Testing for Intermittent Resistive Faults in CMOS Integrated Systems

Hassan Ebrahimi and Hans G. Kerkhoff
Testable Design and Test of Integrated Systems (TDT) Group
University of Twente, Centre for Telematics and Information Technology (CTIT)
Enschede, the Netherlands
Email: {h.ebrahimi, h.g.kerkhoff}@utwente.nl

***Abstract*—The required dependability of integrated CMOS systems has to be continuously increased because nowadays many applications are safety-critical. Having a good knowledge of potential realistic faults plays an important role in this case. The most difficult and expensive category of faults that can occur are the no-faults found (NFF). We have investigated the influence of intermittent resistive faults (IRF), an NFF resulting from e.g. cracks in ball-grid array-to-board interconnections and on-chip interconnections. Previous circuit simulations from us have indicated the potential effect of IRFs on the behavior of analogue as well as simple digital CMOS circuits. In this paper, a hardware IRF generator is presented to accelerate the IRF fault injection process. As a case study, a digital CMOS UART is selected and experimental results for software and hardware -based fault injections are provided. The experimental results demonstrate that the IRF fault injection can be accelerated by 7 orders of magnitude.**

***Keywords*-Intermittent resistive faults, no faults found, IRF measurements, test generator, dependable digital design, CMOS.**

I. Introduction

A key issue in the development of smaller dimensions and higher complexity of electronic systems is the continuous reduction in their dependability. In electronic systems like system chips and printed-circuit boards, interconnection wiring is by far dominating and therefore faults in these parts are of crucial importance.

A family of interconnection faults which is very difficult to detect are the No-Faults-Found (NFF), although they are known under many different names [1]. A specific category of NFFs is intermittent resistive faults (IRF) [2]. They are characterized by random low-level resistive (burst) occurrences in time, randomly fixed in locations, but often repairable if found. It also includes intermittent (highly resistive) opens and (low resistive) shorts. These faults rank among the highest in terms of occurrence ($> 50\%$) as well as in cost and is expected to increase in future technology nodes [3].

An example of a measured intermittent resistive fault is shown in Figure 1 [4]. It shows that over time (aging) the resistance values increase and the durations of faults get longer; eventually they often result in permanent faults which are easy to detect. The key issue is to detect IRFs at the most *early* stage. A likely root cause of IRFs is marginal

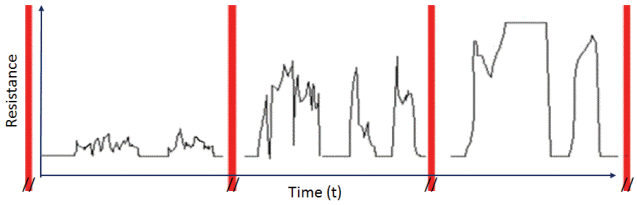

Figure 1. Typical appearance of an IRF and its development in time (aging) [4]

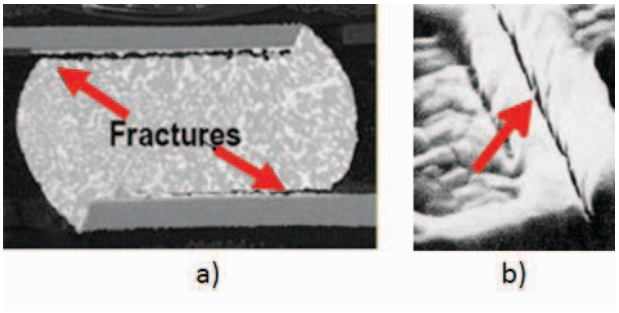

Figure 2. a) Cracks in a ball-grid interconnection and b) crack in a chip interconnection causing both IRFs

or unstable interconnections. In advanced integrated circuits, as well as printed-circuit boards, 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 [3] causing increased instability.

Figure 2 shows two unstable conditions in an interconnection, causing IRFs. Figure 2a shows two cracks between chip package and board which are caused by different (mechanical) stress factors. Figure 2b depicts a crack in chip interconnect as result of a step-coverage problem. Both are very sensitive to temperature and mechanical stress / vibration.

The influence of intermittent faults on digital systems has been studied in many papers [5-9] by simulation-based fault injection at RTL-level. In [10] and [11], we have proposed a simulation model for IRFs and analysed the influence of IRFs on a simple analogue OpAmp and a simple digital circuit (full-adder) at the *transistor* level via simulation. Here, based on our previous IRF model a hardware IRF

978-1-5090-2817-7/16 $31.00 © 2016 IEEE
DOI 10.1109/DSD.2016.58

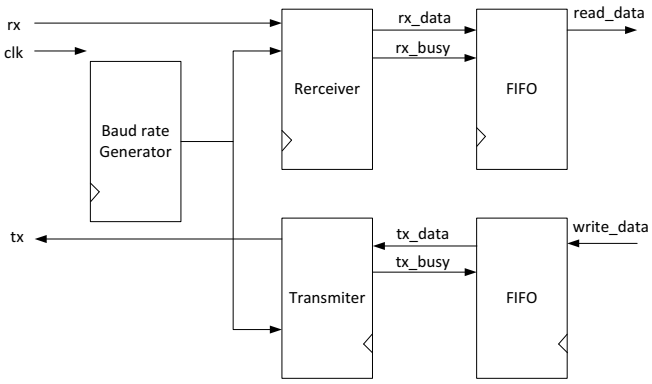

Figure 3. Basic block diagram of a UART

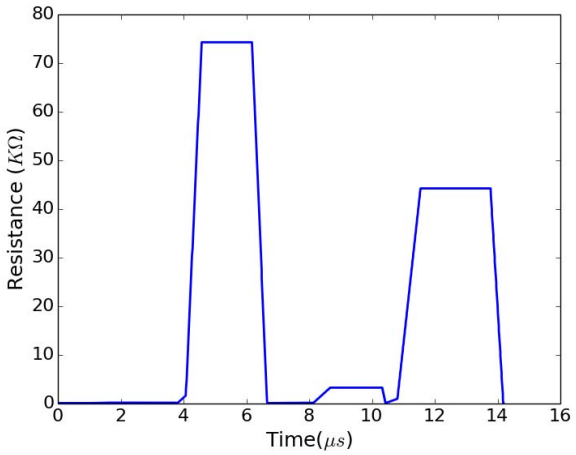

Figure 5. Set of three pulses (IRF) resulting in faulty behaviour

generator has been developed to speed up the fault injection process.

This paper is organized as follows. In section II, our generic simulation model for intermittent resistive faults (IRF) is being used to carry out IRF fault-injection and subsequent evaluation (Cadence Virtuoso) of the behaviour of a 45nm NanGate CMOS UART via simulation. Section III explains the design, implementation and validation of our highly programmable *hardware* IRF generator. In section IV, this generator is used in combination with an FPGA implementation of the previous UART design to actually measure the effects on its logic behaviour. The paper is completed with conclusions and a discussion in section V. The usage of an IRF detector will be suggested to significantly increase the dependability of future safety-critical CMOS systems.

II. Evaluation of IRFs in CMOS systems

In this paper a more complex digital CMOS circuit, a Universal Asynchronous Receiver Transmitter (UART) [12] has been used as an example for IRF evaluation. It has been designed in the 45nm NanGate process, using the Cadence design environment and our IRF model generator [11]. A UART sends data from one system to another system, where parallel data is converted into serial data and vice versa. A basic scheme of a UART is shown in Figure 3. Essential parts are a baud-rate generator, consisting of e.g. a delay-locked loop (DLL) and dividers. Furthermore, it includes a transmitter (with or without a FIFO) and a receiver (with or without a FIFO); usually also several UART registers are included.

A design was made at logic gate level of this UART (excluding FIFOs) which can be downloaded on an FPGA to get the hardware implementation which will be used for actual hardware verification later on in the paper.

A number of different IRF sequences were applied from the programmable IRF simulation model [11] (number of pulses, local pulse durations and different associated resistive values) to the data input (RX) of the UART. A nominal power supply Vdd of 1.1V was used and the clock frequency was around 5.5MHz (baud rate $\sim$ 921 kHz). The UART first receives a start bit and then the less significant bits of 8-bit

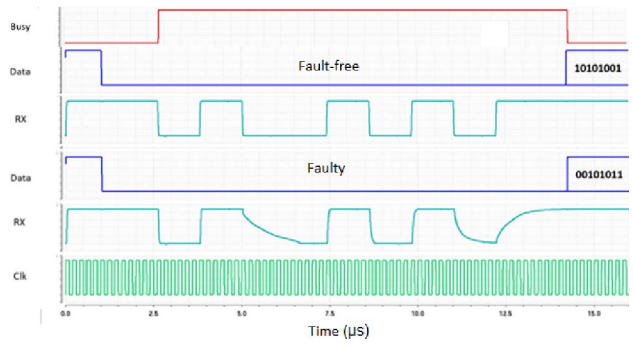

Figure 4. Simulation results of a UART under influence of an IRF at receiver input RX

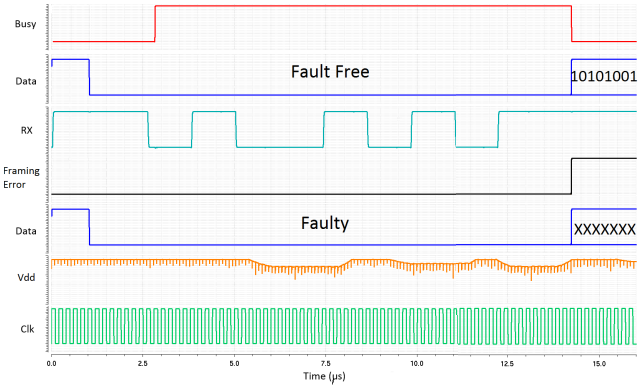

Figure 6. Simulation results of a UART under influence of an IRF at the Vdd terminal resulting in a frame error.

data.

Figure 4 shows actually two situations: on top, where the IRF did not cause any faulty logic behaviour (Fault-free) in the transmitter (TX) data output (Data), and a second (bottom) where a fault did occur (Faulty). The latter was caused by an IRF with pulse duration of 2μs, and a resistive value of 75 $k\Omega$ at the receiver input RX. The injected IRF sequence of three resistance pulses is shown in Figure 5.

In Table 1, all relevant parameters of the IRF generator are shown. For simplicity, uniform distributions were used in all cases, but they can be changed if required. Each burst begins with a random start-time varies between a min and max values. Then, a random activation time (T-active) is chosen, during which a random resistance value R is assigned to this timeframe. After each active time there is a random inactive time during which the faulty connection will have the minimum resistance (here 10 Ω). The burst length indicates the maximum number of resistance pulses in an IRF sequence (e.g. here 5). Figure 4 with the faulty occurrence in the output RX was just chosen arbitrarily to illustrate the procedure. From earlier work [11], it has been shown that IRFs at the power-supply or ground lines are notorious for disturbing correct logic operation.

In a serial data communication unit, like an UART, an IRF can result into a *multiple-bit* error or a *framing* error. In our case, a frame consists of one start bit, 8-bit of data and one or two stop bits respectively. If the receiver gets more or less data than a frame, a framing error will occur.

The following figure shows the simulation result of the UART in the case there is an IRF in the Vdd line. In this case, an IRF has resulted in a framing error, and hence the received data (XXXXXXXX) is not valid. As was expected, the simulation results show that the Vdd line is more sensitive to IRFs than the example in Figure 4. If the IRF has pulses with resistance higher than about 40 $k\Omega$ it leads to a data error.

It is obvious the combination of resistance values as well as pulse duration play a key role to produce a faulty behaviour.

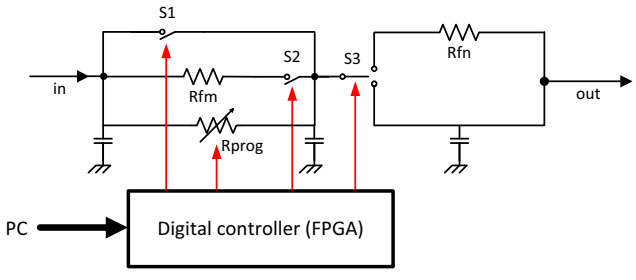

Figure 7. Scheme of the programmable IRF generator in hardware

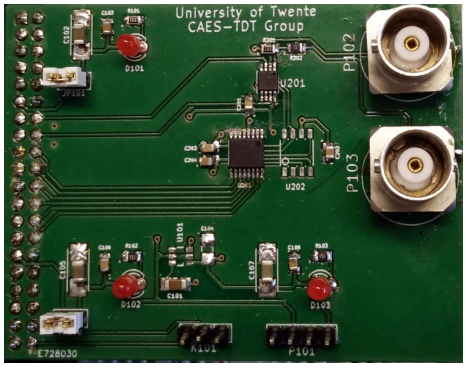

Figure 8. Photo of the implemented hardware IRF generator

Table I
IRF GENERATOR PARAMETERS USED IN MEASUREMENTS FOR UART

Parameter	Minimum	Maximum	Distribution
Start time	0.1 μs	1 μs	Uniform
Resistance	10 Ω	80 $k\Omega$	Uniform
T-Active	0.6 μs	3 μs	Uniform
T-Inactive	0.3 μs	1 μs	Uniform
Burst lenght	1	5	Uniform
Safe time	1 μs	(years)	Uniform

III. Design, Implementation and Test of a Hardware IRF Generator

A simulation-based intermittent resistive fault injection in Verilog-A has been used in the past to evaluate the IRF effect on digital circuits [11]. However, simulation-based fault injection is extremely time-consuming, particularly in the case of IRFs. Emulation-based hardware fault injection is an alternative solution for accelerating fault injection. This technique also allows studying the behaviour of a circuit in real time, and also large (PCB-based) systems can then be validated in real-time [13, 14].

In Figure 7, a rather straight-forward implementation of the hardware IRF generator is shown. It is able to be connected as a programmable resistive wire between any two nodes. There are of course some constraints with regard to the allowed branch voltages and currents and switch positions. The concept is to use a simple network of programmable fixed (Rfm and Rfn) and programmable

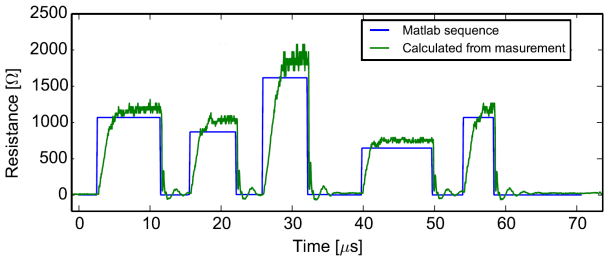

Figure 9. Measured output of the programmable IRF generator

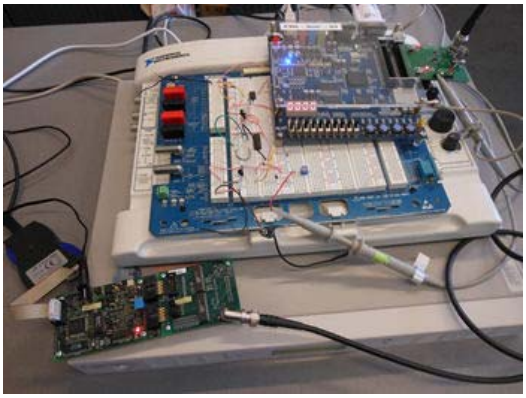

Figure 10. Measurement set-up of the IRF generator and CMOS UART for detecting output failures

variable resistances (Rprog). The total range of resistances, as well as the resolution of the smallest resistance change were the main design parameters. A digital controller (e.g. implemented in an FPGA) generates the desired resistance by providing the required control signals for the resistances and the digital switches (S1, S2, S3) on the board.

The most obvious disadvantage of this architecture is that its speed behaviour is limited by the RC combinations (see Figure 7), of which the intrinsic parasitic capacitances are the most difficult to circumvent. However, in terms of speed this was found to be an acceptable approach at this moment. A photograph of the hardware design implementation is shown in Figure 8.

A personal computer executes a Matlab script which generates a burst of random resistance values based on our model. The generated burst sequence can be transferred to an FPGA board (e.g. Xilinx Zynq-700) by a serial communication link.

The FPGA is responsible to synchronize and generate the actual control signals for the on-board programmable switches and potentiometers on the IRF generator.

An example of a generated IRF signal by our generator is shown in Figure 9. There are two waveforms in the figure; one is a resistance sequence measured from the IRF board (green) and the other (blue) is the expected sequence generated by the Matlab script. As can be seen, the measured resistance sequence approximately follows the expected sequence with an error because of the parasitic capacitances of the existing switches and potentiometers. A calibration step can be easily applied in this case, showing to close the gap between the two signals in the figure, and further proper grounding measures removes the noise.

IV. Measurements of IRFs in logic CMOS circuits

The previously discussed hardware IRF generator has been used for the evaluation of IRF faults in a CMOS UART implemented on an FPGA. The implemented UART design is the same one as used in the previous simulations. An impression of the overall test set-up is shown in Figure 10.

The baud rate in the case of measurements is similar as in the previous simulations. The parameters of the IRF are as depicted in Table 2. The major difference is in the range of resistance. In our case they are below 3 $k\Omega$, which is rather low and hence less likely to cause logic errors.

The UART first receives the start bit and then the less significant bit of an 8-bit data stream. This data is shown in Figure 11, being 01101010 (signal 1). In the case of our IRF, a distorted signal is shown (signal 2). The next signal

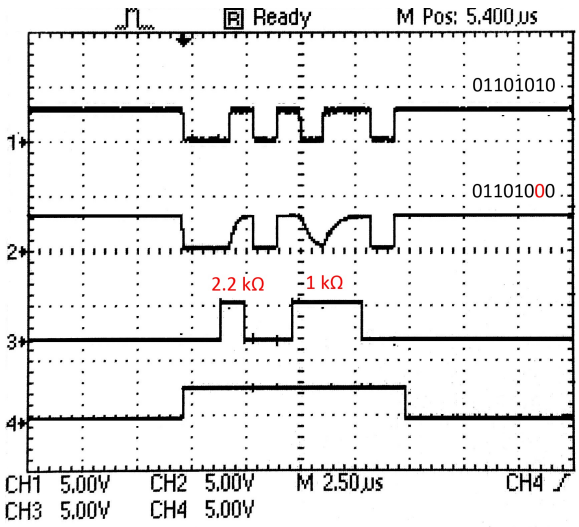

Figure 11. Measurement results on actual CMOS UART and our hardware IRF generator (single bit fault)

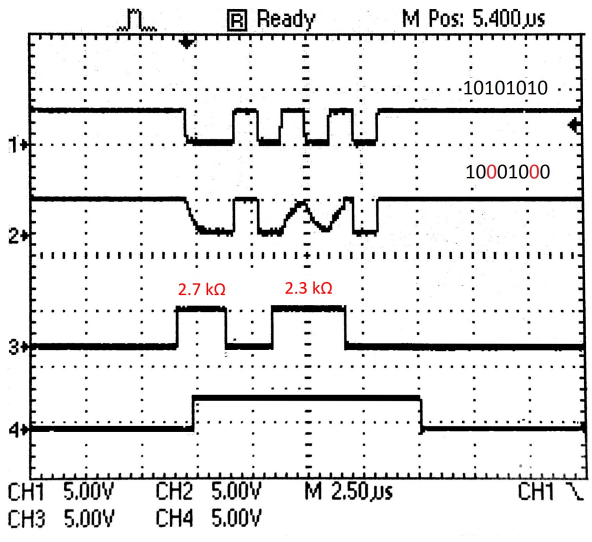

Figure 12. Measurement results of UART using a different IRF resulting in multi-bit faults

Table II
IRF GENERATOR PARAMETERS IN THE CASE OF REAL MEASUREMENT

Parameter	Minimum	Maximum	Distribution
Start time	0.1 μs	1 μs	Uniform
Resistance	2.5 Ω	3 $k\Omega$	Uniform
T-Active	0.6 μs	3 μs	Uniform
T-Inactive	0.3 μs	1 μs	Uniform
Burst lenght	1	2	Uniform
Safe time	1 μs	(years)	Uniform

(3) shows the occurrence of an IRF fault on RX line. The last line is the busy signal (4), when the receiver captures data. As can be seen the 8-bit signal is now 01101000, and shows a difference of 1 bit. Figure 12 shows the situation where the values of the resistive faults are different than the previous example. In this case, two bits (signal 2) are faulty.

Further experimental results show that an IRF can cause no bit error, one bit, several bits or framing error based on the range of resistance changes, activation time and the length of the IRF.

The comparison of the simulation and hardware -based fault injection shows a significant speed enhancement in IR fault injection. In Section II, the fault simulation were done using a 64-bit Linux machine with 8 GB RAM and running at 3.4 GHz. The simulation time was about 5 minutes. Whereas, the hardware-based fault injection in Section IV can be done in about 30 us. This means that the intermittent ressitive fault injection has been accelerated up to 7 orders of magnitude.

As it demonstrated in these experiments, IRFs could cause real-life failures in more complex CMOS logic systems. There is now also a framework to further investigate in other systems how sensitive they are with respect to IRFs. This can help during the design to reduce their effect. In another paper we will present also dedicated embedded instruments using IEEE 1687 which are capable of detecting IRFs at a very early stage, thus dramatically reducing the costs and increasing the dependability of future CMOS systems. As IRFs are located at fixed positions, early detection and time stamping (e.g. temperature conditions) will make debugging and repair much easier and hence less costly.

V. CONCLUSION

The research in this paper has shown in simulation as well as in actual tests that a category of No-Faults Found (NFF), the so-called intermittent resistive faults (IRF), can result in faults in the logic behaviour of integrated CMOS systems. A programmable hardware IRF signal generator has been designed and tested, and subsequently applied to an actual digital CMOS system (UART) to validate the effect of these faults. The experimental result showed the proposed IRF generator can speed up the injection of intermittent resistive fault by several orders of magnitude. In combination with an IRF monitor, discussed in another paper, countermeasures can be taken to reduce this category of faults and therefore significantly enhance the dependability of future CMOS systems in safety-critical applications.

ACKNOWLEDGMENT

This research was carried out within the FP7 BASTION project (#619871), financed by the European Committee (EC) and the Netherlands Enterprise Agency (RVO).

REFERENCES

[1] Accenture Report, "Big Trouble with No Trouble Found Returns", http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture_Returns_Repairs.pdf., 2008.

[2] BASTION Survey, Public Report on the NFF and Aging Fault Study, (D1.1), June 2015.

[3] K. Anderson, "Intermittent Fault Detection & Isolation System (IFDIS)", white paper Synaptics, 13th CTMA Symposium, March 2012.

[4] Ridgetop Group Inc., "SJ BIST", White paper presentation 2013.

[5] 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", in IEEE Trans. on Reliability, vol. 63, 2014, pp. 144-153.

[6] 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, 2012, pp. 777-790.

[7] D. Gil-Toms, J. Gracia-Morn, J. Baraza-Calvo, L.-J. Saiz-Adalid, and P.-J. Gil-Vicente, "Studying the effects of intermittent faults on a microcontroller," Microelectronics Reliability, vol. 52, 2012, pp. 2837-2846.

[8] J. Gracia-Moran, D. Gil-Tomas, L. J. Saiz-Adalid, J. C. Baraza, and P. J. Gil-Vicente, "Experimental validation of a fault tolerant microcomputer system against intermittent faults," in IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN), 2010, pp. 413-418.

[9] 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), 2010, pp. 11-18.

[10] 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), Porto Alegre, Brazil, 2014, pp. 1-6.

[11] H.G. Kerkhoff and H. Ebrahimi, "Intermittent resistive faults in digital CMOS circuits". In: IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, DDECS 2015, ISBN 978-1-4799-6779-722-24, Belgrade Serbia, April 2015, pp. 211-216.

[12] Patongy, "Intro to UARTS", EXAR Powerpoint Shareslides presentation, April 2010.

[13] H.G. Kerkhoff and H. Ebrahimi, "Detection of Intermittent Faults in Electronic Systems based on the Mixed-Signal Boundary-Scan Standard", in Proc. of the Asian Quality Electronic Design Conference (ASQED), DOI 10.1109/ACQED.2015.7274011, Kuala Lumpur, Malaysia, August 2015, pp. 77-82.

[14] H.G. Kerkhoff and H. Ebrahimi, "Investigation of Intermittent Resistive Faults in Digital CMOS Circuits", Journal of Circuits, Systems and Computers, World Scientific Publishing, Vol. 25, October 2015, 17 pages.