

# Intermittent Resistive Faults in Digital CMOS Circuits

Hans G. Kerkhoff and H. Ebrahimi

Testable Design and Test of Integrated Systems (TDT) Group  
University of Twente, Centre for Telematics and Information Technology (CTIT)  
Enschede, the Netherlands  
[h.ebrahimi@utwente.nl](mailto:h.ebrahimi@utwente.nl)

**Abstract**— A major threat in extremely dependable high-end process node integrated systems in e.g. avionics are no failures found (NFF). One category of NFFs is the intermittent resistive fault, 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 nanoseconds, to evoke and detect these faults is a huge scientific challenge. An on-chip data logging system with time stamp and stored environmental conditions, along with the detection, will drastically improve the task of maintenance of avionics and reduce the current high debugging costs.

**Keywords**—*Dependability; Reliability; No Faults Found; Intermittent Resistive Faults; Evoking & Detection of Intermittent Faults*

## I. INTRODUCTION

The 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 the above 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 No-Fault-Found (NFF), although they are known under many different names [1, 2]. A specific category of NFFs is intermittent resistive faults (IRF), characterized by random low-level resistive (burst) occurrences in time, randomly fixed in locations, but repairable (at least in PCBs and cabinets) if found. By definition, also intermittent opens ( $R=\infty$ ) and shorts ( $R=0$ ) are included in this class. Several examples of measured intermittent resistive fault are known, e.g. [3], and one measured by us is shown in Figure 1. This category of faults ranks among the highest in terms of occurrence ( $>50\%$ ) as well as cost and is expected to increase in future technology nodes [4].

The most likely root cause of intermittent resistive faults 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 stress [5] causing increased

instability. In the emerging 3D chips many very deep and stress-sensitive Through-Silicon Vias (TSVs) are used as interconnection [6].

In the above cases intermittent faults from interconnect 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 limit their influence to inputs, outputs and power-supply lines of digital CMOS circuits in this paper.

The paper is organized as follows. In section 2, a generic simulation model for intermittent resistive faults (IRF) is introduced, based on our and others experiences in practice. This model is suitable to be introduced in our fault-injection based CAD environment for evaluating the behaviour of digital CMOS circuits under intermittent resistive faults. Section 3 shows Cadence simulation results of a full-adder circuit where inputs and power-supply are subject to IRFs, while the output (logic) behaviour 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 boundary conditions and an infrastructure are suggested for providing

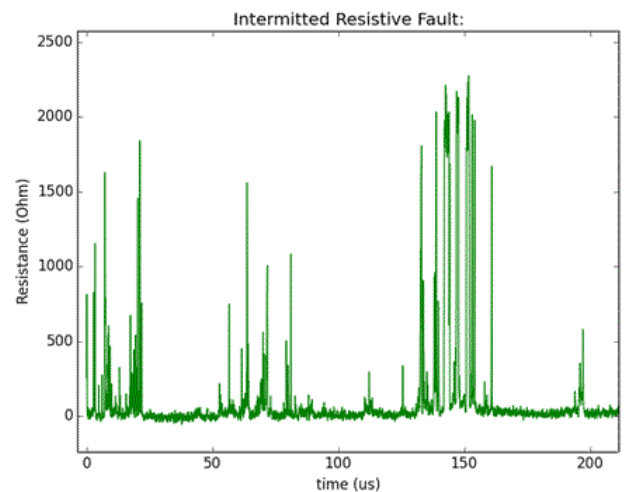


Figure 1. Example of a measured intermittent resistive fault (IRF).

(stored) data to facilitate the debugging of IRFs in a test lab. Options to deal with the second big challenge of IRFs, namely to enhance the probability of evoking IRFs is discussed in section V. The paper is completed with conclusions in section VI.

## II. SIMULATING WITH INTERMITTENT RESISTIVE FAULTS

An example of a measured intermittent resistive fault has been shown in Figure 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 [7]. The basic scheme of our intermittent resistive fault injector is shown in Figure 2. There are six parameters that can be set according to the specific application, with a minimum and maximum value and a certain (random) distribution. The parameters are similar as presented in reference [7]. The actual values and distributions applied for our simulations are listed in Table 1. The start of a high-rate intermittent burst begins with the random start-time generator, with min and max values (Table I) using a uniform distribution; other random distributions (e.g. Gaussian) are also possible. Then, a random activation time ( $T_{active}$ ) is 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 (Figure 2). After the last event of the burst, the safe time is generated, where again there will be a faulty-free situation, thereby completing the intermittent fault procedure. This sometimes long safe time is the major cause of test problems in the case of intermittent faults. For avoiding any convergence problems in the simulator, the discontinuities in the burst are not instantaneous but gradual by adding a very small capacitance  $C$ .

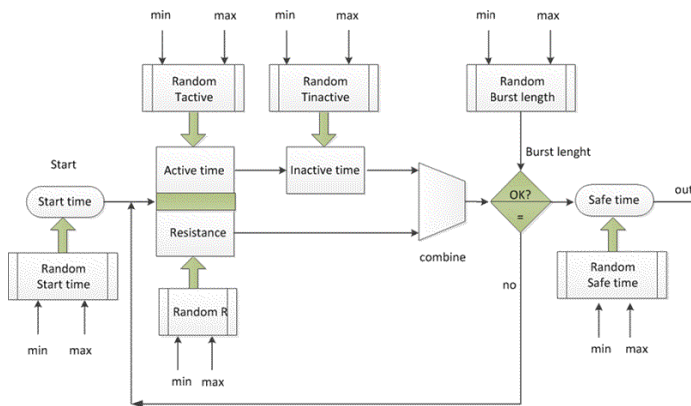


Figure 2. A Verilog-A implementation of intermittent resistive faults [7].

TABLE I. Range of used parameters during fault simulation.

Parameter	Minimum	Maximum	Distribution
Start time	10 ns	100 ns	Uniform
Resistance	10 Ohm	100k $\Omega$	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

The concept of seeds is being used, enabling an easy replication of the same NFFs 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.

A screenshot of our IRF evaluation CAD environment is shown in Figure 3. It shows part of the transistor netlist (background), a (white) window where the intermittent resistive fault parameters can be edited by the reliability test engineer, and the required observation graph(s) of the simulated response of the circuit; the latter can be voltages but also currents. Analogue, mixed-signal [7] as well as digital circuits can be evaluated. The next section will show some results of IRF simulations with respect to digital circuits.

## III. IRF FAULT SIMULATIONS OF DIGITAL CMOS CIRCUITS

We have used the well-known concept of fault injection [7-10] to evaluate the influence of a category of NFFs, being the intermittent resistive faults, on the electrical behaviour of digital

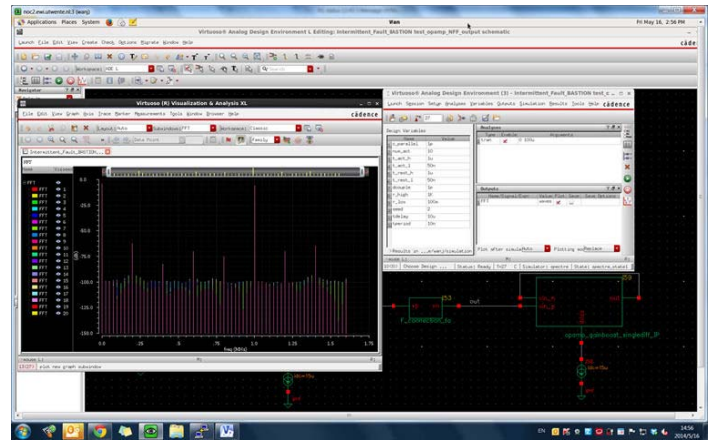
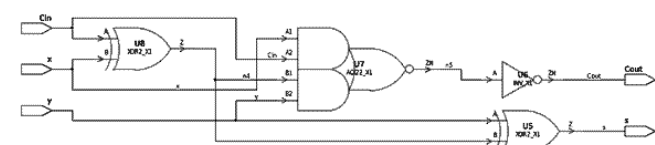
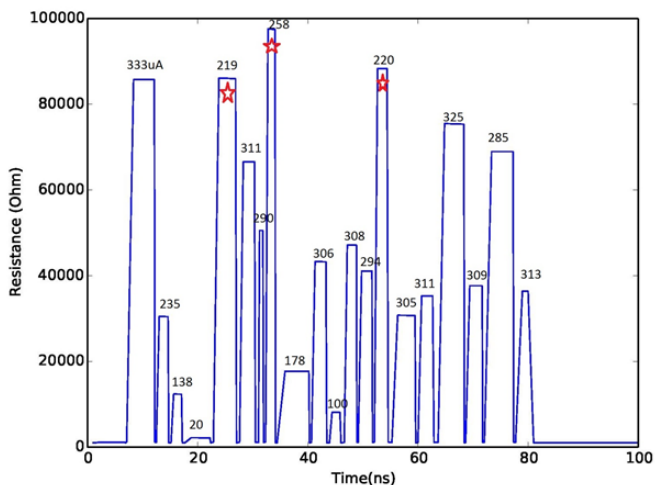


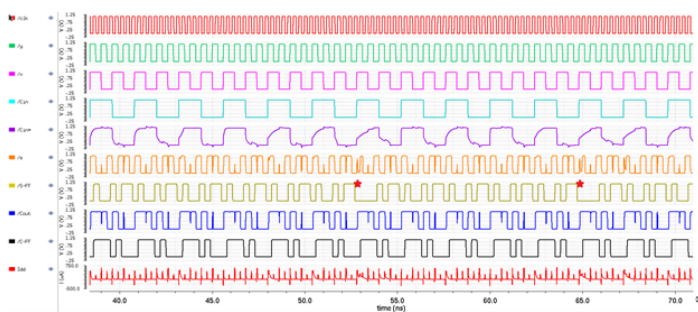
Figure 3. Our Cadence Virtuoso simulation environment for IRFs evaluations.



a)



b)



c)

Figure 4. Simulation of a full adder under influence of an IRF in input Cin. a) logic scheme of a full adder in 45nm NAN CMOS library, b) used IRF input pattern at input Cin (red star shows under which condition a logic error occurred, value on top is the Iddt value), c) the simulated functional output voltages of the full adder, including the dynamic power current. Red stars indicate functional logic failures.

circuits. As a simple example we have used a static CMOS full-adder circuit, its sum and carry outputs latched in D-type flip-flops, in 45nm NAN CMOS technology. The circuit operates at a clock frequency of 3.3GHz (0.3ns). The logic scheme of the combinational part is depicted in Figure 4a; its transistor implementation is provided at a lower hierarchical level. The single (statistically) generated IRF in the carry Cin input is shown in Figure 4b.

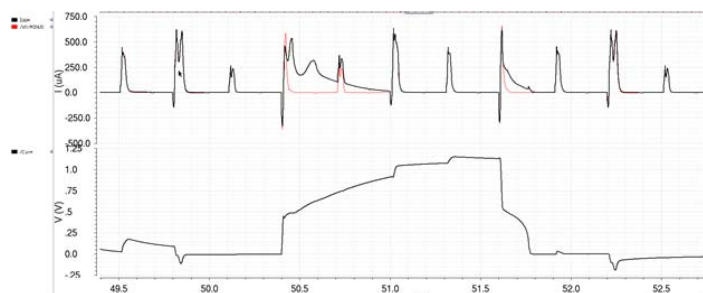


Figure 5. Detail of simulation of a full adder under influence of an IRF in input Cin. Shown is the dynamic power current in the fault-free (black) and faulty (red) situation (top) as is the distorted input voltage of Cin (bottom).

The flip-flop clock clk, inputs (X, Y, Cin) and output (S, S-FF, CO, C-FF) voltages, and dynamic supply current Iddt are shown in Figure 4c.

As can be seen in Figure 4c, the carry input Cin has been disturbed by the IRF (a burst of 20 pulses), but only in two cases (red star) 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 faults (IRF) are being masked. However in the analogue dynamic power current (Iddt), disturbances can be noticed (bottom Figure 4c).

In Figure 5 a detail is depicted of this dynamic power current under these conditions. A difference of around 200 $\mu$ A can be seen in the fault-free (black) and IRF case (red); these values can be detected by existing Iddt monitors (resolution 2 $\mu$ A) [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 can be anticipated. 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 logic output and dynamic current.

As another experiment, also an IRF has been inserted in the Vdd line; the generated IRF at Vdd is identical to Figure 4b. The resulting outputs are shown in Figure 6a, while a detail of Vdd and Iddt are shown in Figure 6b.

As can be concluded from the above simulation results, the impact of an IRF on Vdd (as well as Ground) is quite large. From these results, it becomes clear that analog data is the best way of monitoring IRFs in digital circuits, thus avoiding the logic masking of IRFs.

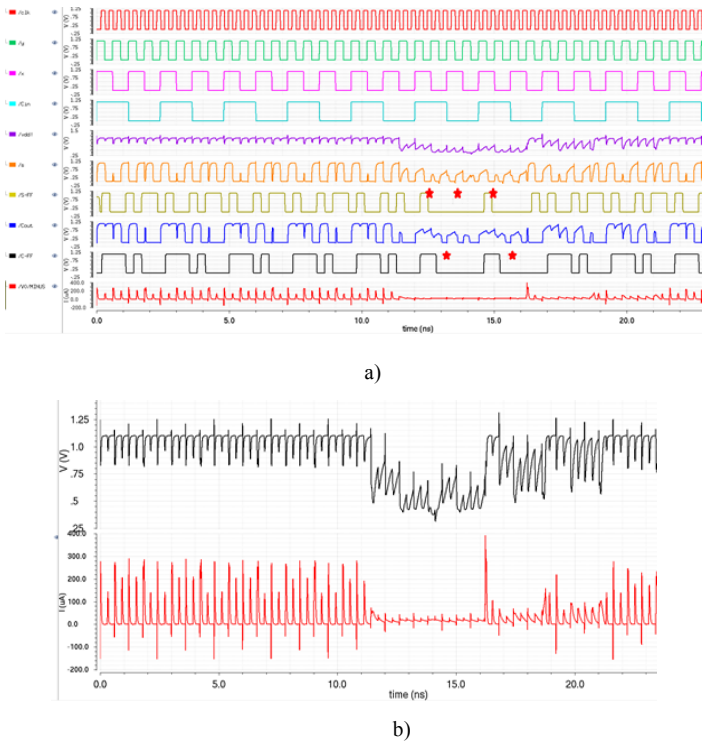


Figure 6. a) The resulting output caused by an IRF at the Vdd of the full adder, and (b) detail of simulated Vdd voltage of the full adder, including the Idid current.

#### IV. DETECTION AND DATALOGGING OF IRFS

There are two main problems with IRFs in a real-life situation [11-13]. 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 treated in detail in section V.

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 no faults found (NFF) is a major source of maintenance costs, especially in avionics. The fact that the only provided information is that the digital system has failed in operation, while during testing in the lab the system behaves well suggests (which explains the expression NFF) that very probably the conditions in the lab are not identical as when the fault occurred. Often, the exact power-supply conditions and environmental conditions (temperature, vibration) [5] are unknown. Providing this data to the test lab would dramatically increase the probability of failure detection during lab testing.

In the next paragraphs these last two issues will be dealt with in more detail.

#### A. Detection of Bursts of Long-Duration Resistive Pulses

Whether or not an IRF manifests itself as a logic fault, or is masked, depends on a number of factors. Lets assume rising-edge clocks are used for sampling in the (state-storing) flip-flops. Important is 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 are in general easy to detect; in the case of a burst of similar pulses (in time and R value) the logic faults will repeat in time. This is actually the situation in Figure 4. 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*-intermittent resistive faults.

#### B. Detection of Bursts of Short-Duration Resistive Pulses

In the case 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. As also the duration is smaller than the clock, the RC time will not affect the circuit. It is hence in practice not possible to detect such an IRF via a *logic* fault. This makes the IRF detection quite difficult.

One possible other option is to detect short pulses of analog values of a voltage (or current) in the circuit as occurring in Figures 5 and 6. A number of circuits have been suggested in the past to handle related tasks, like late transition detection [14].

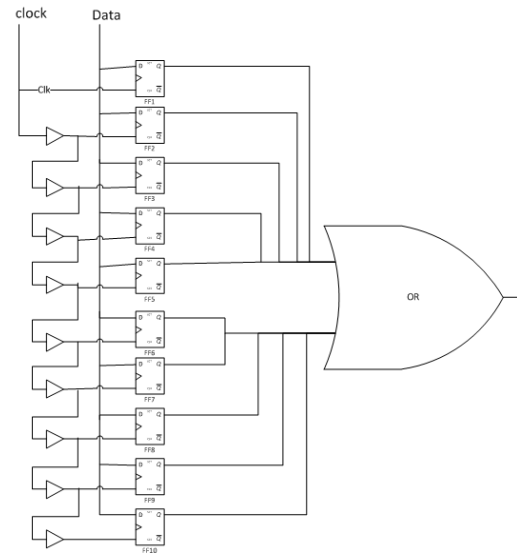


Figure 7. Possible logic circuit to detect very small IRF-related pulses (smaller than the used clock duration).

One possibility is to use a number of flip-flops, say ten, receiving the same input in parallel, but with different clock delays (using the internal clock of the system). The flip-flop outputs are all connected to a multi-input OR gate, basically detecting any output differences. An string of e.g. inverters creates increased clock delays. The minimum delay is the inverter delay (45nm, 4ps), while the maximum delay is roughly equal to the clock period (3GHz, 0.3ns).

The circuit in Figure 7 has been simulated in Cadence with the 45nm CMOS NAN library. The results are shown in Figure 8. The top signal A shows some IRF-derived pulses, shorter than the clock period; simulations have shown that these voltage pulses can indeed result from occurring IRFs. Next, the used clock is shown. The output voltage Out shows that the pulses are all being detected. Experiments have shown that the position of the pulses does not influence the result. The minimum duration of the pulses do play a role. Below 120ps not all pulses are being detected (in the case of 10ps, only 50%).

### C. Data Logging, Time Stamps and Environmental Conditions

The moment an IRF is being detected by the transition detection circuit (TDC, Figure 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 instrument [15] possibly using the IJAG standard IEEE 1687. At almost the same time the local power-supply Vdd is determined, e.g. also via our multi-purpose ring oscillator [15]. Several high risk regions, e.g. many serial vias or TSVs, can be determined by Inductive Fault Analysis Techniques (IFA) [16]. Any vibration could be monitored via existing MEMS-based sensors [17], not necessarily locally integrated on-chip or even on top of the chip, but e.g. located on e.g. 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 NVM, but also an external memory could be used. The global set-up of the scheme is shown in Figure 9. This data can be employed later on by a maintenance

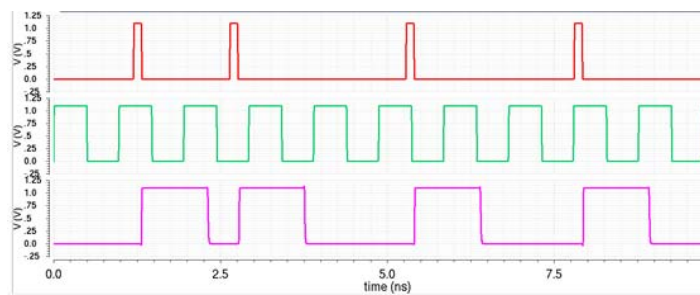


Figure 8. The simulation results of the pulse detection circuit. A (top) are the IRF-related input pulses, the second line is the clock (1GHz) and finally Out (bottom) is the resulting output voltage.

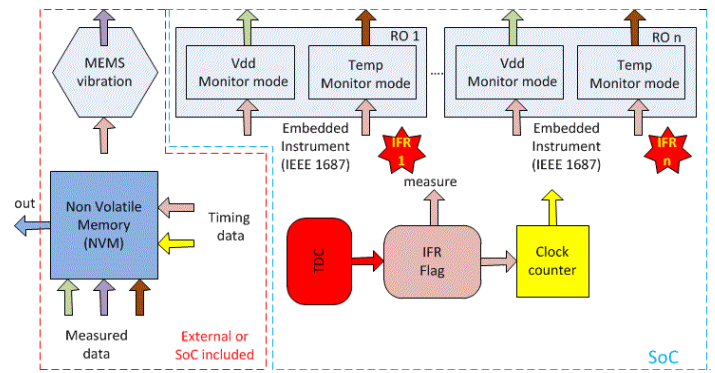


Figure 9. Possible set-up for a (partly on-chip) datalogging system to enable greatly improved debugging facilities for NFFs in the test lab.

engineer for IRF debugging purposes; actually the same infrastructure can be used to be sure by measurement that the same conditions were present at the moment the IRFs appeared.

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

## V. 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.

### A. 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 Synaptics [5] uses both. Temperatures between  $-70^{\circ}\text{C}$  and  $170^{\circ}\text{C}$  are used, and vibrations with 50mm displacement at low frequencies (20Hz – 50Hz) applied to evoke NFFs. It is stressed that this bulky test set-up 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 System-on-Chip and potential IRFs. Try to locally heat up the temperature on-chip near a layout-based high-risk IRF area (using standard IFA techniques [16]) 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 momentarily.

The question remains to what extent this can be emulated on-chip. It is reminded that the structures in a SoC are extremely small, and hence not the same temperature and vibration requirements hold as in the case of IFDIS [5].

#### B. 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 90nm CMOS heterogeneous multi-processor SoC. It uses an LFBGA 233 package, having a thermal resistance of junction-to-ambient of 33°C/W. Assuming an ambient temperature of 21°C and a maximum power dissipation of ~1000mW, 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°C/s, meaning a cool down to ambient would take 5 seconds. This results in a very low mechanical vibration frequency of 0.2 Hz. Compared to the large IFDIS system, the maximum temperature and change is only 30%, while the mechanical vibration frequency is a factor 10 lower. Using the thermal expansion parameter of silicon to be  $2.6 \cdot 10^{-6} \text{ }^\circ\text{C}^{-1}$  and a TSV of 1µm deep with a crack, around  $0.1 \cdot 10^{-3} \mu\text{m}$  change in length would result. How this would affect the actual IRF 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 [18]. Our research is still continued in this area. A very recent interesting paper [19] relates to a somewhat similar issue, basically trying to emulate a burn-in infrastructure on-chip.

## VI. CONCLUSIONS

In this paper, the effects of a special category of No Faults Found, being single intermittent resistive faults 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 intermittent resistive faults 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 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 time stamp 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 inductive fault analysis (IFA) techniques. Finally, an innovative approach to emulate on-chip elevated temperature and even mechanical stress (vibration) could help to enhance the probability of IRF evoking.

## ACKNOWLEDGEMENT

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).

## REFERENCES

- [1] S. Davidson, "Towards and Understanding of No Trouble Found Devices", in Proc. VLSI Test Symposium (VTS), pp. 147-152, 2005.
- [2] Accenture Report, "Big Trouble with 'No Trouble Found' Returns", [http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture\\_Returns\\_Repairs.pdf](http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture_Returns_Repairs.pdf), 2008.
- [3] Ridgetop Group Inc., "SJ BIST", White paper presentation 2013.
- [4] B.A. Sorensen, C.S. Chambers and K. Andersen, "The Right Stuff for Aging Electronics, Intermittence / No Fault Found", White Paper Synaptics Corporation, 2010.
- [5] K. Anderson, "Intermittent Fault Detection & Isolation System (IFDIS)", white paper Synaptics, 13<sup>th</sup> CTMA Symposium, March 2012.
- [6] E.J. Marinessen, "Testing TSV-based Three-Dimensional Stacked ICs", in Design Automation and Test in Europe (DATE), pp. 1689-1694, 2010.
- [7] J. Wan and H.G. Kerkhoff, "The Influence of No Fault Found in Analogue CMOS Circuits", in Proc. IEEE International Mixed-Signal Test Workshop (IMSTW), DOI 10.1109/IMS3TW.2014.6997388, Porto Alegre, Brazil, 2014, pp. 1 - 6.
- [8] J. Gracia-Moran et al., "Searching Representative and Low Cost Fault Models for Intermittent Faults in Microcontrollers: A Case Study", in Proc. Pacific Rim International Symposium on Dependable Computing (PRISDC), pp. 11- 18, 2010.
- [9] S. Pan, Y. Hu and X. Li, "IVF: Characterizing the Vulnerability of Microprocessor structures to Intermittent Faults", IEEE Trans. on VLSI Systems, vol. 20, no. 5, pp. 777-790, 2012.
- [10] D. Gil et al., "Injecting Intermittent Faults for Dependability Validation of Commercial Microcontrollers", in proc. HLDVT, pp. 177-184, 2008.
- [11] H. Qi, S. Ganesan and M. Pecht, "No Fault Found and intermittent failures in electronic products", in Microelectronics Reliability, pp. 663-671, 2008.
- [12] F. Hapke et al., "Cell-aware Experiences in a High Quality Automotive Test Suite", in Proc. European Test Symposium (ETS), Paderborn Germany, pp. 1-6, May 2014 .
- [13] C. Constantinescu, "Intermittent Faults and Effects on Reliability of Integrated Circuits", in Proc. Reliability and Maintainability Symposium (RAMS), pp. 370 - 374, 2008.

- [14] A. Amouri and M. Tahoori, " A Low-Cost Sensor for Aging and Location for Late Transitions Detection in Modern FPGAs", in Internl. Conference on Field Programmable Logic and Applications, pp. 329-335, 2011.
- [15] Y. Zhao and H.G. Kerkhoff, "Design of an Embedded Health Monitoring Infrastructure for Accessing Multi-Processor SoC Degradation", in Proc. 17th Euromicro Conference on Digital System Design (DSD), Verona, Italy, pp. 154-160, 2014.
- [16] A. Jee, F.J. Ferguson, "Carafe: an inductive fault analysis tool for CMOS VLSI circuits", VLSI Test Symposium (VTS), pp. 92 - 98, 1993.
- [17] Analog Devices, " MEMS Vibration Sensor", [www.analog.com](http://www.analog.com), 2014
- [18] H.G. Kerkhoff and G.C.M. Meyer, "An integrated electro-thermal amplitude detector using the Seebeck effect", Digest of Technical Papers of the 5th European Solid-State Circuits Conference, Southampton, United Kingdom, pp. 31-33, September 1979.
- [19] N. Aghaee, Z. Peng, and P. Eles, "Temperature-Gradient-Based Burn-In and Test Scheduling for 3-D Stacked ICs", accepted for IEEE Transactions on VLSI Systems, pp. 1-14, December 2014.