POLITECNICO DI TORINO
Repository ISTITUZIONALE

Towards an automatic approach for hardware verification according to ISO 26262 functional safety standard

Original
Towards an automatic approach for hardware verification according to ISO 26262 functional safety standard / Sini, J., Sonza Reorda, M., Violante, M., Sarson, P.. - ELETTRONICO. - (2018). (24th IEEE International Symposium on On-Line Testing and Robust System Design (IOLTS) Platja d'Aro, Costa Brava (ESP) July 2-4, 2018) [10.1109/IOLTS.2018.8474083].

Availability:
This version is available at: 11583/2712585 since: 2019-06-20T16:25:57Z

Publisher:
IEEE

Published
DOI:10.1109/IOLTS.2018.8474083

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2018 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 collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

18 November 2026

Towards an automatic approach for hardware verification according to ISO 26262 functional safety standard

J. Sini, M. Sonza Reorda, M. Violante
Politecnico di Torino

P. Sarson
Dialog Semiconductor

Abstract—**The Failure Mode, Effect and Diagnostic Analysis (FMEDA) is a technique widely adopted by automotive industry to assess the level of reliability of hardware designs. Although very useful, it has the problem of taking a long time to complete and requires experts with extensive knowledge of the circuit under consideration.**
In this paper, it is presented a comparison between the analysis results obtained from an automatic tool developed by the authors with respect to the ones obtained by hand from a team of experts, followed by a critical review of the strengths and weaknesses, about the rules for automatic classification of the faults effects.

Keywords—. ***Circuit faults; Hardware; Software; Microcontrollers; Safety; Automotive electronics; Embedded systems; failure analysis; ISO 26262 standard; Reliability***

I. INTRODUCTION

Modern vehicles embed a significant number of ECUs, responsible for almost all the vehicle functions. Since some of them perform safety-related functionalities, a strict development process is required. The international standard ISO 26262 [1] contains mandatory guidelines in order to develop these kinds of devices. Based on the criticality level of the provided functions, it is necessary to guarantee some reliability levels. Various techniques there exist to compute this level for a design. The most commonly adopted is the Failure Mode and Effect Analysis (FMEA). If the system itself contains fault detection mechanism, the technique, in this case called Failure modes, effects, and diagnostic analysis (FMEDA) [2], allows assessing how these mechanisms can improve the overall reliability of the item.

This paper presents an industrial case of FMEDA, for which its assessment was performed twice: the first time by a team of experts, by inspecting the design and assessing the criticality of each failure, the second time by an automatic tool able to simulate and classify, on the basis of the results obtained, the behavior of the faulty circuit. The tool has been developed by the authors. It was firstly presented in [3] and, in a version completely integrated into the MATLAB/Simulink environment, in [4]. Then, a comparison between the analysis results obtained by the automatic tool with respect to the ones obtained by hand from experts was carried out. The main focus of this analysis is a critical review of the strengths and weaknesses, about the rules for automatic classification of the faults effects.

II. BACKGROUND

ISO 26262 design process starts with a hazard analysis and risk assessment activity, to be done at the item level. As the output of this activity, designers obtain the item Automotive Safety Integrity Level (ASIL). According to the obtained ASIL level, the standard prescribes a number of techniques to be applied. In particular, Failure Mode and Effect Analysis (FMEA) is strongly recommended for ASIL C and D items.

ISO26262-part 5 prescribes to verify hardware designs by:

- examining the sources of possible failures;
- determining the effects of these failures at the item level.

Based on the ASIL level of the item, at the end of the design verification phase, engineers have to provide robustness evidence about the designed item. The robustness of a design has to be summarized by three metrics: *random hardware fault* rhf, *single point fault metric* $spfm$, and *latent fault metric* lfm. ISO 26262 prescribes the acceptable range of these metrics for each ASIL level.

To compute these three metrics, we can define these rates for a given fault f [3].

- *Failure rate*, λ^f: is the occurrence rate of the fault f expressed as Failure-In-Time (FIT), that is the number of expected failures in a billion hours;
- *Safe Detected (SD) rate*, λ_{SD}^f: defined as the rate of faults that are detected through the functional safety mechanisms the item embeds; even if undetected, these faults could not provoke any harm to the item users;
- *Safe Undetected (SU) rate*, λ_{SU}^f: defined as the rate of faults that are not detected through any functional safety mechanism the item embeds, and that do not provoke any harm to the item users;
- *Dangerous Detected (DD) rate*, λ_{DD}^f: defined as the rate of faults that are detected through the functional safety mechanisms the system embeds; if undetected, these faults are able to provoke harms to the item users;
- *Dangerous Undetected (DU) rate*, λ_{DU}^f: defined as the rate of faults that are not detected through any of the functional safety mechanism the items embed, and able to provoke harms to the item users.

From these rates it is possible to define these item-level rates:

- *Item failure rate*: $\lambda = \sum_f \lambda^f$
- *Single point fault rate:* $spf = \sum_f \lambda^f_{DU}$
- *Residual fault rate*: $rf = \sum_f \lambda^f_{DD}$
- *Latent fault rate*, $lf = \sum_f \lambda^f_{SU}$

At this point, starting from the previously obtained rates, it is possible to compute the three metrics requested from the ISO26262 standard:

- *random hardware fault metric*:
$$rhf = spf + rf$$
- *single point fault metric*:
$$spfm = 1 - \frac{spf}{\lambda}$$
- *latent fault metric*:
$$lfm = 1 - \frac{lf}{\lambda}$$

The hardware design verification process ends only when these metrics fulfill the ISO 26262 requirement for the item ASIL.

III. FMEDA Automatic Tool

As said in II, FMEDA design verification and validation methodology is highly recommended by ISO 26262 for ASIL C and ASIL D items. Also, circuit simulation is strongly recommended by this standard. The proposed approach combines these two requirements by using circuit simulations results to classify the effects of the item failures.

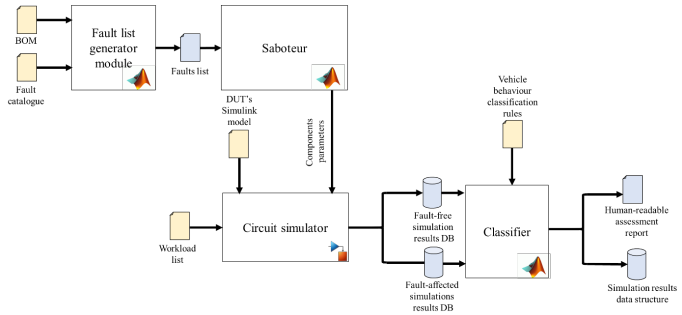


Fig. 1. Tool software architecture. Figure from [4].

As inputs, the tool needs:

- a model of the device under test (DUT), implemented, from the circuit netlist, as a Simulink model using the SimScape toolbox to simulate the electrical components;
- a fault catalog for the components present in the DUT bill of materials (BOM), with the relative FIT values computed during the Reliability, Availability, Maintainability, and Safety (RAMS) analysis;
- failure effect classification rules (as dangerous or safe).

The usual architecture of a microcontroller-based item can be decomposed into three stages: input conditioning, processing, and output conditioning.

For input and output conditioning stages, the fault injection has mainly the purpose to simulate possible failure into the discrete components that implement the analog network. As shown in Fig. 2, the simulation environment prepared for the case study has the same structure of the ideal item, but in this particular set-up without the output conditioning stage. The tool stores four different measurement channels from each simulation: the sensor produced stimuli, the detection channel from the software, the output, in terms of voltage and current, of the conditioning stage.

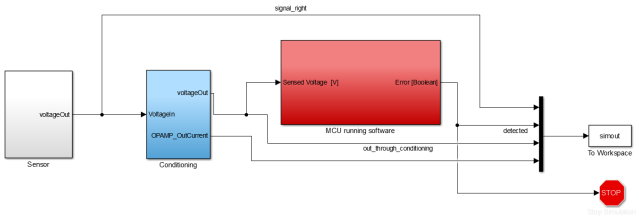


Fig. 2. The simulation environment has the same structure of the item.

The proposed approach has been implemented by means of an FMEDA automation tool.

The tool is fully implemented resorting to the MATLAB/Simulink environment. The item hardware design is modeled through the MathWorks Simulink SimScape toolbox, while the software is modeled as a MATLAB function. The fault list generator, the saboteur, and the classifier modules are implemented as MATLAB script. In this way, it is possible to obtain a unique executable model that capture the relevant characteristics of both hardware and software.

The tool software architecture is shown in Fig. 1. It operates as follows. The fault list generator module takes the BOM of the item, a file containing the components FIT list and the fault catalog. By combining the BOM and the fault catalog, it generates the hardware fault list for the considered item. At this point, thanks to the SimScape model of the circuit, the tool simulates at SPICE-level firstly the item in fault-free (golden) conditions and subsequently by injecting the failures one by one. After each simulation, the classifier compares, by some set of classification rules, the simulation results with the golden ones and assign to each failure the relative effect as safe detected (SD), safe undetected (SU), dangerous detected (DD) and dangerous undetected (DU). At the end of the classification phase, it computes the metrics and generates a human-readable assessment report.

These classification results are stored by means of a MATLAB array of structure. Each structure in the array of results corresponds with a specific workload. The top-level structure has a number of rows equal to the number of the components of the BOM. Each row contains the component name, nominal value, class, and FIT while the second-level struct contained in each row represents the various component failure modes. This sub-structure contains a number of rows equal to the number of failure modes of the component. Each of these rows

contains the failure mode probability, the value injected into the component to simulate the faulty state, and the fault coverage. The fields Safe, Detected and Residual Contribution are filled at run-time by the automatic classifier.

Thanks to this structure the tool is able, if we simulate with more than one workload, to compute the metrics and the assessment for each workload, and to combine all the assessment in a summary report, that allows designers to consider all the worst conditions found in the different workload. For each row, the worst-case failure effect is selected, in descending order of severity, as dangerous undetected (DU), safe undetected (SU), dangerous detected (DD), safe detected (SD). We consider more stringent SU cases than DD ones because, after a fault detection, it is possible for the driver or another ECU to take the appropriate countermeasures to limit possible risks.

A. Fault injection strategies

To inject failures in the simulated circuit, the tool uses various strategies. The simplest one is to modify the nominal value of a component. This approach has been used to simulate failures of the resistors. In those cases where changing the nominal value is not sufficient to properly describe the failure, for example in order to simulate a short circuit between the plate of a capacitor, it is necessary to "instrument" the simulated schematic with saboteur elements. For example, in Fig. 3 it is present a resistance, C1R, that does not exist in the actual schematics but only instruments the capacitor.

B. Failure classification

The system injects one by one the faults in the system, and at the end of the simulations, performs the safe/unsafe classification. This classification can be done by comparing the system outputs with the expected ones, obtained from a set of rules and/or by comparing the system outputs in fault-free (golden condition) with the ones obtained after the failure injection. The circuit object of the experiment deals with measuring the current passing through the load circuit. In the simulation, the load is simulated by a resistance. In the various working conditions, the circuit was first simulated without failures, then injecting faults one by one. The safe/dangerous classification was based on the result produced in conditions of absence of faults: the failure is considered as safe if the output signal produced by the faulty circuitry is within a tolerance of 5% from that of the circuit in fault-free conditions and contemporarily the output current from the OP-AMP is not more than the double of the one in the nominal case, while dangerous in each other condition.

Instead, the detected/undetected classification is obtained directly from the simulation, that contains, as part of the item, the failure detection system. In the case study presented in this work, the feature remains unused since the circuitry, except for the microprocessor, does not embed any failure detection mechanism.

IV. CASE STUDY

The presented case study, provided by a company, is a monitor circuit that has to check if the video interface of an autonomous driving car has the right power consumption. If not, it has to detect the failure. At the end of the risk assessment phase, the device was classified ASIL D. The schematics of the circuit is shown in Fig. 3.

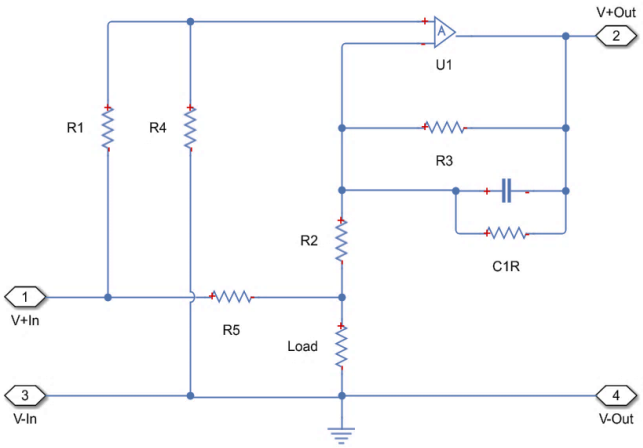


Fig. 3. Case study circuit schematic.

The reliability metrics obtained from both the automatic tool and the expert analysis for the design are reported in Table 1 while the classification results are reported, for each failure mode, in Table 3. For the handmade classification are also reported the classification motivation.

Metric	**Automatic**	**Handmade**
Random hardware fault metric	$0.99 \cdot 10^{-8}$	$1.01 \cdot 10^{-8}$
Single point fault rate	22 %	19 %
Latent fault rate	78%	76 %

Table 1. FMEDA assessment result comparison between the handmade and the automatically performed one.

Automatic / Handmade	**SD**	**SU**	**DD**	**DU**
SD	0	0	0	0
SU	0	2 (equal) [11 %]	0	5 (worst) [28 %]
DD	0	0	0	0
DU	0	2 (better) [11 %]	0	9 (equal) [50 %]

Table 2. Comparison between the failure classifications obtained by the automatic tool and the experts.

From the assessment point of view, we obtain that on 18 different failure modes (the 5 one related to the two integrated circuits U1 and U2 are assessed by hand in any case) for 11 of them we obtain the same assessments. On the other hand, to better analyze the disagreement cases, the result comparison is reported in Table 2. This contingency table shows in horizontal the classification obtained from the automatic tool while in vertical it lists the classifications obtained by the experts. There are 2 cases where the automatic tool has classified the fault as SU while the experts as DU, and 5 cases for the vice versa. On the diagonal, instead, are reported the cases in which the classification obtained in both classifications was the same.

The automatic classifier agrees with the experts' classifications in 61 % of times. We can consider also that the tool considers as unsafe condition conditions considered safe by the expert in 28 % of cases (the tool is more conservative) and as safe condition recognized as unsafe by the expert in 11 % percent of cases (these cases are more problematic since these classifications are less conservative). In particular, by considering the motivation provided by the expert to assign the DU classifications, we have that:

- 1 (6 %) of cases in which the tool, by only simulating the circuit, was not able to detect that the circuit is no more able to work properly;
- 1 (6 %) cases in which the simulation is not able to reach those conditions in which the circuit generates a current readout lower than the real one.

Starting from the data presented above, it is possible to say that there is room to improve the faults classification rules since the only comparison with the outputs of the fault-free simulation with the ones with the faults was found in 39% of the cases in disagreement with the experts.

V. CONCLUSIONS

The main focus of this article has been a comparison between the FMEDA results assessments obtained from a team of experts and from an automatic tool. The classification capability of the simulation-based approach is based on choosing a suitable tolerance level between fault-free and faulty conditions outputs. In this industrial case, the automatic tool and the team of experts agreed in 61% of the classifications, while: in 28% of cases the tool classified as DU an SU failure and in 11% of cases the vice versa, i.e. DU failures were classified like SU ones.

As a future perspective will be useful to generate (even in an automatic manner) and compare each other more complex sets of classification rules. It would be interesting to repeat this type of comparison between experts and automatic tool on a circuit that integrates fault detection mechanisms, in order to verify the goodness of the simulation-based approach also in this context.

REFERENCES

[1] ISO 26262-10:2012, Road vehicles - Functional safety, 2011

[2] W. M. Goble, "Control Systems Safety Evaluation and Reliability", third edition, International Society of Automation, ISBN: 978-1-934394-80-9

[3] Bagalini, E.; Sini, J.; Sonza Reorda, M; Violante, M.; Klimesch H.; Sarson, P.; "An automatic approach to performing the verification of hardware designs according to the ISO 26262 functional safety standard", 18th IEEE Latin America Test Symposium, Bogota, Colombia, 2017

[4] J. Sini, M. Violante, "An Automatic Approach to Perform FMEDA Safety Assessment on Hardware Designs", In: IEEE International Symposium on On-Line Testing and Robust System Design, Platja D'Aro, Costa Brava, Spain, 2018

Component	Failure rate [FIT]	Failure mode	Failure mode rate of occurrence	Automatic failure classification	Handmade failure classification	Classification motivation provided by the experts
R1	2.23	open	50.00%	DU	DU	Current value not available
		increase	25.00%	DU	DU	Lower value detected
		decrease	**25.00%**	**DU**	**SU**	**Higher value detected**
R2	2.23	open	50.00%	DU	DU	Lower value detected
		increase	**25.00%**	**DU**	**SU**	**Higher value detected**
		decrease	25.00%	DU	DU	Lower value detected
R3	2.23	**open**	**50.00%**	**SU**	**DU**	**Current value not available**
		increase	**25.00%**	**SU**	**DU**	**Lower value detected**
		decrease	25.00%	SU	SU	Higher value detected
R4	2.23	open	50.00%	DU	DU	Lower value detected
		increase	**25.00%**	**DU**	**SU**	**Higher value detected**
		decrease	25.00%	DU	DU	Lower value detected
R5	2.23	open	50.00%	DU	DU	Current value not available
		increase	25.00%	SU	SU	Higher value detected
		decrease	25.00%	DU	DU	Lower value detected
C1	2.23	**interruption**	**40.00%**	**DU**	**SU**	**Current filter not available**
		short circuit	10.00%	DU	DU	Lower value detected
		decrease	**50.00%**	**DU**	**SU**	**System not available**
U1	7.51	Interruption of any pin	50.00%	DU	DU	Current value not available
		Short of adjacent pins	50.00%	DU	DU	Lower value detected
U2	5.94	Internal calculation error	50.00%	DD	DD	Lower value detected
		Interruption of any pin	25.00%	DU	DU	Lower value detected
		Short of adjacent pins	25.00%	DU	DU	Lower value detected

Table 3. Comparison between the handmade and the automatic assessments. The differences between the two classifications are highlighted.