Investigation of Intermittent Resistive Faults in Digital CMOS Circuits

Hans G. Kerkhoff* and Hassan Ebrahimi†

*Testable Design and Test of Integrated Systems (TDT) Group,
Centre for Telematics and Information Technology,
University of Twente, Enschede, The Netherlands*

**h.g.kerkhoff@utwente.nl*

†*h.ebrahimi@utwente.nl*

Received 21 August 2015

Accepted 28 August 2015

Published 16 October 2015

No fault found (NFF) is a major threat in extremely dependable high-end process node integrated systems, in e.g., avionics. One category of NFFs is the intermittent resistive fault (IRF), often originating from bad (e.g., via- or TSV-based) interconnections. This paper will show the impact of these faults on the behavior of a digital CMOS circuit via simulation. As the occurrence rate of this kind of defects can take e.g., one month, while the duration of the defect can be as short as 50 ns, thus to evoke and detect these faults is a huge scientific challenge. Two methods to detect short pulses induced by IRFs are proposed. To improve the task of maintenance of avionics and reduce the current high debugging costs, an on-chip data logging system with time stamp and stored environmental conditions is introduced. Finally, a hardware implementation of an IRF generator is presented.

Keywords: Dependability; reliability; no faults found; intermittent resistive faults; evoking & detection of intermittent faults.

1. Introduction

The major drawback of the developments in dimensions and complexity of electronic integrated systems ranging from Systems-on-Chip (SoC) up to Printed-Circuit Board (PCB)-based cabinets is a serious reduction in dependability. In these electronic systems, interconnection wiring is heavily dominating the infrastructure and hence potential faults in these parts are extremely important.

One category of interconnection faults which is extremely difficult to detect is the NFF, although they are known under many different names.^{1,2} A specific category of NFFs is intermittent resistive faults (IRFs), characterized by random low-level

*Corresponding author.

resistive (burst) occurrences in time, randomly fixed in locations, but repairable (at least in PCBs and cabinets) if found. By definition, intermittent opens ($R = \infty$) and shorts ($R = 0$) are also included in this class. Several examples of measured IRFs are known, e.g., Ref. 3 and one measured by us is shown in Fig. 1. This category of faults ranks among the highest in terms of occurrence ($> 50\%$) as well as cost is expected to increase in future technology nodes.^{4,13}

The most likely root cause of IRFs is marginal or unstable interconnections. In advanced integrated circuits, there are a high number of interconnection wires and vias. In terms of aging, they can be subject to electro migration, temperature and mechanical stress⁴ causing increased instability. In the emerging 3D chips many very deep and stress-sensitive Through-Silicon Vias (TSVs) are used as interconnection.⁵

In the above cases, intermittent faults from various interconnections could occur. These interconnections can be used to connect transistors at chip level, but also digital as well as analogue/mixed-signal chips in the case of PCBs or as IPs in a (3D-TSV) SoC. For simplicity, we will consider only IRF on the inputs, outputs and power-supply lines of digital CMOS circuits in this paper.

The influence of intermittent faults on digital systems has been studied in many papers.^{6–11} In Ref. 7 the authors studied the impact of intermittent faults on the behavior of a reduced instruction set computing (RISC) microprocessor by using VHDL-based fault injection. The authors of Ref. 8 proposed a metric intermittent vulnerability factor to characterize the vulnerability of microprocessor structures to intermittent faults.

Fig. 1. Example of a measured IRF.

In Refs. 10 and 11, Gil-Tomás *et al.* have generated fault models for intermittent faults at logic and RTL abstraction levels, and injected these faults in the VHDL model of a microcontroller to study its behavior in the presence of intermittent faults.

However, none of the previous work has considered the problem of IRFs in detail. For example, the authors of Ref. 9 modeled an IRF as an intermittent delay fault. In this paper, it is also shown that the intermittent delay fault model is not sufficiently accurate to model an IRF. In Ref. 12 we have proposed a model for IRFs and analyzed the influence of IRFs on analogue systems at the *transistor* level. In this paper, based on our model, the impact of IRFs on a digital system at transistor level will be investigated.

The paper is organized as follows. In Sec. 2, a generic simulation model for IRFs is introduced, based on ours and others experiences in practice. This model is suitable to be introduced in our fault-injection-based CAD environment for evaluating the behavior of digital CMOS circuits under IRFs. Section 3 shows Cadence Virtuoso simulation results of a full adder circuit where inputs and power-supply are subject to IRFs, while the output (logic) behavior as well as the supply current is being evaluated. Section 4 deals with the first challenge to detect IRFs in digital circuits based on the previous observations; also the boundary conditions and an infrastructure are suggested for providing (stored) data to facilitate the debugging of IRFs in a test laboratory. Options to deal with the second big challenge of IRFs, namely, to enhance the probability of evoking IRFs is discussed in Sec. 5. Section 6 presents an IRF generator hardware platform which allows generating a real physical IRF at an interconnection for emulation-based fault injection. The paper is completed with conclusions in Sec. 7.

2. IRFs Model

To model an IRF, the behavior of several loose connections and cold soldering ball grids under harsh environmental conditions were measured. The values of resistances of loose connections and cold solders were accurately measured over time while thermal and vibration effects were generated using external stimulators. An example of a measured IRF is shown in Fig. 1.

Based on this kind of experimental data, a software module has been developed that is able to provide these faults in a Cadence Virtuoso environment. The basic scheme of our IRF injector is shown in Fig. 2. There are six parameters that can be set by determining their minimum and maximum possible values. In addition, any type of (random) distribution such as, for instance, uniform and Gaussian can be chosen according to the specific requirements.

The actual values and distributions applied for our simulations in this paper are listed in Table 1. The start of a high-rate intermittent burst begins with the random start time between 10 ns and 100 ns. Then, a random activation time (T_{active}) is

Fig. 2. Scheme of the IRFs model generator for injection in simulations.

chosen, during which a random resistance value R is assigned to this timeframe. This is the first event of a potential burst of events (maximum set to 20 in this paper).

After that, an inactivation time (T -inactive) between events is randomly generated in which a fault-free situation exists ($R = 10 \Omega$). In the case of a burst (burst length > 1), there is a feedback loop and the same procedure will be followed again (Fig. 2). After the last event of the burst, the safe time is generated, where again there will be a fault-free situation, thereby completing the intermittent fault procedure. To avoid any convergence problems in the simulator, the discontinuities in the burst are not only instantaneous but also incremental by adding a very small capacitance in the model.

In the model, the concept of seeds in random variable generation is used. It allows an easy replication of the same intermittent fault for comparisons during simulations. The model has been implemented in Verilog-A, replacing a normal wire in the net list by one including an IRF. Therefore, analogue, mixed-signal¹² as well as

Table 1. Range of used parameters during fault simulation.

Parameter	Minimum	Maximum	Distribution
Start time	10 ns	100 ns	Uniform
Resistance	10 Ω	100 k Ω	Uniform
T-active	0.6 ns	10 ns	Uniform
T-inactive	0.3 ns	0.6 ns	Uniform
Burst length	1	20	Uniform
Safe time	1 ns	(Years)	Uniform

digital circuits can be evaluated at the transistor level. The next section will show some results of IRF simulation with respect to digital transistor circuits.

3. IRF Simulations of Digital CMOS Circuits

In order to evaluate the influence of the IRFs on the electrical behavior of digital circuits, a well-known concept of fault injection and simulation-based fault injection⁸⁻¹⁰ are used.

As a simple example, a static CMOS full adder circuit in 45 nm CMOS technology has been used. The circuit operates at a clock frequency of 3.3 GHz. The logic scheme is depicted in Fig. 3(a); its transistor implementation is provided at a lower hierarchical level. The single (statistically) generated IRF in carry input C_{in} is shown in Fig. 3(b). It shows a burst of 20 changes in resistance from 1 k to 100 k ohm during 100 ns.

The flip-flop clock Clk, inputs (X , Y , C_{in}) and outputs (Sum, Sum-L, Cout, Cout-L) voltages and dynamic supply current Iddt are shown in Fig. 3(c).

Since the number of inputs is limited, a test pattern consisting of all possible input combinations is used. Therefore, the waveforms at the output will be repeated after eight clock pulses in case of fault-free behavior. This allows readers to detect any disturbance at outputs very easily by only looking at the waveforms.

As it can be seen in Fig. 3(c), the carry input C_{in} has been disturbed by the IRF (a burst of 20 pulses), but only in few cases (shown by stars) this has translated into an incorrect logic output after the flip-flops. This is because digital CMOS circuits are very robust with regard to disturbances; or in terms of testing, most of the injected faults are being masked.

However, in the analogue domain, dynamic power current (Iddt) disturbances can be noticed (bottom Fig. 3(c)). This current disturbance is depicted in more detail in Fig. 4. The shaded area indicates Iddt increases of about 200 μA if an IRF occurs.

To investigate the relationship between the amount of Iddt disturbance and logic failure, the amount of Iddt in the fault-free and faulty situation was measured. The amount of Iddt increment due to IRF on C_{in} is shown in Fig. 3(b). Stars indicate if a resistive fault has caused a logic failure. As can be seen, there is not a direct relationship between the amount of Iddt increment and logic failure in the circuit. For example, during 7.11 ns to 12.15 ns the maximum amount Iddt increment is 333 μA . However, the circuit has not failed during this period, but it has failed at 25 ns, 33 ns and 53 ns where the Iddt current increment is 219 μA , 258 μA and 220 μA , respectively. These values can be detected by existing Iddt monitors (with a resolution of 2 μA).³ Embedded current instruments for IRFs will be the subject of another publication.

It is obvious that the relation of the used clock frequency in the system and the value of the resistive fault are of crucial importance. Hence, it is no surprise that in the case of the largest resistance, the biggest problems would be anticipated.

(a)

(b)

(c)

Fig. 3. Simulation of a full adder under influence of an IRF in input C_{in} . (a) logic scheme of a full adder in the Nangate 45 nm technology library, (b) used IRF input pattern at input C_{in} . Stars show under which condition a logic failure occurred, and the value on top of each resistance pulse is the amount of I_{ddt} current increment and (c) the simulated functional output voltages of the full adder, including the dynamic power current. Stars indicate functional logic failures.

Fig. 4. Detail of the simulation of a full adder under influence of an IRF in input Cin . The shaded area indicates the difference between dynamic power current in the fault-free and faulty case if the input voltage of Cin is distorted.

Also, the other two inputs (X and Y) of the full adder have been evaluated, but the one shown previously (Cin) shows the largest impact on outputs and dynamic current. As another experiment, also an IRF has been inserted in the V_{dd} line; the generated IRF at V_{dd} is identical to Fig. 3(b). The resulting outputs are shown in Fig. 5(a), while a detail of V_{dd} and I_{ddt} are shown in Fig. 5(b).

It can be concluded from the above simulation results that the impact of an IRF on V_{dd} (as well as with respect to Ground) is quite large. From these results, it

Fig. 5. (a) The resulting output caused by an IRF at the V_{dd} of the full adder and (b) detail of simulated V_{dd} voltage of the full adder, including the I_{ddt} current.

Fig. 5. (Continued)

becomes clear that analogue data is the best way of monitoring IRFs in digital circuits, due to avoiding the logic masking of IRFs.

4. Detection and Data Logging of IRFs

To tackle IRFs, there are two main problems to deal with in a practical situation.^{14–16} The first is the moment of occurrence, which can be any time in the future; major issue is that in worst cases it occurs rarely, and hence a very long test time is required or an online test solution has to be found. This will be discussed in detail in Sec. 5.

The second problem is that the duration of the occurrence can be very short, and often comes in bursts. This requires detection of very short events, and in terms of a pure digital approach, often very high sample rates or accurate small-delay control.

As previously discussed, the issue of NFF is a major source of maintenance costs, especially in avionics. The only provided information is that the digital system has failed in operation, while during testing in the laboratory the system behaves correctly. This fact suggests more likely that the conditions in the laboratory are not identical at the time the fault occurred. Often, the exact power-supply and environmental conditions (temperature, vibration)⁴ are unknown. Providing this data to the test laboratory would dramatically increase the probability of failure detection during the testing.

In the following, these crucial issues i.e., detection of short-duration pulse and data logging will be dealt with in more detail.

4.1. Detection of bursts of long-duration resistive pulses

Whether or not an IRF manifests itself as logic fault, or is masked, depends on a number of factors. Let us assume rising-edge clocks which are used for sampling in

the (state-storing) flip-flops. Important factors are the clock frequency being used in the circuit, and the momentary resistive R value of the IRF pulse(s). If the clock frequency is much faster than the pulse, and the RC (rise & fall) time of the IRF is much larger than the clock, logic faults are likely to appear. These long-duration resistive faults may cause late transition on the output and are more likely to be detected. A number of circuits have been suggested in the past to handle related tasks, like late-transition detection.¹⁷

Bursts with extremely small inactive times and long-duration active times (as compared to the clock duration) will behave similarly as having a single long active time. Sometimes these IRFs are referred to as semi-IRFs.

4.2. Detection of bursts of short-duration resistive pulses

If the resistive pulses are much smaller than the clock, the situation is more difficult. Depending on the location in time of an IRF pulse/burst (four cases), no logic faults will occur. Only in the case of the existence of a pulse during the rising clock edge, there could be a very small chance that a logic fault would occur. Likewise, in the case the duration is smaller than the clock, the RC time will not affect the circuit. Therefore, it is not possible to detect such an IRF via a logic fault in practice. This makes the IRF detection quite difficult.

The simulation results in Sec. 3 show that using analogue data is the best way of monitoring IRFs in digital circuits. The reason is that short-duration resistive pulses slightly increase transition time of the output signal that is, in practice, not detectable by existing late-transition detection sensors. However, they noticeably increase the switching current of circuit which is detectable by existing current sensors.³

By using these current sensors, undetectable short-duration resistive pulses can be converted to detectable short-duration voltage pulses. We have investigated the possibilities of detecting short-duration voltage pulses using synchronous and asynchronous methods.

In the synchronous method, one possibility is to use a number of flip-flops receiving the same input in parallel, but with different clock delays. The flip-flop outputs are all connected to a multi-input OR gate, basically detecting any output differences. The desired delay is constructed by using inverters.

The circuit in Fig. 6 has been simulated in Cadence Virtuoso with the Nangate 45 nm technology library. In the case the clock frequency is 3 GHz, this circuit can detect all pulses larger than 120 ps. The results are shown in Fig. 7. The input signal (A) shows some IRF-derived pulses, shorter than the clock period. The output voltage (Out) shows that all the pulses are being detected. Experiments have shown that the minimum duration of the pulses is important and the position of the pulses does not influence the result.

Another possible method to detect short-duration voltage pulses is to use an asynchronous counter (Fig. 8(a)). In this method, the counter is made of several

Fig. 6. A synchronous circuit to detect a burst of short-duration pulses.

Fig. 7. The simulation results of the synchronous pulse detection circuit. Input A is the IRF-related input pulses; the second line is the clock (1 GHz) and finally, Out is the resulting output voltage.

cascaded flip-flops. Each flip-flop receives its clock from the previous flip-flop's output, except the first flip-flop in which case its clock is connected to a data line (e.g., a current sensor output). The counter increments by one each time a low-to-high transition occurs on its input line. Basically, an asynchronous counter can capture any pulses with duration larger than the propagation delay of a flip-flop. Fig. 8(b) depicts, how this method can detect IRFs. A D-type flip-flop from the Nangate 45 nm technology library was used with the propagation time about 50 ps. As it can be seen, there are 16 short pulses in the range [50 ps, 70 ps] in line A. All

Fig. 8. (a) An asynchronous circuit to detect a burst of short-duration pulses. (b) The simulation results of the asynchronous pulse detection circuit. A is the IRF-related input pulse and Q1 to Q5 are output voltages of flip-flops.

short pulses are captured by the proposed circuit. This circuit is also able to detect IRFs by detecting a burst of short pulses.

From the above, it can be concluded that both synchronous as well as asynchronous methods can be used to detect short-duration pulses induced by IRFs. In comparison with the synchronous method, the asynchronous method can detect pulses with shorter duration. It is also simpler and requires less hardware for implementation. However, to detect a burst occurrence, the asynchronous method considers only the number of transitions and not the burst duration, whereas the synchronous method is more accurate because it can take into account the number of transitions in a burst as well as its duration.

The burst length obtained from synchronous or asynchronous method can be used by data logger to distinguish between transient and intermittent faults.

4.3. Data logging, time stamps and environmental conditions

In this case an IRF is being detected by a transition detection circuit (TDC, Fig. 9), a flag is raised to enable the measurement of the most local temperature, e.g., via a diode-based or our ring-oscillator embedded instrument¹⁸ possibly using the IJTAG standard IEEE 1687. At almost the same time, the local power-supply Vdd is

Fig. 9. Possible setup for a (partly on-chip) data-logging system to enable greatly improved debugging facilities for NFFs in the test lab.

determined, e.g., also via our multi-purpose ring oscillator.¹⁸ Several high risk regions, e.g., many serial vias or TSVs, can be determined by inductive fault analysis (IFA) techniques.¹⁹ Any vibration could be monitored via existing MEMS-based sensors,²⁰ not necessarily locally integrated on-chip or even on top of the chip but, for example, located on the SoC-housing PCB. An internal clock in the SoC provides a timestamp of the IRF event. All this data can be loaded in an on-chip nonvolatile memory, but also an external memory could be used. The global setup of the scheme is shown in Fig. 9. This data can be employed later on by a maintenance engineer for IRF debugging purposes; actually the same infrastructure can be used to make sure by measurement that the same conditions were present at the moment the IRFs appeared.

It is obvious that this suggested infrastructure only makes (economically) sense in highly dependable systems, like avionics. In this case, it could greatly reduce the debug time of NFFs and hence (excessive repair) costs.

5. Enhancing the Probability of Evoking IRFs

As discussed before, the first problem in IRFs is the moment of occurrence, which can be any time in the future; major issue is that in worst cases it occurs rarely, and hence a very long test time is required or an online test solution has to be found. In this section, a potential approach is presented.

5.1. Increasing the probability of evoking IRFs in general

Probably the most difficult part with regard to IRFs is to be able to evoke the event within a reasonable (test) time scale. It is stressed that one has to think rather in

terms of *stochastic* than deterministic occurrence of IRFs. It has been shown that IRFs caused by bad interconnections (lines, vias, TSVs) are sensitive to low and high temperatures, as well as vibration. Actually, the large IFDIS system of Universal Synaptics⁴ uses both. Temperatures between -70°C and 170°C are used, and vibrations with 50 mm displacement at low frequencies (20 Hz–50 Hz) applied to evoke NFFs. It is stressed that this bulky test setup is being used for large electrical modules and the wiring and connectors in between.

Based on this, the following idea has emerged with regard to e.g., interconnections between IPs in a SoC and potential IRFs. Try to locally heat up the temperature on-chip near a layout-based high-risk IRF area (using standard IFA techniques¹⁹) and by removing the heat subsequently (cool down after heating). By repeating this, a kind of temperature cycling is emulated, also causing mechanical stress (low vibration frequency). Both will increase the probability that e.g., cracks in vias and TSV will deteriorate momentary.

The question remains to what extent this can be emulated on-chip. It is reminded that the structures in an SoC are extremely small, and hence not the same temperature and vibration requirements hold as in the case of IFDIS.⁴

5.2. Increasing the probability of evoking of IRFs at chip level

We have investigated the possibilities of the above using a practical industrial example. We used a 90 nm CMOS heterogeneous multi-processor SoC. It uses an LFBGA 233 package, having a thermal resistance of junction-to-ambient of $33^{\circ}\text{C}/\text{W}$. Assuming an ambient temperature of 21°C and a maximum power dissipation of $\sim 1000\text{ mW}$, the junction temperature will rise to 53°C in the case of maximum power dissipation. The possible change in temperature of the package is given to be $6^{\circ}\text{C}/\text{s}$, meaning a cool down to ambient would take 5 s. This results in a very low mechanical vibration frequency of 0.2 Hz. Compared to the large IFDIS system 4, the maximum temperature change is only 30%, while the mechanical vibration frequency is a factor 10 lower. Using the thermal expansion parameter of silicon to be $2.6 \times 10^{-6} \text{ }^{\circ}\text{C}^{-1}$ and a TSV of $1\text{ }\mu\text{m}$ deep with a crack, around $0.1 \times 10^{-3} \text{ }\mu\text{m}$ change in length would result. How this would affect the actual resistance of a crack/void has to be shown empirically.

However, the above calculation is based on the global temperature changes of the whole chip. Locally on-chip, the temperatures can be monumentally much higher very near the source of power consumption, as is the mechanical vibration frequency resulting from rapid temperature changes; temperature variations have shown to follow 10 kHz changes at a few μm distances.²¹ Our research in this area is still ongoing. An interesting paper²² relates to a somewhat similar issue, basically trying to emulate a burn-in infrastructure on-chip.

6. Emulation-Based IRF Injection Generator Implemented in Hardware

In Sec. 3, a simulation-based intermittent fault injection has been used to evaluate IRF effect on digital systems. However, simulation-based fault injection is time-consuming, particularly in the case of IRFs. Emulation-based fault injection²³ is an alternative solution for accelerating fault injection. This technique also allows studying the behavior of a circuit in real time, and also large (PCB-based) systems can be validated in real-time.²⁴

An automated hardware/software platform for emulation-based IRF injection has been developed. The proposed emulation-based platform is composed of a host computer, an FPGA and a home-made PCB board which is shown in Fig. 10. The host computer executes a MATLAB script which generates a burst of random resistance values based on the model which was described in Sec. 2. The generated burst sequence can be transferred to the FPGA board by a serial communication link. The FPGA is responsible to synchronize and generate the control signals for the on-board programmable switches and potentiometers on the PCB board. The desired resistance range is provided by a combination of fast digital potentiometers and fixed range resistors. An example of a generated IRF by our generator is shown in Fig. 11.

There are two waveforms in this figure; one is a resistance sequence measured from the PCB board and the other is the expected sequence generated by the script. As it can be seen, the measured resistance sequence approximately follows the expected sequence with an inevitable error because of the parasitic capacitances of the existing

Fig. 10. Photograph of the IRF generator, showing FPGA under control of an USB port and the actual pattern generator.

Fig. 11. An example of a generated IRF showing the ideal case and actually measured case.

switches and potentiometers. In future work, the proposed IRF generator will be used to evaluate the influence of IRFs on analogue and digital systems in real time.

7. Conclusions

In this paper, the effects of a special category of NFF, being single IRFs have been discussed. IRFs result from interconnection flaws which are random in time, but not in location(s). They are extremely difficult to detect and diagnose and are hence very costly. Future processing nodes are likely to encounter NFFs much more than nowadays. A simulation fault-injection model for IRFs has been developed, based on measurement experiences. The parameters in this simulation fault-injection model can be extended and changed at will. A simple digital example has been investigated; at several locations (inputs and power lines), this type of fault was introduced and its outputs and currents investigated. Especially, IRFs in the power line have a significant influence on logic and analog Iddt currents. Detection of IRF-related pulses has been discussed and a possible solution for the difficult case of very short pulses provided and validated. Monitoring output voltages and power-supply currents, together with power-supply and temperature/vibration data can log potential anomalies accompanied by a timestamp and subsequently stored in a log memory to support debug later on. In terms of scalability of the approach, a (top) ranking of the probabilities of IRF locations is suggested via existing IFA techniques. In addition, an innovative approach to emulate on-chip elevated temperature and even mechanical stress (vibration) could help to enhance the probability of IRF evoking. Finally, a hardware implementation of a fully programmable IRF generator has been designed and measured, which can help to detect weak spots in digital as well as analogue integrated circuits with respect to this important category of faults.

Acknowledgments

This research was partly carried out within the FP7 BASTION project and partly within the ENIAC ELESIS project, both financed by the European Committee (EC) and the Netherlands Enterprise Agency (RVO).

The authors would like to acknowledge the contributions of Marc Brakels and Dirk-Jan van de Sanden with regards to the IRF generator.

References

1. S. Davidson, Towards and understanding of no trouble found devices, *Proc. VLSI Test Symp. (VTS)*, Palm Springs, California, USA (2005), pp. 147–152.
2. Accenture Report, Big trouble with ‘no trouble found’ returns (2008), http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture>Returns_Repairs.pdf.
3. Ridgetop Group Inc., SJ BIST, White Paper Presentation, 2013.
4. K. Anderson, Intermittent fault detection & isolation system IFDIS, White Paper Synaptics, *13th CTMA Symp*, Utah, USA, March 2012.
5. E. J. Marinessen, Testing TSV-based three-dimensional stacked ICs, *Proc. Design, Automation and Test in Europe (DATE)*, Dresden, Germany (2010), pp. 1689–1694.
6. J. Gracia-Moran, D. Gil-Tomas, L. Saiz-Adalid, J. Baraza-Calvo and P. Gil-Vicente, Defining a representative and low cost fault model set for intermittent faults in microprocessor buses, *Latin-American Symp on Dependable Computing (LADC)*, Rio de Janeiro, Brazil (2013), pp. 98–103.
7. J. Gracia-Moran, J. C. Baraza-Calvo, D. Gil-Tomas, L. J. Saiz-Adalid and P. J. Gil-Vicente, Effects of intermittent faults on the reliability of a reduced instruction set computing (RISC) microprocessor, *IEEE Trans. on Reliab.* **63** (2014) 144–153.
8. S. Pan, Y. Hu and X. Li, IVF: Characterizing the vulnerability of microprocessor structures to intermittent faults, *IEEE Trans. very Large Scale Integr. (VLSI) Syst.* **20** (2012), pp. 777–790.
9. J. Gracia-Moran *et al.*, Searching representative and low cost fault models for intermittent faults in microcontrollers: A case study, *Proc. Pacific Rim Int. Symp. Dependable Computing (PRISDC)*, Tokyo, Japan (2010), pp. 11–18.
10. D. Gil-Tomás, J. Gracia-Morán, J. Baraza-Calvo, L.-J. Saiz-Adalid and P.-J. Gil-Vicente, Studying the effects of intermittent faults on a microcontroller, *Microelectron. Reliab.* **52** (2012) 2837–2846.
11. J. Gracia-Morán, D. Gil-Tomás, L. J. Saiz-Adalid, J. C. Baraza and P. J. Gil-Vicente, Experimental validation of a fault tolerant microcomputer system against intermittent faults, *IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, Chicago, USA (2010), pp. 413–418.
12. J. Wan and H. G. Kerkhoff, The influence of no fault found in analogue CMOS circuits, *Proc. IEEE Int. Mixed-Signal Test Workshop (IMSTW)*, Porto Alegre, Brazil (2014), pp. 1–6.
13. B. A. Sorensen, C. S. Chambers and K. Andersen, The right stuff for aging electronics, intermittence/no fault found, white paper Synaptics Corporation, 2010.
14. H. Qi, S. Ganesan and M. Pecht, No fault found and intermittent failures in electronic products, *Microelectron. Reliab.* **48** (2008) 663–671.
15. F. Hapke *et al.*, Cell-aware experiences in a high quality automotive test suite, *Proc. European Test Symp. (ETS)*, Paderborn, Germany, May 2014, pp. 1–6.
16. C. Constantinescu, Intermittent faults and effects on reliability of integrated circuits, *Proc. Reliability and Maintainability Symp. (RAMS)*, Las Vegas, USA (2008), pp. 370–374.

17. A. Amouri and M. Tahoori, A low-cost sensor for aging and location for late transitions detection in modern FPGAs, *Int. Conf. Field Programmable Logic and Applications*, (2011), Chania, Greece, pp. 329–335.
18. Y. Zhao and H. G. Kerkhoff, Design of an embedded health monitoring infrastructure for accessing multi-processor SoC degradation, *Proc. Euromicro Conf. Digital System Design (DSD)*, Verona, Italy (2014), pp. 154–160.
19. A. Jee and F. J. Ferguson, Carafe: An inductive fault analysis tool for CMOS VLSI circuits, *VLSI Test Symp. (VTS)*, Atlantic City, NJ, USA, 1993, pp. 92–98.
20. Analog Devices, MEMS vibration sensor, (2014) www.analog.com.
21. H. G. Kerkhoff and G. C. M. Meyer, An integrated electro-thermal amplitude detector using the Seebeck effect, *Digest of Technical Papers of the European Solid-State Circuits Conf. (ESSCIRC)*, Southampton, United Kingdom, 1979, pp. 31–33.
22. N. Aghaee, Z. Peng and P. Eles, Temperature-gradient-based burn-in and test scheduling for 3-D stacked ICs, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* (2014), pp. 1–14.
23. L. Entrena, M. Garcia-Valderas, R. Fernandez-Cardenal, A. Lindoso, M. Portela and C. Lopez-Ongil, Soft error sensitivity evaluation of microprocessors by multilevel emulation-based fault injection, *IEEE Trans. Comput.* **61** (2012) 313–322.
24. H. G. Kerkhoff and H. Ebrahimi, Detection of intermittent faults in electronic systems based on the mixed-signal boundary-scan standard, *Proc. of the Asian Quality Electronic Design Conf. (ASQED)*, Kuala Lumpur, Malaysia, August 2015, pp. 1–6.