarty K (2008) Test-quality/cost optimization using output-deviation-based reordering of test patterns. *IEEE Trans Comput-Aided Des Integr Circ Syst* 27(2):352–365
26. Wang Z, Chakrabarty K, Goessel M (2006) Test set enrichment using a probabilistic fault model and the theory of output deviations. In: Design automation test in Europe conference, vol 1, pp 1–6
27. Zoellin CG, Wunderlich HJ, Polian I, Becker B (2008) Selective hardening in early design steps. In: 2008 13th European test symposium, pp 185–190

Mauricio D. Gutierrez received the B.Sc. degree in electronics engineering from the Monterrey Institute of Technology, Mexico in 2009 and the M.Sc. in System on Chip from the University of Southampton, UK in 2012, where he is currently a PhD candidate. His research interests include, online testing, low power design and approximate computing.

Vasileios Tenentes received the B.Sc. degree in computer science from the University of Piraeus, Piraeus, Greece, in 2003. He also received the M.Sc. degree in computer science and the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece, in 2007 and 2013, respectively. In the past, he was a R&D Engineer on telecommunication networks with Siemens Athens, and a Software Engineer on CAD tools for mixed-signal designs with Helic, S.A. Athens. He has been a Research Fellow with the University of Southampton, U.K., since 2014 and an ARM Research Engineer, Cambridge, UK, since 2017. His research interests include electronic design automation, design for testability, analysis of wear-out effects, system fault models, design for reliability and energy efficiency, and reliability assessment for IoT infrastructure.

Daniele Rossi received the Laurea degree in electronic engineering and the Ph.D. degree in electronics and computer engineering from the University of Bologna, Italy, in 2001 and 2005, respectively. He is currently a Senior Lecturer at the University of Westminster, UK, in the Applied DSP and VLSI research group. His research interests include fault modeling and design for reliability and test, focusing on low power and reliable digital design, robust design for soft error and aging resiliency, high quality test for low power systems and power electronics. He has co-authored more than 90 papers appeared in international journals and conference proceedings, and holds one patent. Dr. Rossi is a Senior Member of IEEE.

Tom J. Kazmierski received the M.S. degree in electronic engineering from the Warsaw University of Technology, Warsaw, Poland, in 1973, and the Ph.D. degree from the Military University of Technology, Warsaw, in 1976. In 1984, he joined the Department of Electronics and Computer Science, University of Southampton, U.K., where he pursues research into numerical modelling, simulation, and synthesis techniques for computer-aided design of VLSI circuits, modeling of carbonnanotube and graphene devices, and energy-efficient electronic systems. From 1990 to 1991, he worked as a Visiting Research Scientist at IBM VLSI Technology Division, San Jose, CA, USA, where he developed and patented synchronization techniques for multisolver simulation backplanes. He has published more than 150 papers in academic journals and international conferences. Dr. Kazmierski served as the Chair of the IEEE DASC WG 1076.1 (VHLDAMS), General Chair of Forum for Design Languages 2010, and General Chair of Virtual Worldwide Forum for Electronic Design Automation 2011.