

“© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Representing Gate-Level SET Faults by Multiple SEU Faults at RTL

Ahmet Cagri Bagbaba^{*†}, Maksim Jenihhin[†], Raimund Ubar[†], Christian Sauer^{*}

^{*}Cadence Design Systems, Munich, Germany; [†]Tallinn University of Technology, Tallinn, Estonia

Email: ^{*}{abagbaba, sauerc}@cadence.com, [†]{maksim.jenihhin, raimund.ubar}@taltech.ee

Abstract—The advanced complex electronic systems increasingly demand safer and more secure hardware parts. Correspondingly, fault injection became a major verification milestone for both safety- and security-critical applications. However, fault injection campaigns for gate-level designs suffer from huge execution times. Therefore, designers need to apply early design evaluation techniques to reduce the execution time of fault injection campaigns. In this work, we propose a method to represent gate-level Single-Event Transient (SET) faults by multiple Single-Event Upset (SEU) faults at the Register-Transfer Level. Introduced approach is to identify true and false logic paths for each SET in the flip-flops' fan-in logic cones to obtain more accurate sets of flip-flops for multiple SEUs injections at RTL. Experimental results demonstrate the feasibility of the proposed method to successfully reduce the fault space and also its advantage with respect to state of the art. It was shown that the approach is able to reduce the fault space, and therefore the fault-injection effort, by up to tens to hundreds of times.

Index Terms—SET, SEU, multiple faults, functional safety, hardware security, fault injection

I. INTRODUCTION

The fault injection technique is widely used for evaluating functional safety [1] and security threats resilience [2] in integrated circuits. For safety-critical applications, it is an established, accurate method to assess the effectiveness of the deployed safety mechanisms. For security-critical applications, the technique is efficient to mimic an attack by physical fault injection aimed to alter the program flow or the processed data [3]. However, depending on the abstraction level of the circuit and the size of the fault space, a fault injection campaign can be very costly.

One of the challenges of fault injection campaigns is the vast number of possible fault locations. For a simulation-based fault injection campaign [4], engineers simulate a fault-free design and its copies with faults injected one at a time. This may imply enormous execution times, especially for the gate level fault analysis. Hence, there is a high demand for methodologies that can support designers in the early-stage design exploration of reliability factors. Moreover, fault injection into gate-level models is quite late in the integrated circuit development cycle, and any design modifications become more expensive in terms of the required engineering effort. Several researchers delved into the early-stage explorations of the designs for both safety and security applications [5]–[8]. In both safety and security-related applications, early design evaluation is necessary to minimize design iterations and resources, thus to enable faster design closure times.

In this work, we focus on SET faults at the gate level and propose an efficient solution to represent them by multiple SEU faults at the RT level. The relevance of this problem for safety-critical applications grows with the downscaling of the technology nodes, forcing designers to evaluate system's safety against SET faults, which affect combinational elements of the circuit. However, this comprehensive evaluation at the gate level is not affordable in terms of the execution time of fault injection campaigns for the industrial-sized designs. From the security point of view, SET faults at the gate level represent laser fault attacks, which can be observed in flip-flops (FFs) as single or multiple errors [9]. Here, it is crucial to evaluate laser attacks in order to determine which vulnerable SET faults create single or multiple errors in the sequential elements of the design.

To tackle the listed problems, we propose a methodology for representing gate-level transient faults, such as SETs, by Multiple Flip-Flop Upset (MFFU) at RTL. In the case of Soft Error Reliability (SER) assessment for safety applications such as automotive, MFFU becomes functionally equivalent for EDA tools to multiple simultaneous SEUs. For vulnerability analysis against fault-injection attacks on security-critical designs, MFFU refers to single and multi-bit fault injections. In this work, first, we identify static fan-in cones of each FF at the gate level. Second, we perform propagation analysis to identify SET faults that have true (sensitizable) paths to FF inputs. In this way, we obtain optimized FF sets as representatives of all SET faults to guide RTL multiple SEU fault injection campaigns. As a result, this method can successfully reduce the fault space and enhance the high complexity of fault injection campaigns. Without loss of generality, the proposed methodology is demonstrated on a Cadence EDA (Electronic Design Automation) tool flow, but it remains applicable to other tool flows as well. The main contribution of this work is as follows:

- An approach to move the gate-level SET vulnerability analysis to RTL
- A technique to reduce the fault space at RTL by applying gate-level propagation analysis
- A systematic and workload-independent methodology for representing the gate-level SETs by multiple SEUs at RTL supported by industrial-grade EDA tool flow

The rest of the paper is organized as follows. In Section II, we give an overview of the related work. The proposed

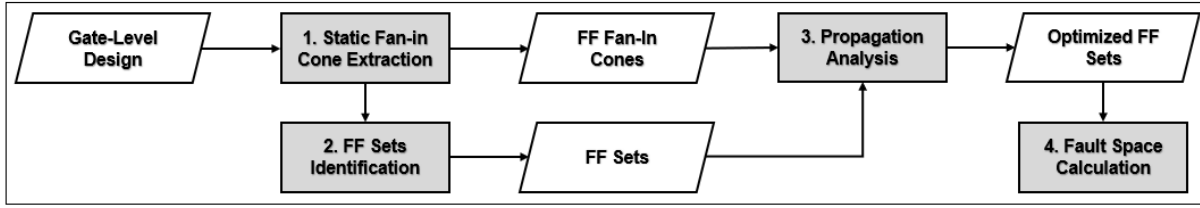


Fig. 1. Steps of proposed methodology.

methodology is explained in Section III. The experimental results are discussed in Section IV. Finally, Section V concludes the paper.

II. RELATED WORKS

Relevant solutions for the above problem are proposed in [7] and [8]. However, these state-of-the-art approaches rely on the static cones pre-analysis only and do not consider if a SET fault actually propagates to the FF inputs. [7] proposes an RTL fault injection model which is representative for laser fault attacks. To do that, the authors analyse the circuits structurally and find intersection cones which guide the fault injection in advance. On the other hand, they neither create FF sets that cover all SET faults nor optimize FF sets by considering true/false paths. Similarly, [8] models the locality of a laser attack in case of multiple-bit faults. The authors analyse the circuits structurally as well and, afterwards, create FF sets. However, the authors consider only the supersets and reject all the subsets. In this way, each combination of SEUs in the superset is a trial to hit a fault in any smaller cone intersection. Yet, the probability of hitting a SET in case of any superset by selected random multiple SEU is low.

There are other studies which investigate the impact of SET faults. [10] estimates the impact of SET faults without layout information by identifying a pair of gates in which SET can propagate to multiple outputs. [11] analyzes the impact of SETs through Algebraic Decision Diagrams and Binary Decision Diagrams (BDD) and [12] improves this method by considering multiple effects. Finally, [13] suggests performing a stochastic gate-level simulation for small circuits. Last but not least, there are some works that investigate the combination of different fault analysis technologies such as [14] and [15]. These works combine the strength of formal methods and fault injection simulators; however, they analyse only permanent faults and do not analyse the representation of gate-level SET faults at RTL.

Different from the works listed above, this paper proposes a more efficient technique to prune the fault space by considering the propagation of SET faults. The significant speedup is achieved by running the RTL fault injection procedure on the accurately selected multiple flip-flop upset faults.

III. REPRESENTING GATE-LEVEL SET FAULTS BY MULTIPLE SEU FAULTS AT RTL

In this work, the aim is to identify Multiple Flip-flop Upset sets for RTL fault injection, which represent all gate-level

SET faults. By doing so, we reduce the number of injections required to evaluate the effect of SET faults.

The SET fault model implies flipping the value of a signal in the combinational cloud and holding the value for a specified period of time. SEU fault model implies flipping the value of the output of a sequential element and holding it until it is overwritten with new data. SEUs can be applied on the outputs of sequential elements, such as memories, FFs and latches. We apply SET faults for one clock cycle length. The proposed flow is shown in Fig. 1 and starts with the (1) extraction of static fan-in cones of each FF in gate-level netlist. In the next step (2), FF sets are created to represent each SET faults on the fan-in cones of FFs. Then, we perform propagation analysis (3) to check if SET faults propagate to the FF inputs. If a SET fault does not propagate, then we check if this changes created FF sets. In this way, we obtain optimized FF sets, which are representative of all SET faults, which propagate to the FF inputs. Finally (4), we calculate the fault space to see the reduction when compared to state-of-the-art and random multi-bit injection approaches. The following subsections explain each step of the proposed method in detail.

A. Static Fan-in Cone Extraction of Flip-Flops at gate level

As a first step, we extract fan-in cones of each FF at the gate level, as it is illustrated in Fig. 2. In the beginning, we generate a list of all faults in the design. Then, we extract fan-in information from all FFs in the ingress combinational part of the design. Each fan-in cone search starts from a FF and expands backward, i.e. in the direction of inputs

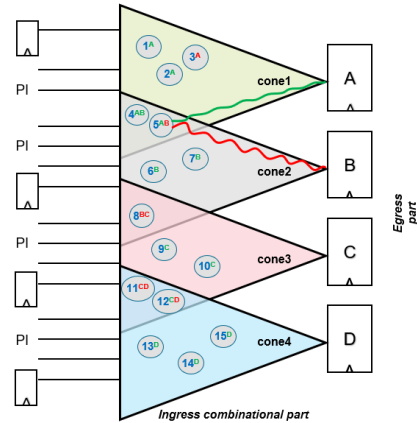


Fig. 2. Extracting fan-in cones of each FF and finding propagation paths.

TABLE I
RESULTS OF EXAMPLE DESIGN GIVEN IN FIG. 2

Affected Cone	FF Sets	Multiplicity	Optimized FF Sets	Optimized Multiplicity
Cone 1	A, B	2	A, B	2
Cone 2	A, B, C	3	A, B	2
Cone 3	B, C, D	3	C	1
Cone 4	C, D	2	D	1

of the combinational cloud until it encounters a FF output or a primary input (PI). Finally, all SETs in each cone are enumerated to map each SET to a FF set. This step is performed by using Cadence® JasperGold Functional Safety Verification App.

B. Flip-Flop Sets Identification

The second step of the proposed methodology is the identification of FF sets, which will be used as a MFFU injection target in the following steps. To do that, we consider each fan-in cone independently and determine FF sets, which cover all possible scenarios, as shown in the second column of Table I. For instance, if cone-1 is affected by a SET fault, we can cover this SET fault by injecting multiple MFFUs on A and B because cone-1 has an intersection with cone-2 which is the fan-in cone of B. This process is repeated for each cone, and FF sets are obtained with a size between 1 (in case the cone does not intersect with any other cones) and N FFs (in case all cones have an intersection).

Extracted FF sets are flip-flops of the circuit potentially affected by a SET. Therefore, MFFU injection can be limited to this set of FF. Table I also shows the multiplicity information of each FF set. The multiplicity of a FF set is the number of FF in a set. For instance, if a SET fault occurs in cone-2, it can propagate to the A, B, C FFs, causing different combinations of upsets on this set. This means that the less is the number of FF in a set (less multiplicity), the higher is the probability of hitting a real MFFU. We will use this information in the following steps. Moreover, multiplicity is important for the calculation of fault space, which will be given in the next sections. It is obvious that there are 8 combinations in one FF set with a multiplicity 3.

C. Propagation Analysis

In this work, unlike state-of-the-art researches, we also take propagation of faults into consideration in order to reduce fault space more. For this step, we deploy the formal techniques to investigate the behaviour of a design under fault. The theory behind formal techniques is creating of Boolean function representation of a design under test so that formal proves can be used. In order to achieve better performance in the modern formal tools, BDDs [16] and Multiway Decision Graphs (MDGs) [17] are widely used.

The formal analysis deploys formal methods to determine the propagation of faults. Propagation analysis verifies if there is a combination of inputs that provoke fault propagation. If a fault propagates to FF inputs, we accept that the fault has a true

path to FF inputs. Otherwise, it has a false path and should be excluded from the analysis. In this step, formal properties to perform the analysis are automatically generated and verified with respect to all possible input stimuli.

The simple and high-level example in Fig. 2 illustrates that there are some SET faults in the intersection cones with a false path to the FF inputs. In this figure, green paths and superscripts point the true paths (fault propagates) while red ones show that the related fault has a false path (fault does not propagate). As a result of this step, we obtain optimized FF sets, as shown in the fourth column of Table I. It is obvious that some larger FF sets are disappeared due to non-observable faults that cannot be propagated. In this way, optimized multiplicities are obtained along with the reduced number of FF sets in some circuits. This step is performed by using Cadence® JasperGold Functional Safety Verification App. In the following subsection, we show a more detailed motivational example for the propagation analysis.

Motivational Example: Removing the paths which cannot be propagated

To explain the propagation analysis in detail, we use a motivational example given in Fig. 3 which has fan-out nodes. The circuit includes an input x , and outputs of the gates **AND1**, **OR1** and **OR2**. The SETs may be simulated only for these fan-outs. The steps of the approach can be listed as follows:

- Static fan-in cone analysis gives us the following FF sets of MFFU faults: (1, 2, 3, 4) for x , (1, 2, 3, 4) for **AND1**, (1, 2) for **OR1**, (2, 3) for **OR2**.
- After removing of duplicated sets, we get the initial sets of MFFU faults: (1, 2, 3, 4), (1, 2), (2, 3).

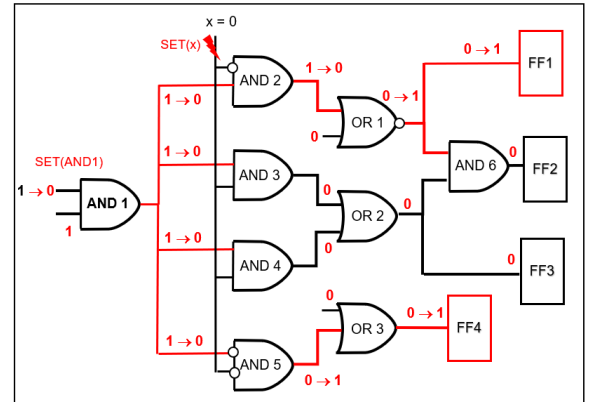


Fig. 3. Motivational example to find propagated and not-propagated faults.

- By propagation analysis, we see that for SET on **AND1** we never reach all FFs, rather only either (1, 4) or (2, 3) due to the fact that the propagation of a SET at **AND1** is controlled by signal $x=0$ (by blocking two of four AND gates). Therefore, the superset (1, 2, 3, 4) for **AND1** should be replaced by subsets (1, 4) and (2, 3). In other saying, SET(**AND1**) is mapped to (1, 4) and (2, 3) FF sets.
- Moreover, the SET on the input x is always blocked either on **AND5** (if output of **AND1**=1), or on **AND2**, **AND3**, **AND4** (if output of **AND1**=0). Hence, the superset (1, 2, 3, 4) for SET(x) should be replaced by (1, 2, 3).
- As a result, we get instead of initial (1, 2, 3, 4), (1, 2), (2, 3), optimized FF sets (1, 2), (2, 3), (1, 4) (2, 3), (1, 2, 3), where (2, 3) can be removed as it is duplicated.
- Thus, the final optimized FF sets: (1, 2), (2, 3), (1, 4), (1, 2, 3).

In this motivational example, we analyzed the propagation of SETs only on x and the outputs of **AND1**, **OR1** and **OR2**. The propagation analysis is sufficient for the SETs at these four locations that also represent the remaining SET faults in the fan-out free regions.

D. Fault Space Calculation

In the fault injection procedure, SEUs are injected in all possible locations and at each clock cycle [18]. Therefore, the number of injections required for a single transient fault is large, especially for the industrial-sized designs. When considering the size and low speed of fault injection simulations at the gate level, optimization methods should be applied. Hence, considering the huge number of SET injections at the gate level, our proposed method significantly reduces the number of injections by identifying optimized FF sets when compared to state-of-the-art and random multi-bit injection approaches applied in safety and security applications.

Our proposed methodology can significantly reduce the fault space by leveraging the FF sets with propagation analysis. In this work, we compare our results with the state-of-the-art and random multi-bit injection. State-of-the-art researches such as [7] and [8] rely on only a static approach and do not consider the propagation analysis. Similarly, the random multi-bit injection method considers all possible FF combinations. In order to calculate fault space or the number of injections, we use the following equation where N is the number of FFs, k_1, k_2, \dots, k_N are the numbers of FF in each set and $1 \leq k_i \leq N$, given in [8].

$$FaultSpace_{Total} = \sum_{i=1}^N (2^{k_i} - 1) \quad (1)$$

By using the above equation, the total fault space for the example given in Fig. 3 can be calculated effortlessly. As it is explained in Section III-C, we have the initial and not-optimized sets which represent the state-of-the-art approach as (1, 2, 3, 4), (1, 2) and (2, 3). By using the given formula, the total number of faults is 21. On the other hand, we have optimized FF sets as (1, 2), (2, 3), (1, 4), (1, 2, 3), which

require 16 number of injections. Therefore, our proposed method can reduce the total fault space from 21 to 16 for the motivational example given in Fig. 3.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In order to verify the effectiveness of proposed methodology, we evaluate our methodology on the ITC'99 [19] benchmark circuits.

In order to perform fan-in cone analysis and propagation analysis, we deploy Cadence tools along with the developed script sets, which execute on gate-level design. Meanwhile, all applied methods remain applicable to other tool flows. In the beginning, we synthesize Verilog or VHDL design through Cadence® Genus™ Synthesis Solution to obtain gate-level representation of the design. Then, steps 1, 2 and 3 shown in Fig. 1 are performed on our application which deploys Cadence® JasperGold Functional Safety Verification App.

We use three methods to show the fault space reduction and compare the results. The first method is "without propagation analysis" which represents the state-of-the-art as in [8]. The main difference between our proposed methodology "with propagation analysis" and the state-of-the-art is the identification of true (sensitizable) paths. We leverage the analysis by identifying SET faults which do not propagate to FF inputs so that fault space is reduced more. In other words, we cut down the pessimism in the results. The third approach used for comparison is "Random Multi-Bit injection". This is basically injecting faults on all possible combinations of FFs randomly that naturally causes huge fault space. Our application is capable of building the fault space for each method and given design without any significant effort.

All experimental results are presented in Table II. The selected designs include various designs from the ITC'99 benchmark. During creating of FF sets, we remove faults on clock and reset signals from the analysis due to the fact that the clock tree is not known in this stage of the design. Other faults except clock and reset are kept as they are. This step is done in our application automatically. We show the number of sets, number of supersets, maximum multiplicity and calculated fault spaces for each analysis and design. The number of sets shows the number of all identified FF sets before duplicated ones are removed. In contrast, the number of supersets points the same after duplicated ones are removed. Total Faults are calculated by using the Equation 1.

In Table II, it can be seen that our proposed methodology reduces the Total Faults significantly when compared to both state-of-the-art and the random multi-bit injection approaches. For some circuits such as **b01** and **b08**, we are able to reduce only the number of supersets while the maximum multiplicity is still the same in both cases. Moreover, there is no optimization achieved in **b06**. For the rest of the circuits given in Table II, we both optimize the number of supersets and maximum multiplicity. Thereby, the total set of faults are optimized significantly, as shown in Fig. 4 (values are normalized). It is observable that total faults in the proposed methodology (orange bars) are less than the other

TABLE II
EXPERIMENTAL RESULTS: FAULT SPACES ACHIEVED BY THREE METHODS

Circuit	# FF	without propagation analysis				with propagation analysis				Random Multi-Bit Injection
		# sets	# superset	max multiplicity	Total Faults	# sets	# superset	max multiplicity	Total Faults	
b01	5	5	2	4	1.80E+01	3	1	4	1.50E+01	3.20E+01 - 1
b02	4	4	1	3	7.00E+00	1	1	2	3.00E+00	1.60E+01 - 1
b03	30	8	3	12	4.14E+03	3	1	9	5.11E+02	1.07E+09 - 1
b04	66	27	10	19	4.00E+06	5	4	8	1.02E+03	7.38E+19 - 1
b05	34	62	2	33	9.00E+09	61	5	31	2.00E+09	1.72E+10 - 1
b06	8	7	5	4	4.30E+01	7	5	4	4.30E+01	2.56E+02 - 1
b07	46	51	2	35	4.00E+10	43	3	26	8.00E+07	7.04E+13 - 1
b08	21	19	2	18	2.70E+05	11	2	18	2.62E+05	2.10E+06 - 1
b09	28	14	1	28	3.00E+08	7	1	27	1.00E+08	2.68E+08 - 1
b10	17	45	9	11	5.91E+03	13	4	11	2.62E+03	1.31E+05 - 1
b11	31	43	9	18	4.65E+05	9	2	16	6.60E+04	2.15E+09 - 1
b13	50	40	13	13	9.15E+03	20	9	9	9.47E+02	1.13E+15 - 1

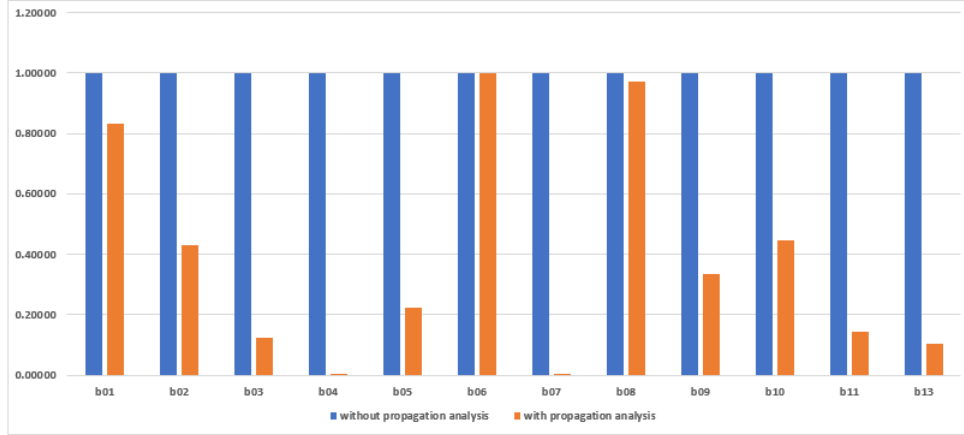


Fig. 4. Fault Space comparison.

two methods. We also add that we reduce the fault space from 1.20 times to a few hundred times when compared without propagation analysis, depending on the circuit.

Moreover, we also compare our results with the well-known Statistical Fault Injection (SFI) approach [20] in case initial population sizes calculated before are used. SFI can be used for transient fault injection campaigns to reduce the execution times while keeping a meaningful number of injections with an error margin. This is one of the possible ways to perform RTL fault injection campaigns after FF sets are defined by using the methodology presented in this paper. In an SFI campaign, the sample size or the margin of the error with a certain confidence level are determined by using the Equation 2 defined in [20]. In this way, it is possible to obtain precise results while injecting a small number of faults [20]. The technique allows to know the margin of error while restricting the campaign time to the minimum. To sum up, there are three confidence levels in SFI as 90%, 95%, and 99.8%. In this work, we only use the 95% confidence level as it is the one that is practically used in the industry. Also, three error margins are defined as 5%, 1% and 0.1%.

$$n = \frac{N}{1 + e^2 \times \left(\frac{N-1}{t^2 \times p \times (1-p)} \right)} \quad (2)$$

In Table III, we show the SFI results. In this table, N shows the initial population. In our case, N is equal to the total faults shown in Table II. Moreover, n(5%), n(1%) and n(0.1%) show the required sample size with the error margins 5%, 1% and 0.1% respectively. This shows that our proposed methodology can prune the fault space from 1.12 times to a few hundred times in case faults are injected by using SFI. Note, the results for some sample sizes remain similar due to the fact that the initial population is always finite. Even so, we show that a significant reduction is achieved by using the proposed methodology, especially when we reduce the error margins. Therefore, it is efficient to use the proposed methodology and to select a sample for fault injection among the pre-defined initial populations in the MFFU space identified using the method "with propagation analysis".

V. CONCLUSIONS

In this work, we propose a methodology to represent gate-level SET faults by multiple SEU faults at RTL. It enables a solution for the high complexity problem of expensive gate-level fault injection campaigns by changing the abstraction level. We improve the state-of-the-art by considering propagation analysis of each SET fault. First, we find static fan-in cones of each FF at the gate level. Second, FF sets are

TABLE III
COMPARISON OF THREE METHODS IN A SFI CAMPAIGN WITH 95% CONFIDENCE LEVEL

Circuit	without propagation analysis				with propagation analysis				Random Multi-Bit Injection			
	N	n(5%)	n(1%)	n(0.1%)	N	n(5%)	n(1%)	n(0.1%)	N	n(5%)	n(1%)	n(0.1%)
b01	1.80E+01	1.70E+01	1.80E+01	1.80E+01	1.50E+01	1.40E+01	1.50E+01	1.50E+01	3.20E+01 - 1	3.00E+01	3.20E+01	3.20E+01
b02	7.00E+00	7.00E+00	7.00E+00	7.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	1.60E+01 - 1	1.50E+01	1.60E+01	1.60E+01
b03	4.14E+03	3.52E+02	2.89E+03	4.12E+03	5.11E+02	2.20E+02	4.85E+02	5.11E+02	1.07E+09 - 1	3.84E+02	9.60E+03	9.60E+05
b04	4.00E+06	3.84E+02	9.58E+03	7.74E+05	1.02E+03	2.79E+02	9.22E+02	1.02E+03	7.38E+19 - 1	3.84E+02	9.60E+03	9.60E+05
b05	9.00E+09	3.84E+02	9.60E+03	9.60E+05	2.00E+09	3.84E+02	9.60E+03	9.60E+05	3.44E+10 - 1	3.84E+02	9.60E+03	9.60E+05
b06	4.30E+01	3.90E+01	4.30E+01	4.30E+01	4.30E+01	3.90E+01	4.30E+01	4.30E+01	2.56E+02 - 1	1.54E+02	2.49E+02	2.56E+02
b07	4.00E+10	3.84E+02	9.60E+03	9.60E+05	8.00E+07	3.84E+02	9.60E+03	9.49E+05	7.04E+13 - 1	3.84E+02	9.60E+03	9.60E+05
b08	2.70E+05	3.84E+02	9.28E+03	2.11E+05	2.62E+05	3.84E+02	9.27E+03	2.06E+05	2.10E+06 - 1	3.84E+02	9.56E+03	6.59E+05
b09	3.00E+08	3.84E+02	9.60E+03	9.57E+05	1.00E+08	3.84E+02	9.60E+03	9.51E+05	2.68E+08 - 1	3.84E+02	9.60E+03	9.57E+05
b10	5.91E+03	3.61E+02	3.66E+03	5.88E+03	2.62E+03	3.35E+02	2.06E+03	2.62E+03	1.31E+05 - 1	3.83E+02	8.95E+03	1.15E+05
b11	4.65E+05	3.84E+02	9.41E+03	3.14E+05	6.60E+04	3.82E+02	8.39E+03	6.18E+04	2.15E+09 - 1	3.84E+02	9.60E+03	9.60E+05
b13	9.15E+03	3.69E+02	4.69E+03	9.06E+03	9.47E+02	2.74E+02	8.62E+02	9.46E+02	1.13E+15 - 1	3.84E+02	9.60E+03	9.60E+05

created pessimistically, meaning that propagation analysis is not considered. Third, we execute propagation analysis by using a formal approach to find SET faults that propagate to FF inputs. Then, optimized FF sets are created again with less pessimism. Finally, we calculate the fault space to show the effectiveness of the proposed methodology. In this way, we significantly reduce the number of fault injections and obtain a higher probability of hitting a true multiple SEU fault. Experimental results show that we make the fault space smaller by up to tens to hundreds of times.

As future work, we aim to apply this methodology for functional safety and security evaluation in industrial-sized CPU designs.

ACKNOWLEDGMENT

This research was supported by project RESCUE funded from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 722325.

REFERENCES

- [1] B. Tabacaru, M. Chaari, W. Ecker, T. Kruse, and C. Novello, "Speeding up safety verification by fault abstraction and simulation to transaction level," in *2016 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Sep. 2016, pp. 1–6.
- [2] R. Leveugle, "Early analysis of fault-based attack effects in secure circuits," *IEEE Transactions on Computers*, vol. 56, no. 10, pp. 1431–1434, Oct 2007.
- [3] N. Wiersma and R. Pareja, "Safety != security: On the resilience of asil-d certified microcontrollers against fault injection attacks," in *2017 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, Sep. 2017, pp. 9–16.
- [4] H. Ziade, R. Ayoubi, and R. Velazco. A Survey on Fault Injection Techniques. (2003). [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.167.966>
- [5] C. Kumar, F. Maamari, K. Vittal, W. Pradeep, R. Tiwari, and S. Ravi, "Methodology for early rtl testability and coverage analysis and its application to industrial designs," in *2014 IEEE 23rd Asian Test Symposium*, Nov 2014, pp. 125–130.
- [6] S. Mirkhani and J. A. Abraham, "Eagle: A regression model for fault coverage estimation using a simulation based metric," in *2014 International Test Conference*, Oct 2014, pp. 1–10.
- [7] A. Papadimitriou, D. Hély, V. Beroulle, P. Maistri, and R. Leveugle, "A multiple fault injection methodology based on cone partitioning towards rtl modeling of laser attacks," in *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2014, pp. 1–4.

- [8] P. Vanhauwaert, P. Maistri, R. Leveugle, A. Papadimitriou, D. Hély, and V. Beroulle, "On error models for rtl security evaluations," in *2014 9th IEEE International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS)*, May 2014, pp. 1–6.
- [9] J. Grinschgl, A. Krieg, C. Steger, R. Weiss, H. Bock, and J. Haid, "Modular fault injector for multiple fault dependability and security evaluations," in *2011 14th Euromicro Conference on Digital System Design*, Aug 2011, pp. 550–557.
- [10] D. Rossi, M. Omana, F. Toma, and C. Metra, "Multiple transient faults in logic: an issue for next generation ics?" in *20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'05)*, Oct 2005, pp. 352–360.
- [11] N. Miskov-Zivanov and D. Marculescu, "Mars-c: modeling and reduction of soft errors in combinational circuits," in *2006 43rd ACM/IEEE Design Automation Conference*, July 2006, pp. 767–772.
- [12] N. Miskov-Zivanov and D. Marculescu, "Multiple transient faults in combinational and sequential circuits: A systematic approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 10, pp. 1614–1627, Oct 2010.
- [13] A. Mochizuki, N. Onizawa, A. Tamakoshi, and T. Hanyu, "Multiple-event transient soft-error gate-level simulator for harsh radiation environments," in *TENCON 2015 - 2015 IEEE Region 10 Conference*, Nov 2015, pp. 1–6.
- [14] F. Augusto da Silva, A. C. Bagbaba, S. Hamdioui, and C. Sauer, "Combining fault analysis technologies for iso26262 functional safety verification," in *2019 IEEE 28th Asian Test Symposium (ATS)*, Dec 2019, pp. 129–1295.
- [15] F. A. d. Silva, A. C. Bagbaba, S. Hamdioui, and C. Sauer, "Efficient methodology for iso26262 functional safety verification," in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, July 2019, pp. 255–256.
- [16] G. Cabodi and M. Murciano, "Bdd-based hardware verification," in *Formal Methods for Hardware Verification*, M. Bernardo and A. Cimatti, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 78–107.
- [17] F. Corella, Z. Zhou, X. Song, M. Langevin, and E. Cerny, "Multiway decision graphs for automated hardware verification," 1996.
- [18] A. C. Bagbaba, M. Jenihhin, J. Raik, and C. Sauer, "Accelerating transient fault injection campaigns by using dynamic hdl slicing," in *2019 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*, Oct 2019, pp. 1–7.
- [19] F. Corno, M. S. Reorda, and G. Squillero, "Rt-level itc'99 benchmarks and first atpg results," *IEEE Design Test of Computers*, vol. 17, no. 3, pp. 44–53, July 2000.
- [20] R. Leveugle, A. Calvez, P. Maistri, and P. Vanhauwaert, "Statistical fault injection: Quantified error and confidence," in *2009 Design, Automation Test in Europe Conference Exhibition*, April 2009, pp. 502–506.