FP7-ICT / FET-OPEN – 309129 / i-RISC

D2.2

Higher abstraction fault models and their simulation methodology

Editor:	Alexandru Amaricai (UPT)
Deliverable nature:	Public
Due date:	October 31, 2014
Delivery date:	November 14, 2014
Version	2.0
Date of current version:	April 15, 2016
Total number of pages:	72
Reviewed by:	i-RISC partners
Keywords:	Delay & Energy Models, Probability Density Function, Reliability Assessment, Fault Injection & Emulation, Probabilistic Fault Models for Interconnects.

Abstract

This deliverable presents the overview of the main results and activities carried out within the Work Package 2 (WP2) framework during the Month 13 to Month 21 (M13-M21). The main contributions included in this deliverable are related to: (i) comprehensive delay estimation models for timing analysis in CMOS circuits (Task 2.1), (ii) correlated errors modeling and degradation quantification for Probability Density Function (PDF)-based circuit reliability assessment (Task 2.3), (iii) probabilistic gate level/Register Transfer Level (RTL) fault models for interconnects (Tasks 2.1, 2.4), (iv) data dependent Simulated Fault Injection (SFI) for RTL circuit descriptions (Task 2.4), (v) cost effective FPGA fault emulation for probabilistic circuits (Task 2.4), and (vi) the envisaged sub-powered CMOS circuits energy modeling methodology and some preliminary energy evaluation results.

List of Authors

Participant	Author
TU-Delft	Joyan Chen
	Nicoleta Cucu-Laurenciu
	Sorin Cotofana
	Thomas Marconi
UPT	Alexandru Amaricai
	Sergiu Nimara
	Flavius Pater
	Mircea Popa
	Oana Boncalo
UCC	Emanuel Popovici
	Satish Kumar Grandhi
	Christian Spagnol

Table of Contents

List of Figures

List of Tables

Abbreviations

AES	Advanced Encryption Standard
AIG	And-Inverter Graph
BER	Bit Error Rate
BN	Bayesian Network
BRAM	Block Random Access Memory
CMOS	Complementary Metal Oxide Semiconductor
CPE	Codeword Prediction Encoding
CUT	Circuit Under Test
DAG	Directed Acyclic Graph
DFF	D Flip-Flop
DEMUX	De-multiplexer
DUT	Design Under Test
ECC	Error Correcting Code
EFI	Emulated Fault Injection
FER	Frame Error Rate
FO	Fan-Out
FOC	Fan-Out of Current gate
FOP	Fan-Out of Previous gate
FPGA	Field Programmable Gate Array
GD	Gaussian Distribution
GF	Galois Field
GL	Gate Level
HDL	Hardware Description Language
IGD	Inverse Gaussian Distribution
ICON	Integrated Controller
ILA	Integrated Logic Analyzer
ISCAS	International Symposium on Circuits and Systems
JTAG	Joint Test Action Group
LDPC	Low Density Parity Code
LFSR	Left Feedback Shift Register
LUT	Look-Up Table
MAJ	Majority Voter
MCS	Monte Carlo Simulation
MUX	Multiplexer

OCV	On-Chip Variability
PDF	Probability Density Function
PRNG	Pseudo Random Number Generator
PVT	Process Voltage Temperature
QoS	Quality of Service
RAM	Random Access Memory
RLC	Resistance, Inductance and Capacitance
RNG	Random Number Generator
RTL	Register Transfer Level
SEU	Single Event Upset
SFI	Simulated Fault Injection
SNR	Signal to Noise Ratio
TK	Taylor Kuznetsov
TLM	Transaction Level Modeling
TMR	Triple Modular Redundancy
TRNG	True Random Number Generator
VIO	Virtual Input / Output
WP	Work Package

1. Executive Summary

In the period Month 13 to Month 21 (M13-M21), Work Package 2 (WP2) activities addressed the achievement of Objective 2.2 - Development of higher abstraction levels (gate level, RTL, functional) fault and error models and the corresponding simulated fault injection methodology. In this line of reasoning, WP2 has been tasked with two main contributions towards the i-RISC goals.

The first objective was to provide correlated fault models, which accurately capture the physical phenomena occurring at device level, and to bring their implication at the gate and circuit levels (Task 2.3). For this purpose a thorough circuit level analysis (Task 2.1) has been performed in order to determine the correlation between the variation in physical parameters, e.g., temperature, supply voltage, threshold voltage, oxide thickness, and the reliability of sub-powered digital circuits captured by means of Probability Density Functions (PDFs). We further utilized such PDFs as degradation quantifiers to develop the reliability assessment method introduced in Deliverable 5.1 **[D5.1]** into a PDF-based circuit reliability assessment framework (Task 2.3). Furthermore, work has been initiated towards the development of gate level energy models for probabilistic CMOS circuits (Task 2.5).

The second objective was to define fault models for higher than transistor abstraction levels, i.e., Gate Level (GL) and Register Transfer Level (RTL) and to propose and develop methodologies for their Simulated Fault Injection (SFI) according to the Task 2.4 description. We note that GL/RTL simulation based techniques are essential for the project as they allow for reliability assessment and validation of the proof-of-concept circuits and architectures under development with the WP6 framework. In order to achieve this goal, we have pursued a hierarchical approach. This type of approach provides means for accuracy-scalability trade-offs characteristic to simulation based analysis as: (i) low level analysis presents good accuracy, but does not scale for complex circuits, while (ii) assessment at high abstraction layers has good scalability, but also low accuracy. The proposed methodology tries to combine the GL accuracy with the RTL scalability: GL simulations are used to derive the probabilities for RTL blocks, while the entire system's reliability is assessed using RTL simulations. Furthermore, we have addressed Field Programmable Gate Arrays (FPGA) fault emulation techniques for probabilistic error analysis as means to accelerate both GL and RTL simulations.

Figure 1-1 presents the WP2 Gantt diagram, which indicates that the tasks addressed and initiated during the period M13-M21 are:

- Task 2.1 - SPICE analysis of sub-powered CMOS circuits in deep sub-micron technologies;
- Task 2.3 - Accurate fault models for correlated errors;
- Task 2.4 - Development of simulated fault injection methodology at higher abstraction levels;
- Task 2.5 – Energy models for sub-powered circuits.

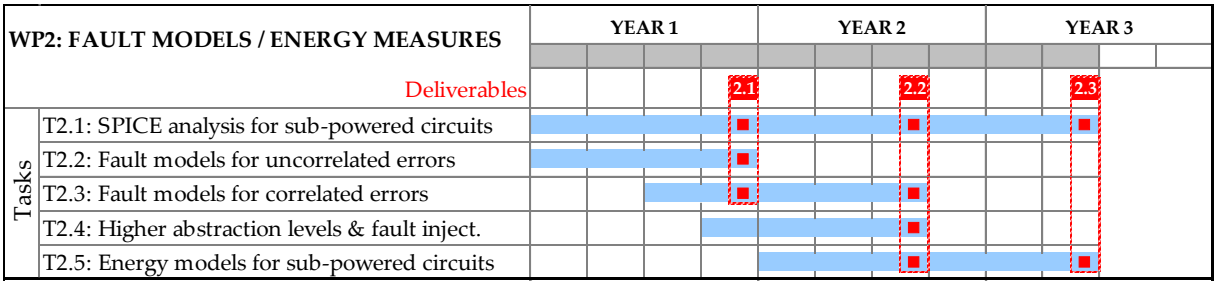


Figure 1-1 - WP2 Gantt Diagram

Related to these tasks the main technical contributions presented in this deliverable can be summarized as:

- **Inverse Gaussian Distribution (IGD) based timing analysis of sub-powered CMOS circuits (Task 2.1) –** We enhanced the IGD model capabilities such that it can be applied to sequential elements and can to cover some important circuit behavior related factors, such as fan-out and transition time. Overall, a comprehensive delay model for sub-powered circuits with highly accurate estimation capability is provided, which outperforms the traditional Gaussian Distribution (GD) based model. Moreover, to demonstrate the accuracy of the IGD based delay estimation, a circuit comprised of D Flip-Flops (DFFs) with 8-bit De-multiplexer (DEMUX) and Multiplexer (MUX) was implemented while considering the newly added features. The experiments indicate that our method provides a high accuracy, an average error less than 1.2% when compared with the results of a Monte Carlo simulation in HSPICE, while diminishing the simulation time with orders of magnitude.

- **Correlated error modeling and degradation quantification for PDF-based circuits reliability assessment** (Task 2.3) – We investigated aspects afferent to the practical utilization of the framework introduced in Deliverable 5.1 **[D5.1]**, which asses a circuit reliability based on given inputs and prior Probability Density Functions (PDFs) of its comprising gates. To this end, we addressed the error modelling and degradation quantification from both theoretical and simulation points of view. The PDF-based gate reliability design-time pre-characterization was discussed and exemplified for an inverter and a NAND2 gate, as discussion vehicles. We proposed to employ a high-level degradation quantifier, i.e., an output voltage based PDF, in order to capture a circuit multiple correlated degradation effects, when being exposed to different aggression profiles. Furthermore, we note that when propagating the PDF figure throughout the circuit according to the reliability assessment we introduced in Deliverable 5.1, the correlation between the different comprised circuit gate behaviors is inherently captured, and thus the correlation of different errors encountered in the circuit is being accounted for. We also introduced a practical PDF based simulation framework and evaluated the reliability of a set of ISCAS'85 circuits, based on the prior PDFs that individual gates have accrued from the design-time pre-characterization step.

- **Multi-level simulation based reliability assessment of RTL designs (Task 2.4) –** We propose a simulation based methodology which allows the evaluation of data dependent probabilistic fault on RTL descriptions. It combines the high accuracy of data dependent gate level simulation with the low simulation times associated for RTL abstraction. The proposed methodology consists of five phases: (i) hierarchical block decomposition, which has the goal of providing simple blocks that can be simulated at GL, (ii) RTL correct simulation, which aims at extracting the inputs for the considered blocks, (iii) logic synthesis, which outputs the GL netlists for the considered components, (iv) data dependent GL SFI, which provides the probabilities of correctness for each considered sub-circuit, and (v) RTL SFI, after which the reliability estimates of the entire system are derived. We have applied the proposed

methodology on a crypto-core, for which the GL analysis could not be performed (due to the large memory requirements of the GL simulation).

- **Cost effective FPGA emulated fault injection for probabilistic errors (Task 2.4)** – We have proposed a novel FPGA emulation scheme for probabilistic errors. It presents accurate probabilistic fault modeling capability. Our emulation methodology is based on a True Random Number Generator (TRNG) for probabilistic fault generation and a shift register for fault insertion to their corresponding fault locations. The usage of TRNG leads to a high modeling accuracy due to avoidance of correlations between errors. The drawback of the proposed methodology is represented by the high emulation time, due to shift register loading. In order to reduce the emulation time, several TRNG – shift registers schemes are used.

- **Probabilistic fault models and simulation fault injection for interconnects (Task 2.1, Task 2.4)** – We have addressed the data dependent probabilistic fault modeling for GL and RTL analysis of interconnects. We have proposed the following data dependent fault models for interconnects: the simple standard signal probabilistic fault model, the probabilistic switch-aware fault model, the full data dependent switching probabilistic fault modes, and the partial data dependent switching probabilistic fault model. The full data dependent fault model takes into consideration the effects of all the wires within a bus on the correct switching probability of a specific wire; it captures accurately the influence of both capacitive and inductive crosstalk. The partial data dependent fault model takes into consideration the effect of neighboring wires on the correct switching probability of a wire; it is based on the fact that the effects of inductive crosstalk (which spans over multiple wires) are negligible with respect to the effects of capacitive crosstalk.

- **Progress Towards Energy Modeling of Sub-Powered CMOS Logic Circuits (Task 2.5)** – We present preliminary results regarding IGD based energy modeling for logic gates operating at near and sub-threshold voltages. Furthermore, we present the progress towards energy modeling and assessment for faulty LDPC decoders.

The deliverable is organized as follows: Chapter 2 is dedicated to the timing analysis of sub-powered CMOS circuits by means of the Inverse Gaussian Distribution approach. Chapter 3 introduces correlated error modeling and PDF based degradation quantification targeting the reliability estimation of complex CMOS circuits. The Multi-level Simulated Fault Injection (SFI) approach for RTL descriptions is detailed in Chapter 4, while the proposed cost effective FPGA emulation methodology is presented in Chapter 5. Chapter 6 describes data dependent probabilistic fault modeling for interconnects. The progress towards the development of energy models is reported in Chapter 7. The last chapter is dedicated to concluding remarks and future work.

2. Inverse Gaussian Distribution Based Timing Analysis of Sub-powered CMOS Circuits

Abstract: In the era of deep submicron CMOS technology, spatial unreliability or process variability, and temporal unreliability, cause less predictable device behavior. This reflects on difficulty in timing analysis/estimation and has significant impacts on the reliability of the entire circuit. This issue is further deteriorated in near/sub threshold region, which is of interest when low power consumption is envisaged. Given that traditional delay models or delay estimation methodologies struggle to accurately capture the circuit behavior, an Inverse Gaussian Distribution (IGD) based delay for combinational circuits has been introduced in Deliverable 2.1 **[D2.1]** and has been published in **[Chen14]**. In this work, we extend the proposed IGD model for sequential elements and to cover some important circuit behavior related factors such as fan-out, transition time. Overall, we provide a complete delay model for sub-powered circuits with highly accurate estimation capability, which outperforms the traditional Gaussian Distribution (GD) based model **[Zaynoun12]**. In order to demonstrate the accuracy of IGD based delay estimation, a circuit comprised of D Flip-Flops (DFFs) with 8-bit De-multiplexer (DEMUX) and Multiplexer (MUX) is implemented where the newly added features are covered. When compared with Monte Carlo simulation in HSPICE, our approach produces high matching accuracy, with an average error less than 1.2%.

Publications: To be submitted to publication

2.1. Previous Work and Motivation

In previous deliverables, we proposed a delay estimation model, namely IGD model. The close match with Monte Carlo Simulation (MCS) has been demonstrated. Moreover, linear compositionality of IGD model has also been investigated. To complete and take full advantage of the IGD model, two additional aspects have to be taken into account: 1) sequential elements and 2) fan-out and transition time effect. The evolved IGD model thereby can be used to estimate the delay path which involves more complicated conditions.

2.2. IGD Based Delay Model for Sequential Circuits

In synchronous CMOS circuits, data are synchronized through sequential elements such as D-latch (DL) and D flip-flop, and then fed into combinational circuits to carry out the logic evaluation for the next stages, which is regarded as a timing segment in static timing analysis. Each delay segment is then accumulated together to finalize the timing analysis, known as Register→Register (R2R) delay. The equation of the typical propagation delay is given by:

$$D_{R2R} = D_{C2Q} + D_{LOGIC} + D_{SETUP} \quad (2\text{-}1)$$

where D_{R2R} is the total delay of a timing path, T_{C2Q} is the delay of D flip-flop from the clock rising/falling edge to the output, D_{LOGIC} is the propagation delay caused by combinational logic circuits, finally D_{SETUP} is the setup time for output registers. Unlike the conventional methodology which sums up of exact delay value of each component, our approach propose to estimate the aforementioned IGD key parameters μ and λ for each of these terms. For the time being we just focus on the D_{C2Q} *and* D_{LOGIC} delay models as D_{SETUP} is significantly smaller than the others. The

investigation of D_{SETUP} will be carried out in the future work. In Deliverable 2.1 **[D2.1]**, we have introduced a method to compute μ and λ of the delay at the output of a combinatorial logic using a linear combination of the corresponding parameters of the gates in the longest path. In view of this Equation can be translated into:

$$\mu_{R2R} = \mu_{C2Q} + \mu_{LOGIC}$$

$$\lambda_{R2R} = \lambda_{C2Q} + \lambda_{LOGIC} \quad (2\text{-}2)$$

In the following subsections, the key parameters for DFF are obtained, which will be utilized later in validation.

As an important part in circuit designs, data are synchronized through sequential elements such as DL and DFF which are significant in terms of timing analysis. Unlike combinational circuits, usually cross-coupled circuits are involved for data retention. Therefore, it is important to verify that the IGD model fits well also for these for such sequential elements. A DFF is composed of two adjacent DLs known as Master and Slave controlled by complementary clock signals. The DFF built with eight NAND gates and two INVs is depicted in Figure 2-1. The IGD and GD fitted PDFs of DFF along with MC simulation data with FO=1 are presented in Figure 2-2 where both 1→0 and 0→1 events at the input end (D) are considered. The key parameters, μ and λ under different conditions are summarized in Table 2-1 along with the key parameters for INV and NAND gate reported in **[Chen14]** which are going to be utilized in implementing DEMUX and MUX. It can be seen that the values for 1→0 transition is greater than that for 0→1 transition which means that discharging event takes more time than that of charging event. Again, the IGD fits the MC simulation data better than the GD. So far, the practicability of fitting sequential gates using the IGD model is presented. The shapes of the data and IGD fitting curves in Figure 2-2 are not symmetric, which gives evidence of inappropriateness of the GD model once more **[Chen14]**. A solid platform for delay estimation in a typical timing path based on IGD model is presented. In next section, the effect of fan-out and input transition time in our approach is discussed.

2.3. Expansion of IGD Model with Fan-out Effect

In last section, a basic equation to calculate the IGD key parameters in a timing path was introduced along with several IGD fittings and the corresponding key parameters for sub-powered gates. Moreover, the linear compositionality of the IGD model for combinational circuits has been demonstrated in Deliverable 2.1 **[D2.1]** and **[Chen14]**. To complete our delay model, additional factors have to be taken into account to meet more realistic conditions. Among them, fan-out is a crucial component. Fan-out, also regarded as capacitive load, at the output of a gate can significantly affect the transition time of output signals and the propagation delay. In fact, there are two types of fan-out affect the delay 1)Fan-Out of Current gate, FOC 2)Fan-Out of Previous gate, FOP. FOC has direct impact on delay. On the other hand, high FOP results in long transition time at output signals (serving as inputs to coming gates), which subsequently increase the propagation delay. However, the driving ability of sub-powered circuits is relatively weak which limits the load capacity at the outputs. In other word, high fan-out number is not suitable or requires careful designs in near/sub-threshold circuits. In this work, the maximum fan-out number considered is four.

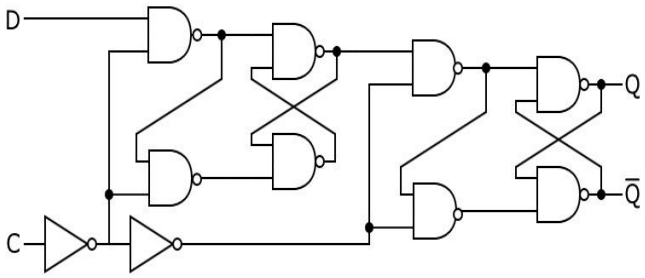


Figure 2-1 - Schematic of a D Flip-Flop.

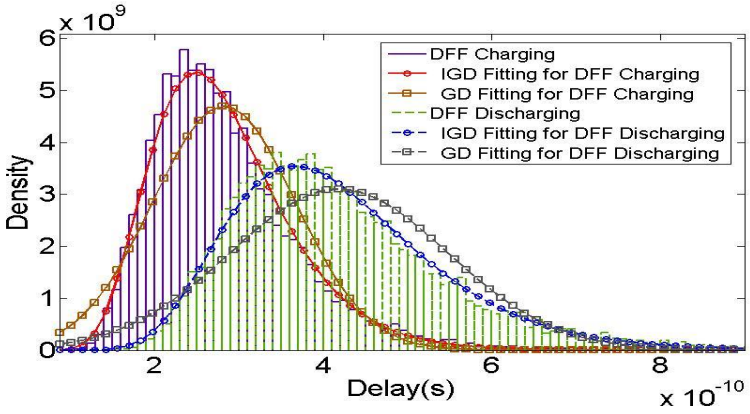


Figure 2-2 - IGD and GD Fittings for Charging and Discharging Events of a DFF.

Table 2-1 - μ and λ for INV, NAND, DFF

GATE	Charging		Discharging	
	μ (e^{-11})	λ (e^{-10})	μ (e^{-10})	λ (e^{-10})
INV	4.8	9.3	5.8	9.6
NAND	6.2	11.3	7.9	7.7
DFF	28.2	33.4	41.8	47.2

The corresponding method to calculate the key IGD parameters of combinational elements, i.e., INV and NAND, with various FOs and transition time are explored in this section. When high fan-out is needed for sequential elements, i.e. DFFs, INVs (Buffers) are inserted to distribute the signals instead of directly connecting several logic gates to the DFFs output. This practice justify narrowing our study on the impact of FO only to combinatorial elements.

This section is touch upon three problems A) how to enhance the approach to deal with FOC and B) how to address the change of FOP for the IGD delay model C) how to calculate the key IGD parameters under different combinations of FOC and FOP.

2.3.1. Estimation methodology for fan-out effect

It is now of interest to develop a way to link FOC and propagation delay, the logical effort method **[Sutherland99]** is considered here. The method of logical effort is a straightforward technique used to estimate delay in a CMOS circuit. The normalized delay, D, in a logic gate can be expressed as

a summation of two primary factors: parasitic delay, P, which can be found by considering the gate driving no load, and stage effort, F, which depends on the load of the gate.

$$D = NF + P \tag{2-3}$$

where N is the path branching effort which indicates fan-out number, which can be represented as FOP here.

We propose to compute the key parameters of the final output IGD by applying the logical effort methods to them using the following equation:

$$\mu_{FOC} = FOC*F_{\mu} + P_{\mu}$$

$$\lambda_{FOC} = FOC*F_{\lambda} + P_{\lambda} \tag{2-4}$$

To derive F_{μ}, P_{μ} and F_{λ}, P_{λ} for INV and NAND, two sets of data (μ and λ), i.e. FOC=1,2 for each gate, are collected and thereafter calculated using the equation above. Once all the values (F_{μ}, P_{μ}, F_{λ}, P_{λ}) are calculated, μ and λ with various FOC can be evaluated. Those key coefficients are summarized in Table 2-2. It should be noted that the input transition time is set at 100ps for capturing these values.

Table 2-2 - Key Parameters of FOC for INV and NAND

GATE	Charging				Discharging			
	(e^{-11})		(e^{-10})		(e^{-11})		(e^{-10})	
	Pμ	Fμ	Pλ	Fλ	Pμ	Fμ	Pλ	Fλ
INV	3.8	0.9	7.2	1.26	4.6	1.2	8.3	0.4
NAND	5.0	1.2	9.9	0.6	6.0	1.9	6.5	0.8

2.3.2. Estimation methodology for transition time effect

In last subsection, the FOC effect on key parameters of the IGD model has been explored and the methodology to calculate theses values has been introduced. Here, the effect of FOP on propagation will be discussed in the form of namely transition time. It is understandable that high fan-out causes long transition time at the output which serves as input for coming stages, and consequently the higher propagation delay of themselves. A look-up table is generated to exhibit a direct link between FOP and corresponding output transition time. INV gates with FOP=1,2,3,4 are simulated with the same variation set-up used in Deliverable 2.1. The corresponding output transition time are listed in Table 2-3, it can be observed that with 100ps as input transition time, the output transition time of INV increases notably and the increment is quite steady following the increase of FOP for both charging and discharging events. When it comes to FOP=4, the rise time and fall time at outputs (inputs for the following gates) exceed 200ps which can greatly increase the propagation delay of the driven gates.

Based on the data in Table 2-3, it is of interest to investigate the corresponding change in μ and λ for different FOP when FOC=1 for both INV and NAND gate. The corresponding data is listed in Table 2-4, which determines the key parameter increment, namely T_{μ} and T_{λ}.

Table 2-3 - FOP Effect on Output Transition Time

Input Transition 100ps	Output Transition Time – Rise (ps)	Increment (ps)	Output Transition Time – Fall (ps)	Increment (ps)
FOP=1	85	/	99	/
FOP=2	124	39	153	54
FOP=3	165	41	204	51
FOP=4	206	41	256	52

Table 2-4 - FOP Effect on Key Parameters

INV	Charging				Discharging			
	(e^{-11})		(e^{-10})		(e^{-11})		(e^{-10})	
	μ	Tμ	λ	Tλ	μ	Tμ	λ	Tλ
FOP=1	3.8	/	7.2	/	4.6	/	8.3	/
FOP=2	4.6	0.8	8.1	0.9	5.5	0.9	12.5	4.2
FOP=3	5.4	0.8	9.2	1.1	7.2	0.9	16.9	4.4
FOP=4	6.2	0.8	10.2	1.0	8.0	0.8	21.2	4.3

NAND	Charging				Discharging			
	(e^{-11})		(e^{-10})		(e^{-11})		(e^{-10})	
	μ	Tμ	λ	Tλ	μ	Tμ	λ	Tλ
FOP=1	5.0	/	9.9	/	6.0	/	6.5	/
FOP=2	5.8	0.8	12.3	2.4	6.9	0.9	7.9	1.4
FOP=3	6.6	0.8	14.8	2.5	7.8	0.9	9.3	1.4
FOP=4	7.4	0.8	17.4	2.6	8.6	0.8	10.5	1.2

Based on the observation of Table IV, the steady increment in μ and λ can be found represented by constant T_μ and T_λ. Therefore, the effect of FOP on our IGD model can be simply calculated as follows:

$$\mu_{FOP} = (FOP-1)T_\mu$$

$$\lambda_{FOP} = (FOP-1)T_\lambda \qquad (2\text{-}5)$$

2.3.3. The combination of FOC and FOP effects

As two types of FO and their effects on the propagation delay as well as the key parameters of the IGD model have been presented respectively in this section. A straightforward combination of key parameters for FOC and FOP is given in the following equation.

$$\mu_{LOGIC} = FOC*F_{\mu} + P_{\mu} + (FOP-1)T_{\mu}$$
$$\lambda_{LOGIC} = FOC*F_{\lambda} + P_{\lambda} + (FOP-1)T_{\lambda} \quad (2\text{-}6)$$

An example based on INVs where INV2 has FOC=3, FOP=2 is illustrated in Figure 2-3 and the calculation of μ and λ of charging event of *INV2* is carried out based on the values from Table 2-2 and Table 2-4.

$$\mu_{INV2} = 3*0.9e^{-11} + 3.8\,e^{-11} + (2-1)*0.8\,e^{-11} = 7.3\,e^{-11}$$

$$\lambda_{INV2} = 3*1.26\,e^{-10} + 7.2\,e^{-10} + (2-1)*0.9\,e^{-10} = 11.88\,e^{-10}$$

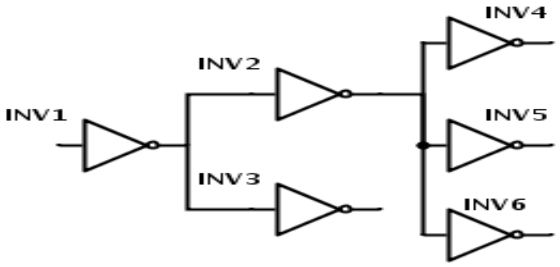


Figure 2-3 - A Sample Circuit with FOC=3 and FOP=2.

2.4. Model Validation for Synchronous Timing Path

To prove that our proposed IGD model together with the proposed method to propagate the key parameters, are valid and applicable to generic circuits we compare the results of the proposed method against results from MCS for a circuit consisting of DFFs + 8-bit DEMUX and MUX, where different fan-out numbers are contained and thereby the method discussed in last section is utilized. The Cumulative Distribution Functions (CDF), which is the PDF integral, is utilized to more clearly quantify the differences between the proposed model and MCS data.

2.4.1. DFFs + 8-bit DEMUX and MUX

According to the methodology presented in Deliverable 2.1, Section 0 and Section 2.3 , the corresponding key parameters of the IGD of the output of an 8-bit DEMUX and MUX with DFFs can be evaluated by using the data and methodology presented in the previous sections, i.e., fan-out number, and the entailing transition time. The schematic of the 8-bit DEMUX and MUX is displayed in Figure 2-4 with only INVs and NANDs being used. In Figure 2-5, the CDFs acquired by MCS, the one obtained with the IGD based estimation, and the one based on GD fitting, are pictured for a delay range between 0ns to 6ns. The curve based on IGD estimation closely approximate the MC simulation with a slight deviation around 2ns. Overall, it is better than the GD fitting. Table 2-5 lists the CDF deviations between MC simulation and the IGD estimation as well as the GD fitting ranging

from 1ns to 6ns with the highest mismatch recorded being 3.4% at 2ns and average overall error being 1.2% for the IGD estimation while the average error for the GD fitting is 7.3%. It is worth

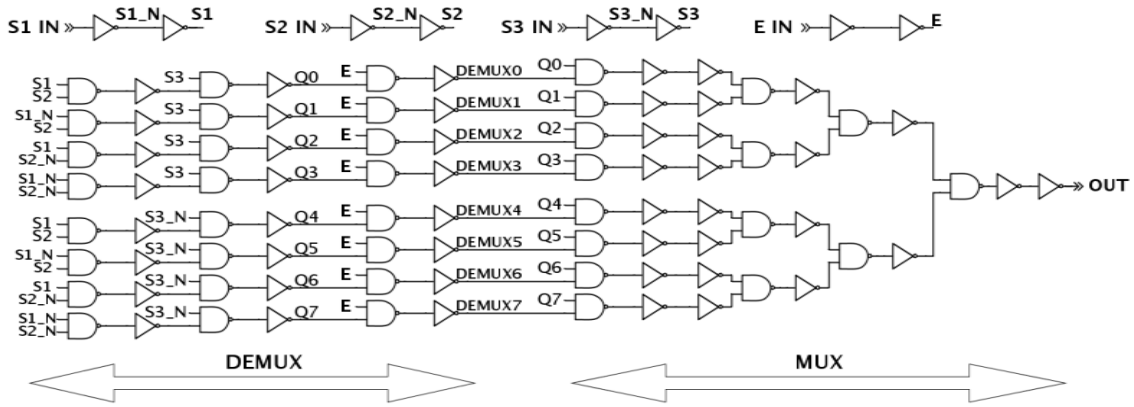


Figure 2-4 - Schematic of 8-bit DEMUX and MUX.

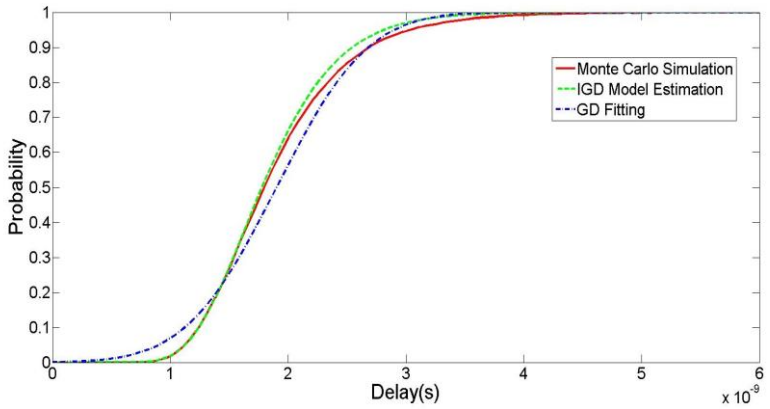


Figure 2-5 - DFFs + 8-bit DEMUX and MUX CDFs.

Table 2-5 - DFFs + 8-bit DEMUX and MUX CDF Deviations

Deviation	1ns	2ns	3ns	4ns	5ns	6ns	Average (1-6ns)
IGD Estimation	2.6%	3.4%	2.4%	0.6%	0.1%	<0.1%	1.2%
GD Fitting	208.8%	11.8%	2.1%	0.7%	0.1%	<0.1%	7.3%

mentioning that, due to the non-zero-crossing of the CDF of the GD, the deviation for the GD fitting will be too large if we choose the timing range from 0ns.

2.5. Conclusion

In this work, an accurate delay model based on IGD along with its corresponding approach taking fan-out and transition time into account was proposed and compared with the state of the art. The IGD model is suitable for both combinational and sequential gates. Our model not only provides high accuracy (close match to MC simulation results), but more important, shows great flexibility against process and voltage supply variations. The calculation of the key parameters of IGD model is very

straightforward which is beneficial for delay estimation of large circuits. In contrast to MCS data, in the form of: DFFs + 8-bit DEMUX and MUX, the average mismatches from our approach is 1.2% while, on the other hand, orders of magnitude simulation time was saved. Moreover, our IGD based estimation shows even higher accuracy than GD based fitting.

3. PDF-based Error Modelling and Reliability Assessment

Abstract: In Deliverable 5.1 **[D5.1]**, we proposed to quantify the high-level voltage figure as a Probability Density Function (PDF). We believe that considering a range of probability values (i.e., a PDF), instead of a sole probability value, is a more appropriate approach to model the faulty circuits stochastic behavior, when the circuit is exposed to multiple correlated degradation phenomena. The theoretical framework of a circuit reliability assessment based on given inputs, and prior PDFs of its comprising components, was presented in Deliverable 5.1. In this Chapter we shall address the afferent practical considerations, notably: (i) the error modeling and degradation quantification theoretical aspects and simulation methodology in order to obtain the prior PDFs of circuit comprising components. For illustrative reasons, we employ as discussion vehicles an inverter and a NAND gate, and (ii) a general practical simulation scenario and preliminary results for ISCAS'85 circuits in order to exemplify the PDF-based reliability assessment of a circuit. We conclude the chapter with some remarks concerning the following work to be undertaken.

Publications: Unpublished Work

Deep sub-micron devices suffer multiple degradations concurrently, due to manufacturing and environmental fluctuations, as well as run-time aging effects. As a result, the device critical physical and electrical parameters exhibit a time dependent drift. The low-level physical parameter variations of a circuit comprising transistors are reflected at the circuit level as performance degradation, such as the increase of the circuit propagation delay. Eventually, the circuit delay degradation can exceed the maximum clock period, and as a consequence, wrong values may be sampled at its output, resulting in an erroneous behavior of the entire computation platform (application). In practice circuits are exposed to a plethora of degradation inducing agents, whose effects are typically correlated. Thus employing a low level parameter for error modeling and degradation quantification, not only poses measurement difficulties, as the run-time measurement has to be non-invasive in order not to disturb the circuit correct operation but also complex error models able to capture the correlation have to be developed, cautioning against the practical utility.

In this line of reasoning, we propose to employ a high-level degradation quantifier, specifically, a voltage related figure, which inherently captures the degradation-induced correlation of the low-level physical parameters variations. Otherwise stated, the high-level voltage related figure, is merely a functional of low-level physical parameters, e.g., threshold voltage, electrons mobility, temperature, etc. Furthermore, as these figures, in order to assess the circuit reliability, are propagated throughout the circuit using the methodology described in Deliverable 5.1 **[D5.1]**, the correlations between the different comprised circuit gates are inherently captured, and thus the correlation of different errors encountered in the circuit is being accounted for.

3.1. PDF-based Error Modelling and Degradation Quantification

As far as the high-level degradation quantifier is concerned, we consider a circuit output voltage whose variations may induce a functional failure (i.e., an incorrect circuit output). In this section, a comprehensive reliability characterization at CMOS gate level is targeted.

For digital CMOS circuits each output can assume one of the two possible logic levels, i.e., logic ”1” and logic ”0”. The circuit error PDF is dependent on multiple factors, such as temperature, threshold

voltage, supply voltage, electrons mobility, etc. Varying these parameters and performing a Monte Carlo simulation, two PDFs can be obtained covering the distribution of the circuit output voltage corresponding to logic ”0” and to logic ”1”, respectively. Specifically, when considering a certain gate, the output voltage is sampled in each Monte Carlo iteration at a time moment equal to the gate nominal propagation delay. The distribution of the measured output voltage for the gate undergoing variations, grouped as the logic ”0” distribution and logic ”1” distribution, is depicted in Figure 3-1. The error probability of interest, i.e., the probability of a logic ”1” being treated as a logic ”0” and vice versa, is represented by the intersection of the two Gaussian curves. Figure 3-2 presents the valid analog output voltage ranges which correspond to logic ”0” and ”1” levels.

According to the voltage output relationship to the power rails (i.e., nominal VDD and VSS), we have: (i) V_{OH-} and V_{OH+} the minimum and the maximum, respectively, voltage values interpreted by the gate as a logic ”1”, (ii) V_{OL-} and V_{OL+} the minimum and the maximum, respectively, voltage values interpreted by the gate as a logic ”0”, and (iii) voltage values in the ”not allowed” region renders the output logically indeterminate.

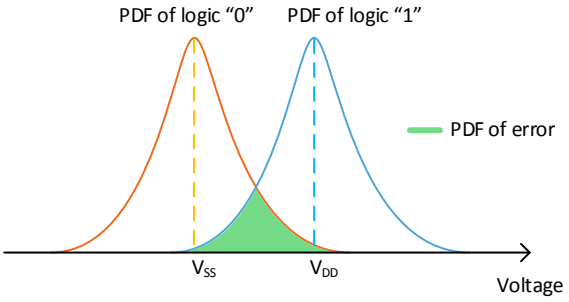


Figure 3-1 – Gate Output Voltage Distribution.

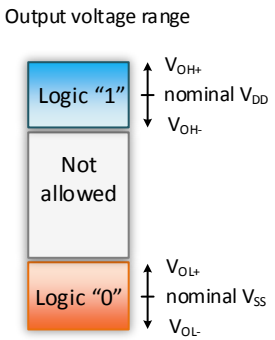


Figure 3-2 – Gate Output CMOS Levels.

At design-time, a circuit is exposed to different aggression profiles in order to characterize it from the reliability point of view. This one-time reliability characterization of a circuit basic building blocks serves as initial belief of how the gates may behave under various conditions which may be encountered at run-time. To appropriately reflect how multiple correlated degradations affect the behavior of the circuits, all-around aspects which could undermine the circuit reliability have been taken into account, including spatial unreliability or process variability, as well as various environmental related aggression profiles (e.g., supply voltage and temperature variations, transient errors). These are reflected as variations of several key parameters of gates/transistors (e.g., threshold voltages, thickness of oxide, effective length, electron mobility, as well as V_{DD} and V_{SS} fluctuations) and ultimately as variations of a high-level circuit voltage figure. In order to propagate the PDFs throughout circuits, firstly, PDFs of generic building blocks, such as inverter, NAND gate, etc., have to be captured.

Figure 3-3 depicts the output voltage of an inverter with Fan-Out of one (FO1), for the charging (“0” -> “1” output switching) and discharging (“1” to “0” output switching) case in two situations: (i) the inverter is exposed to the aforementioned parameters variations, and (ii) when the inverter operates in normal conditions (i.e., the nominal case, without any variations). The sampling time is set to the nominal propagation delay of an inverter gate without any parameters variations. It can be seen that with the involvement of parameters variations, the switching time of the output can be either faster

or slower than the nominal situation. However, this sampling point can be adjusted depending on the requirement of the frequency constraints for instance, and as a result modulating the gate error PDF.

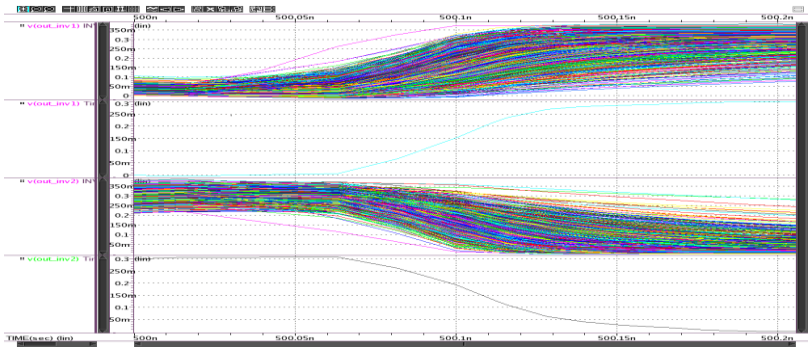


Figure 3-3 – CMOS Inverter Voltage Output With and Without Variation.

The corresponding PDFs of an Inverter and a NAND gate output voltage for the logic “0”, and logic “1” levels, both gates with a fan-out of one, are graphically illustrated in Figure 3-4 and Figure 3-5, respectively.

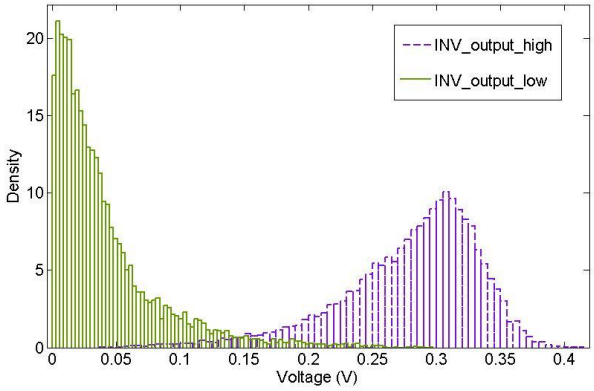


Figure 3-4 – Inverter Output Voltage PDFs

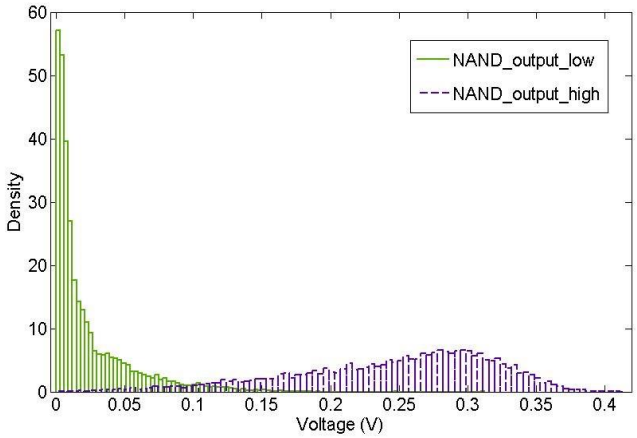


Figure 3-5 – NAND2 Output Voltage PDFs

The voltage data are captured through ten thousand times Monte Carlo simulation in HSPICE where 45nm PTM CMOS process **[PTM45]** is used. All the low-level key aging and degradation-reflecting

parameters such as transistors threshold voltages, gate oxide thickness, effective length, electron mobility, as well as, V_{DD} and V_{SS} values are varied. We consider that all varied parameters follow a Gaussian distribution, the variation details being stated as follows: 1) V_{DD} and V_{SS} both have 30mV variations from their mean nominal value; 2) the transistor age has the mean value of two years and standard deviation of one year and is quantified in the threshold voltage and electrons mobility variations; and 3) all the other elements have a 10% deviation from their own nominal mean value.

One can observe in Figure 3-4 and Figure 3-5 that the PDF profile is highly dependent on the circuit different topology, i.e., the inverter (INV) and the NAND gate. For the inverter, it is much more balanced than for NAND although the variation of charging (purple solid histogram) is still larger than that of the discharging case (green dotted histogram). To some extent, it is due to the aging effect, which mainly affects the PMOS rather than NMOS performance. On the other hand, the charging event in NAND struggles considerably due to the asymmetric structure plus the aging effect.

Furthermore, it can be observed that a small overlapping area occurs for both INV and NAND gate. The overlapping area between the logic “1” output voltage PDF and the logic “0” output voltage PDF, i.e., the error probability of a logic “1” being interpreted as a logic “0”, and vice versa, constitutes the gate error PDF, as previously discussed. With the increase of this region, lower reliability/ higher error probability can be anticipated. Figure 3-6 thereby highlights the overlapping area, i.e., the error PDF corresponding to the INV gate.

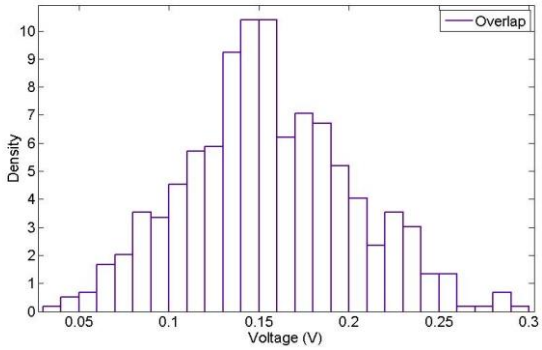


Figure 3-6 – Inverter Gate Error PDF

Following the propagation/evaluation of signals throughout the circuits, those two PDFs may have some overlap (see Figure 3-6), which makes the output no longer hold its certainty at some point. This kind of situation may get worse as the propagation path extends as indicated by Figure 3-7, which presents the output voltage PDFs (two cases: charging and discharging) of a nine inverter chain.

In this case the output is sampled which a period corresponding to nine times the propagation delay of a single inverter. Different to the one INV case, along with the increase of the overlapping area there are also two peaks occurring in this area after propagation through nine INVs, which can be understood as more cases being detected as non-switched cases. From the mathematical modeling point of view, in such a case, it no longer suffices to model the error PDF as a sole Gaussian, and more complicated distributions have to be employed.

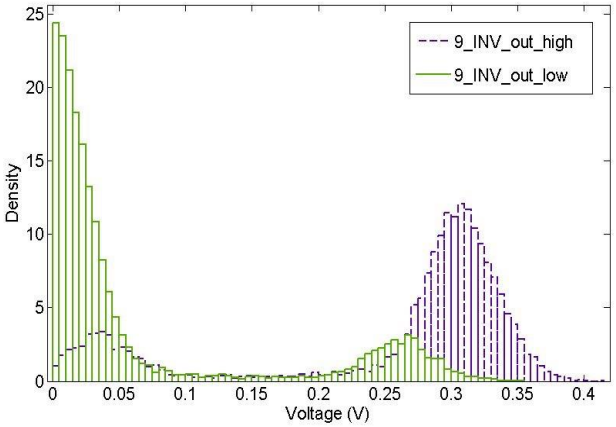


Figure 3-7 – Nine Cascaded INVs Output Voltage PDFs

Despite its versatility, a major drawback of the aforementioned gate voltage based failure PDF approach, is that no information about the gate output voltage value is provided in the case of failure. The output voltage value is different depending on the source of the gate failure. For instance, if we consider a CMOS inverter, and apply a logic ”1” at the input, then the inverter might fail to correctly switch to ”0” if the PMOS transistor fails to switch off, and/or if the NMOS fails to switch on. The inverter output voltage value is different for each failure case (e.g., PMOS on - NMOS off, PMOS on - NMOS off, PMOS off - NMOS off). The gate output voltage information is critical, as it constitutes the input value for the following gates. In view of this, instead of the overlap of the two PDFs (i.e., a gate output probability of failure), a more general and better suited approach would be to consider the superposition of the two PDFs (i.e., the output voltage distribution corresponding to both output logic ”1”, and logic ”0”) as a gate degradation-related figure to be propagated throughout the circuit. In this way, besides the gate failure probability, we propagate also its output voltage values, with an equally complex distribution as in the case of only the overlap distribution, from the mathematical point of view. The final failure PDF at the output of the entire circuit is then determined by the overlapped area of two superimposed PDFs (for logic ”1” and for logic ”0”).

3.2. General PDF Propagation Simulation Setup

Employing the previously discussed reliability characterization of a circuit gates, (each gate having accrued for a specific aggression profile a PDF which reflects its primary output probabilistic error status), we are now in position to address the overall circuit reliability assessment. Subsequently, we first outline the Bayesian network setup afferent to a circuit, specifically, the distribution modelling choices, and then present preliminary simulation results obtained for the PDF-based reliability assessment of some ISCAS’85 circuits.

3.2.1. Bayesian Network Setup

We model a circuit as a Directed Acyclic Graph (DAG), specifically as a Bayesian Network (BN) **[Koller09]**. The graph nodes correspond to circuit gates and wires, and are represented by random variables. The graph edges encode the causal relationships between the nodes, thus accounting inherently for reconvergent fan-outs. For instance, a gate output is conditionally dependent on the values applied at its input. Conversely, the absence of an edge between two nodes signifies their conditional independence (e.g., the state of one node does not directly depend on the state of the

other node). Let $\{T_1, \ldots, T_K\}$ denote the set of all DAG nodes. Then, every DAG node T is a random variable and has associated a PDF, specifically the probability of that node given its ancestor nodes, $p(T|\text{Ancestors}(T))$. The BN afferent to a particular circuit is thus specified by the DAG topology which encodes the circuit internal dependence relationships and by the conditional probability of each DAG node. A Bayesian network induces a probability distribution over its nodes. The joint probability density function of all the DAG nodes $\{T_1, \ldots, T_K\}$ can be factorized using the conditional probabilities of each node, given its ancestors, i.e.,

$$p(T_1, \ldots, T_K) = \prod_{j=1}^{K} p\big(T_j \,\big|\, \text{Ancestors}(T_j)\big). \tag{3-1}$$

As far as the DAG nodes representation is concerned, the following considerations are in order:

Generally speaking, a graph node conditional probability can be either a discrete (a single probability value) or a continuous (a probability density function) random variable. Considering a graph node and its direct ancestors, there are three possible cases: (i) the node, as well as its ancestors are discrete, (ii) the node is continuous but its ancestors are discrete, (iii) the node, as well as its ancestors are continuous, and (iv) the node is continuous and it has both discrete and continuous ancestors. For our purpose, we consider the more general case when both the node and its ancestors are represented by continuous random variables (PDFs).

A second aspect to consider is related to the DAG nodes conditional distributions. There are several possible parametric family choices for representing the conditional probability distributions $p\big(T_j \,\big|\, \text{Ancestors}(T_j)\big)$.

Under the assumption of normally distributed continuous DAG nodes, the simplest and the most straightforward family of conditional probabilities are the Linear Gaussian models. **[Dey00]** In this model, if a node T ancestors are given by a set $\text{Ancestors}(T) = \{U_1, \ldots, U_i\}$, then:

$$p(T \mid \text{Ancestors}(T)) \sim \mathcal{N}\left(T \,\middle|\, \mu + \sum_{j=1}^{i} w_j\big(U_j - \mu_j\big), \sigma\right), \tag{3-2}$$

where each graph node T is a single continuous random variable having a Gaussian distribution, μ is the unconditional mean of T, w_j are real coefficients which determine the influence of U_j on T, and σ is the conditional variance of T given its DAG ancestors. In other words, T is normally distributed around a mean that linearly depends on the ancestors' values. However, using a single Gaussian distribution to model a node conditional probability is restrictive and cautions against its practical relevance. Different low-level physical parameters with degradation-induced variations may not all follow a Gaussian distribution and may induce non-linear dependence relationships. Thus a gate output PDF can significantly deviate from the Gaussian distribution. In view of this, we model the distribution afferent to each DAG node as a multivariate Gaussian distribution. A DAG node $\boldsymbol{T} = \{T_1, \ldots, T_p\}$ has a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if:

$$p(\boldsymbol{T} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \cdot e^{-\frac{1}{2(\boldsymbol{T}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{T}-\boldsymbol{\mu})}},$$

with $\boldsymbol{\mu}$, the p-dimensional mean vector and $\boldsymbol{\Sigma}$, the $p \times p$ positive definite covariance matrix. Specifically, the node PDF is expressed as a linear combination of multiple Gaussian distributions, each with its own mean and covariance. By using a sufficiently large number of Gaussian components, and by adjusting each Gaussian component mean, and covariance (and possibly its mixing coefficient in the linear superposition), almost any continuous distribution can be approximated with an arbitrary accuracy. The joint PDF of a bi-variate Gaussian distribution, and its two marginal distributions are illustrated in Figure 3-9 and Figure 3-8, respectively.

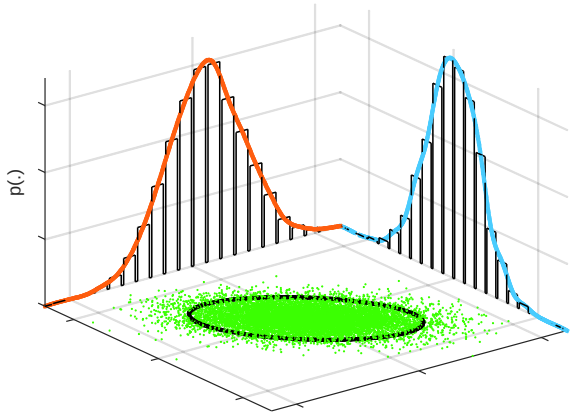


Figure 3-8 – Bi-variate Gaussian Distribution Marginal PDFS

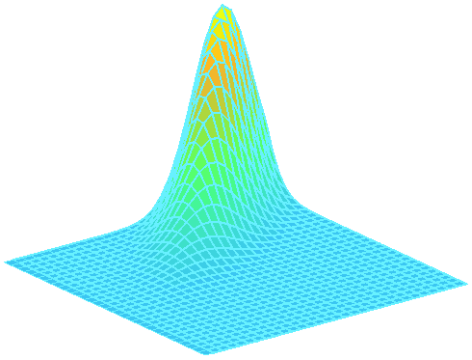


Figure 3-9 – Bi-variate Gaussian Distribution Joint PDF

The linear Gaussian model can be easily extended to the case of interest for circuit scenarios, in which the graph nodes represent multivariate Gaussian variables. In such a case, the conditional distribution for node $\boldsymbol{T}$ in (3-2) becomes:

$$p(\boldsymbol{T} \mid \text{Ancestors}(\boldsymbol{T})) \sim \mathcal{N}\left(\boldsymbol{T} \middle| \boldsymbol{\mu} + \sum_{j=1}^{i} \boldsymbol{W}_j\left(\boldsymbol{U}_j - \boldsymbol{\mu}_j\right), \boldsymbol{\Sigma}\right), \tag{3-3}$$

where $\boldsymbol{W}_j$ is a matrix (non-square if the $\boldsymbol{T}$ and $\boldsymbol{U}_j$ have different dimensions). As concerns the prior joint distribution of $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for each node, we assume it Gaussian-Wishart, as this is the conjugate prior for a multivariate normal distribution. Specifically for a p-dimensional node $\boldsymbol{T}$, we have:

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{k=1}^{p} p(\mu_k \mid \Lambda_k) \cdot p(\Lambda_k) \sim \prod_{k=1}^{p} \mathcal{N}(\mu_k \mid m, (\lambda\Lambda_k)^{-1}) \times \mathcal{W}(\Lambda_k \mid S_k, \upsilon), \tag{3-4}$$

where m, λ, S_k, and ν are the Gaussian-Wishart distribution parameters. The latter two parameters govern the Wishart distribution, while λ is a normalization parameter, and m denotes the mean for the Gaussian distribution. In (3-4), $\Lambda_k = \Sigma_k^{-1}$ denotes the precision matrix, which is the inverse of the covariance matrix Σ. Working in terms of precision matrix, rather than covariance matrix, simplifies the mathematical derivations to a certain extent. We have:

$$p(\mu_k|\Lambda_k) = \frac{\lambda^{\frac{1}{2}}|\Lambda_k|^{1/2}}{(2\pi)^{\frac{p}{2}}} \cdot e^{-\frac{1}{2}(\mu_k - m)^T \lambda \Lambda_k (\mu_k - m)} = \frac{\lambda^{\frac{1}{2}}|\Lambda_k|^{\frac{1}{2}}}{(2\pi)^{\frac{p}{2}}} \cdot e^{-\frac{1}{2}\mathrm{Tr}[\lambda \Lambda_k (\mu_k - m)(\mu_k - m)^T]}, \quad (3\text{-}5)$$

and

$$p(\Lambda_k) = R(S_k, \upsilon) \cdot |\Lambda_k|^{\frac{\upsilon_k - p - 1}{2}} \cdot e^{-\frac{1}{2}\mathrm{Tr}(S_k^{-1}\Lambda_k)}, \quad (3\text{-}6)$$

where:

$$R(S_k, \upsilon_0) = |S_k|^{-\upsilon/2} \cdot 2^{-\frac{\upsilon p}{2}} \cdot \pi^{-\frac{p(p-1)}{4}} \left(\prod_{i=1}^{d} \Gamma\left(\frac{\upsilon + 1 - i}{2}\right) \right)^{-1},$$

and the mean of Λ_k is υS_k^{-1}.

Having discussed the distribution choices, we are now in position to further outline the considered simulation scenario.

3.2.2. Circuit Afferent Bayesian Network

For a given circuit, we discriminate the graph nodes into the following three sets:

- E, the set of evidence nodes, which are associated to the circuit primary inputs;
- Y, the set of latent (hidden) nodes, which are associated the circuit primary outputs; and
- X, the latent (hidden) intermediary nodes, which correspond to the remaining circuit nodes.

For illustration purposes, in Figure 3-10 and Figure 3-11, we depict the DAG afferent to the c17 ISCAS'85 circuit.

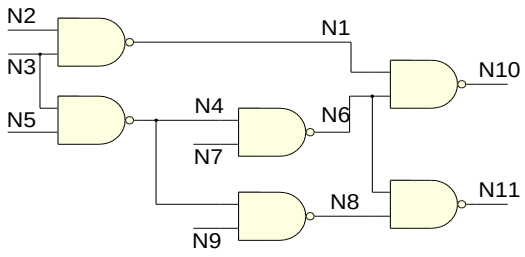


Figure 3-10 – ISCAS C17 Gate Level.

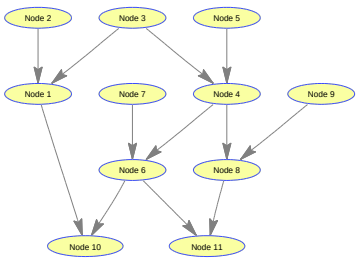


Figure 3-11 – ISCAS C17 DAG

Thus, given the PDFs of the evidence nodes from E (circuit primary inputs) and the prior PDFs of the intermediary nodes from set X (i.e., the initial belief about the gates/wires PDFs, which is obtained as presented in Section 3.1, we are interested to infer the PDFs of the primary output nodes in set Y. The first step in deriving the optimization bound is to divide the DAG nodes into nodes to be marginalized, i.e., to be integrated out, and nodes to be parameterized, i.e., the lower bound nodes to be optimized. The lower bound is then derived, and the gradient with respect to the mean and precision parameters is computed. At this stage, the optimization is carried out in the Euclidian space. Further discussion will be given, as the follow-up work, which targets the optimization in the Riemannian space, as allowing the optimization to be performed on a restricted space when

compared to the Euclidian space, enables advantages in terms of convergence speed and solution accuracy.

As the first primary objective at this stage is the PDF inference validation, instead of the Monte-Carlo derived PDFs, as described in the previous section, we employed synthetic data as prior distributions. As test vehicles, we used the ISCAS'85 circuits, a summary of whose topology statistics is presented in Table 3-1. For each circuit, we derived its DAG, initialized the model with synthetic data and applied both the Variational Bayes inference method, which serves as comparison reference, and the proposed variational inference method, with Polack Ribiere conjugate gradient.

Table 3-1 – ISCAS'85 Test Circuits

Circuit	# Gates	# Inputs	# Outputs	# BN nodes
C432	160	36	7	196
C499	202	41	32	243
C880	383	60	26	443
C1355	546	41	32	587
C1908	880	33	25	913
C2670	1193	233	140	1350
C3540	1669	50	22	1719

Figure 3-12 summarizes the convergence time for both the proposed and the reference inference method. The figures are obtained as an average of 50 restarts (a restart accounting for a new set of prior distributions).

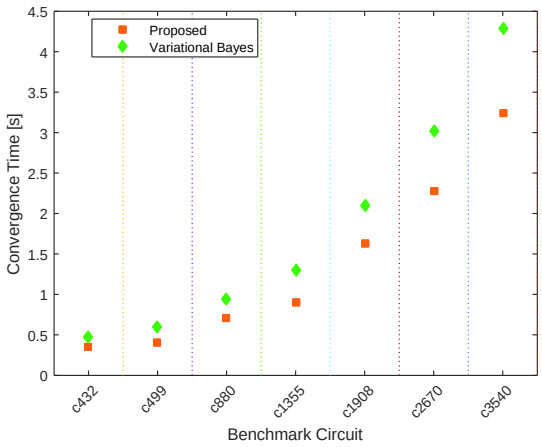


Figure 3-12 – Convergence Analysis

3.3. Conclusion

In this chapter, we have addressed practical implementation details concerning the theoretical framework (introduced in Deliverable 5.1) towards circuit reliability assessment based on given inputs, and prior PDFs of its comprising components.

Specifically, we have first addressed a high-level error modeling and degradation quantifier, and discussed its feasibility both from a theoretical and practical, SPICE simulation based perspective. For the purpose of illustration, we have presented the PDF based characterization of two gate types: inverter and NAND gate.

Then, we have outlined a general simulation setup of the PDF-based circuit reliability assessment and applied it for the reliability evaluation of some ISCAS'85 circuits. For conciseness and simulation purposes, we employed synthetic data as gates prior PDFs. However, this setup can be also applied when using prior gate PDFs obtained by means of HSPICE simulation.

As future work we plan to evaluate the PDF-based reliability assessment approach by comparing the derived output PDFs against Monte Carlo SPICE simulation measured circuit primary output voltages (more precisely, the afferent PDFs of logic "0" and "1") of the entire circuit. The follow-up work also includes the SPICE-based Monte Carlo characterization for other gate types, the optimization in the Riemannian space implementation, as well the augmentation of the test circuits set.

4. Multi-Level Simulated Fault Injection for Reliability Analysis of Register Transfer Level Circuit Descriptions

Abstract: In this chapter, we present data dependent reliability assessment methodology for digital systems described at Register Transfer Level (RTL). The proposed method uses a hierarchical approach, which uses Gate Level (GL) data dependent Simulated Fault Injection (SFI) for the reliability metric extraction of building blocks and RTL simulation for system level analysis. This way, we aim at approaching the accuracy of the GL SFI, while maintaining the simulation overhead specific to RTL based evaluation. The methodology has the following phases: correct simulation for a specific set of inputs of the RTL description, which aims at capturing the inputs for each of the component, hierarchical block decomposition, which splits the RTL designs in simple building blocks, logic synthesis of the components obtained after the previous step, data dependent SFI of the GL netlists and the RTL SFI using the probabilities derived in the previous step. We have validated our methodology for a 128-bit Advanced Encryption Standard (AES) crypto-core, for which the GL simulation could not be performed.

Publications: To be submitted for publication

4.1. Motivation

Fault injection techniques are frequently used for the reliability evaluation of circuits. They can be performed at each level of abstraction of a digital system: circuit level, GL, RTL and functional level. GL SFI provides accurate results, as it captures faithfully data dependency; however, it becomes computationally prohibitive (in terms of simulation time and required memory) for complex systems. Performing SFI at higher levels of abstraction, such as RTL, requires order of magnitude lower simulation times with respect to GL SFI. Furthermore, high level descriptions are available in earlier design phases. Thus, changes in the design after SFI are easier to perform **[Maniatakos09].** Combining GL and RTL SFI could represent a trade-off solution between accuracy and computational resources (simulation overhead and memory requirements).

In the Deliverable 2.1 **[D2.1]**, corresponding to the work accomplished during the first year of the project **[Amaricai14]**, we have proposed probabilistic data dependent GL SFI. The proposed methodology relied on a mutant based approach for fault injection. Four types of fault models have been proposed:

- Gate Output Probabilistic (GOP) - for this model, the gate implements the correct logic function with a given probability;
- Gate Output Switching probabilistic (GOS) – for this model, the probabilistic behavior of the occurs only when the gate switches; this models considers that the logic gate cannot perform the corresponding switching in a given amount of time;
- Gate Output Switching Type probabilistic (GOST) – two probabilities are considered for this model, for both charging and discharging processes; this model take into account the different driving strength of the nMOS and pMOS stages;

- Gate Input Switching Probabilistic (GISP) – this model associates a probability of correct switching for each input transition;

These models have been used for the analysis of 6-bit ripple carry adders, carry-select adders, as well as ripple carry adders protected using triple modular redundancy. The obtained simulation times where reasonable for these small and medium circuit. However, when analyzing the behavior of complex digital systems, GL simulation becomes unfeasible. Therefore, evaluation at higher abstraction layers (such as RTL) has to be addressed. One issue regarding high level simulation is represented by the extraction of data dependency. Therefore, we propose a multi-level simulation approach, which use GL simulation for simple building blocks in order to capture the data dependency for these blocks, while probabilities obtained for these components are used for RTL SFI.

4.2. RTL Simulated Fault Injection

RTL SFI have been used for reliability analysis of digital systems, a wide range of works concentrating on developing SFI components for RTL descriptions or improving the simulation overhead of RTL based analysis **[Bombieri11][Gil08]**. The RTL based SFI has been used either for testing purpose **[Thaker00]** – to derive the fault coverage in early design phases of specific test vectors - , either for reliability assessment purposes **[Baraza05] [Baraza08] [Maniatakos09]**.

Regarding the SFI components for RTL, two types of approaches have been proposed. One approach is based on altering the signals within the RTL design **[Thaker00].** In this case, the modification of behavioral statements is not considered. A second approach is based on the altering the behavioral components of the RTL descriptions **[Baraza08][Gil08]**. These include: replacing the values of conditions in *if* and *case* statements (*stuck-then, stuck-else, dead process, dead clause*), disturbing assignment statements (*assignment control, global stuck-data*), or disturbing operators in expressions (*micro-operation, local stuck-data*), etc. They can model in an accurate way simple faults, such as stuck-at faults. However, these approaches cannot be used to accurately model transient type or errors or probabilistic faults.

A multi-level approach is proposed in **[Evans13]**. It aims at developing reliability analysis for single-event upsets (SEU) type of errors. It uses GL simulations for building blocks in order to derive appropriate fault models for the RTL simulation. However, this approach does not take into account the input stimuli. Although the SEU fault model does not considers data dependency, the input stimuli represent an important parameter in SFI campaigns, as errors may be masked by different input combinations.

The proposed approach uses the hierarchical analysis in the reliability assessment. A key feature of our methodology is represented by the data dependent analysis. We perform two RTL simulations: a correct simulation, which is used to extract input set for each block, and one simulation with fault injection components inserted. This way, we capture in accurate way the data dependency for the RTL SFI.

4.3. Data Dependent Multi-Level Methodology

We have developed a simulated fault injection based reliability evaluation methodology, which tries to combine the accuracy characteristic to the gate level analysis and the computational requirements (memory and simulation time) of the RTL. The proposed technique is based on a multi-phase simulation:

- block based SFI performed for GL netlists
- system level simulation performed at RTL

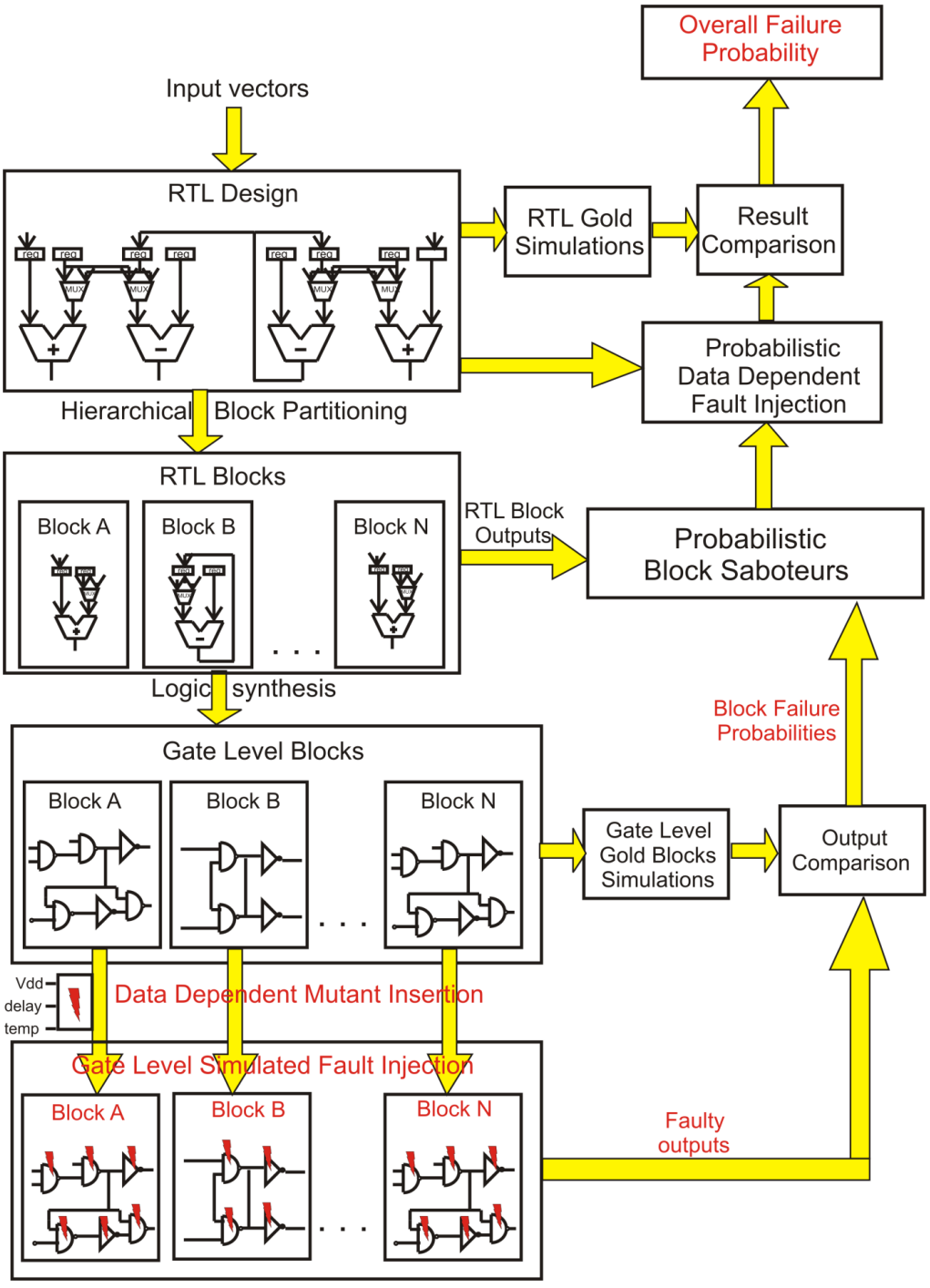


Figure 4-1 – Multi-level simulated based reliability evaluation methodology

The gate GL analysis uses the approach and the data dependent FI components developed during the first year of the project (presented in Deliverable 2.1 **[D2.1]**). The RTL analysis uses probabilistic saboteurs which have been derived based on the GL SFI. This technique has been implemented using Verilog and consists of 5 main phases:

- Hierarchical block decomposition
- RTL correct simulation

- Logic synthesis
- Data dependent GL SFI
- Probabilistic RTL SFI

Figure 4-1 depicts the developed reliability evaluation process for RTL circuit descriptions.

4.3.1. Hierarchical block decomposition

This phase has the goal of partitioning the RTL systems in blocks of low complexity, in order to allow GL simulation. We also go through the module hierarchy of the targeted design. In order to further simplify the gate level analysis, the obtained components are either fully combinational, either composed of only storage elements.

4.3.2. RTL correct simulation

The second phase is represented by the RTL correct simulation with a given input set of stimuli for the entire system. The goal of this simulation is to extract the inputs and the correct outputs for each block obtained after the first phase. The block level inputs will be used in the GL data dependent analysis. The correct outputs for each block will be used in the development of saboteur components used for RTL SFI.

4.3.3. Logic synthesis

Logic synthesis is performed for each block obtained after the first phase. This way, the GL netlist for each system component is obtained. The reliability measures for the resulted netlists will be derived using GL SFI.

4.3.4. Gate level data dependent SFI

Each component of the netlist (logic gate or storing element) is mutated according to one of the four probabilistic fault models defined during the first year of the project: GOP, GOS, GOST or GISP. We are using a mutant based analysis, with each gate mutated according to one of the four fault models. The set of inputs for each block has been extracted during phase 2. We obtain probability of failure for each output signal of the considered block, by comparing the results of the GL SFI with the correct outputs extracted during phase 2.

4.3.5. RTL saboteur based SFI

The last phase consists of the development of RTL probabilistic saboteurs. The probabilities for the saboteurs have been derived during the previous phase. These saboteurs are used in order to perform the RTL SFI of the entire system. The final failure probability is obtained by comparing the outputs of the saboteur based SFI with the correct outputs derived during phase 2.

4.4. Case Study: Multi-Level SFI for 128-bit AES Crypto-Core

We have used the proposed methodology in order to analyze a 128-bit AES Crypto-Core, available as open-source on the OpenCores platform **[OpencoresAES]**. The analysis of circuits used for cryptographic application is important, as side channels attacks based on reliability evaluation under different conditions is common. Furthermore, circuits which implemented cryptographic functions are considered for proof-of-concept design within the i-Risc project.

Regarding the AES operation, the plain text for algorithm is represented by the initial 128-bit state, which is modified by the round transformation and becomes the final state, which represents the output cipher text. The state is organized as a 4*4 matrix of bytes and the round transformation scrambles these bytes either individually, row-wise or column-wise by applying the functions *SubBytes, ShiftRows, MixColumns* and *AddRoundKey* sequentially. The function *SubBytes* is the only non-linear function in AES, which substitutes all bytes of the state using table lookup, which is often called S-box. The *ShiftRows* function rotates the rows of the state by an offset, which equals the row index. The MixColumns function accesses the state column-wise and interprets a column as a polynomial over GF(2^8). The *AddRoundKey* function adds a round key to the state, a new round key being derived in every iteration from the previous round key **[Feldhofer05]**.

The architecture of the AES crypto-core is depicted in Figure 4-2.

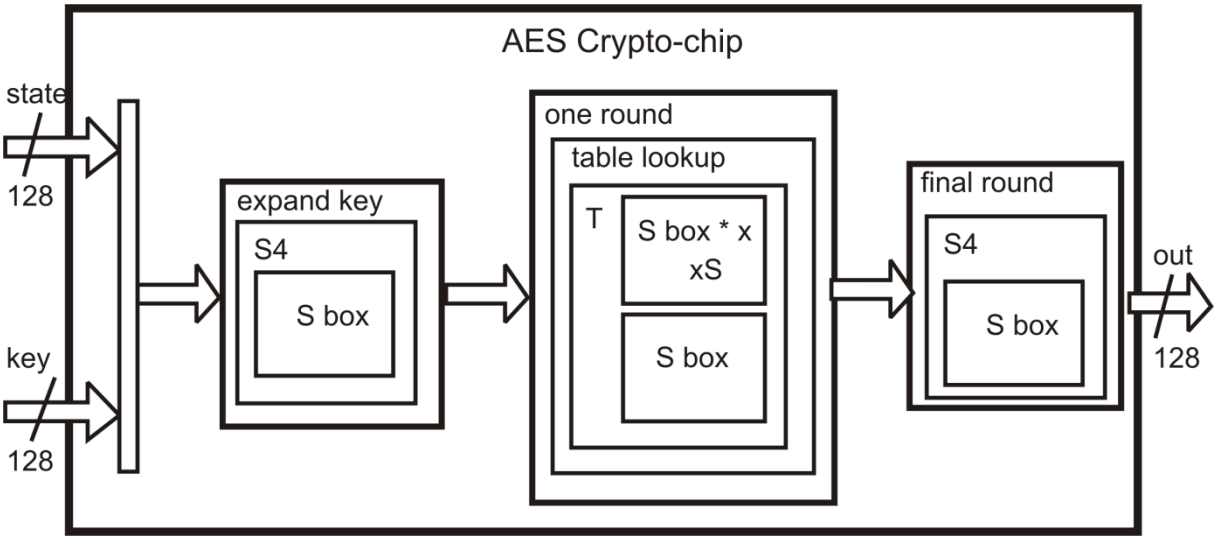


Figure 4-2 – Architecture of the 128-bit AES core

Regarding the complexity of the analyzed core, the synthesis results for Xilinx Spartan-6 LX45T FPGA using the Xilinx ISE 14.4 software presented the following estimates:

- 5792 out of 54576 slice registers (10% of the total capacity),
- 10992 out of 27288 slice LUTs (40% of the total capacity),
- 29 out of 166 block RAM (25% of the total capacity).

The first phase of the methodology has consisted in partitioning the AES circuit in 9 functional blocks; each block has been further divided in combinational and sequential sub-blocks. The 9 blocks and their functions are as follows:

- block A – the AES crypto-chip top modules, which receives the 128-bit key and the 128-bit state as inputs, performs an exclusive-or on the two vectors and instantiates blocks B, C and D;
- block B – referred as “expand key” in Figure 4-2, performs the expansion operation on the 128-bit key and instantiates block E;
- block C – referred as “one round” in Figure 4-2, performs XOR operations on the key bytes and instantiates block G;
- block D – referred as “final round” in Figure 4-2, instantiates block E;
- block E – referred as “S4” in Figure 4-2, substitutes four bytes in a word by calling 4 times the block F module;
- block F – referred as “S” or “S-box” in Figure 4-2, performs a table lookup operation;
- block G – referred as “table lookup” in Figure 4-2, uses the results provided by block H instances
- block H – referred as “T” transformation in Figure 4-2, uses the results provided by blocks F and I instances;
- block I - referred as “xS” in Figure 4-2, is similar to block F, performs a table lookup operations;

The second phase consists of RTL correct simulation of the entire system for a specific set of inputs. For each combinational and sequential sub-blocks of a certain block, the input vectors and the outputs have been extracted; they have been stored in files containing the correct results for that corresponding block, for each run. Also, during this step, the simulation of block *i*, which contains at least one instance of block *i+1,* generates two files associated to block *i+1:* one contains the input vectors applied to all the instances of block *i+1;* the other one contains all the correct outputs which correspond to those inputs.

The third phase has been represented by logic synthesis, which has been performed for each of the nine blocks in the design. The resulted netlists contained only 2-inputs NAND gates for the combinational part and D flip-flops for the sequential part. We have used in the synthesis process, Synopsys Design Compiler and the ABC synthesis tools in order to generate mapped netlists of those modules, represented in terms of inverters, NAND gates, NOR gates and registers. Each inverter and NOR gate has been then implemented using only NAND gates.

The mutants inserted for each NAND gate during the fourth phase implement the input data dependent model – GISP model, which is the most accurate. This model uses 4 probability values, one for each input transition which leads an output transition. SFI has been performed on these mutant-based netlists in order to determine the probability of failure of the blocks situated at the bottom of the design, by comparing the faulty outputs with the correct ones.

The last phase and it consisted in the development of RTL saboteurs and the RTL SFI. This phase used a bottom-up approach in deriving the probabilistic saboteurs for each component. The probabilities obtained at one level in the hierarchy have been used in order to construct the SFI components for the next level in the hierarchy. RTL SFI has been applied for each level of module hierarchy.

4.4.1. Simulation results

The simulations have been performed using Modelsim 10.05 SE commercial simulator on a desktop computer with Intel Core i5 processor at 3.1 GHz and 4 GB of RAM with Windows 7 operating system.

The resulted gate-level AES crypto-chip design contains approximately 1,100,000 instances of NAND gates and D flip-flops. The gate level implementation of the analyzed core could not be simulated on the computing platform described above, due to insufficient memory.

The main goal of these simulation campaigns is to show the flexibility of the proposed methodology and to measure the simulation overhead required, respective to the gold circuit. The input parameters considered for the mutant insertion are depicted in Table 4-1.

Table 4-1 – Input Parameters for Gate-Level Mutant Insertion

Input parameters	Vdd (V)	Delay (ns)	Temp (°C)	Fault model	Average Probability of failure
NAND Gate	0.30	3.00	50	GISP	0.3314%
D Flip-flop	0.30	2.50	50	GISP	0.1251%

The average probability of failure for each of the 9 modules that compose the AES design, along with the associated simulation time, are depicted in Table 4-2. Due to the hierarchical structure of the design, the 100 input vectors of the crypto-chip can generate thousands or tens of thousands of input vectors for the blocks situated at the bottom of the design. The exact number of input vectors for each block of the design is shown in Table 4-2, column 2. We have monitored the simulation time per run. One run represents the simulation of a block for one set of input vectors. Therefore, the number of input vectors in column 2 of Table 4-2 is equal to the number of runs performed for a certain block. The total simulation time for a block is equal with the number of inputs multiplied to the simulation time of each run.

Although the probability of failure of a single logic gate or flip-flop is extremely low, the resulting probability of failure of one block is quite high, due to the prevalence of XOR operations and bytes scrambling required by the AES algorithm, which facilitates the propagation of faults. Analyzing the results in Table 4-2, we can conclude that the probability of failure of a block increases as we move from the bottom to the top of the design, a fact justified by the increased complexity of upper blocks, which use multiple instances of the lower blocks.

Table 4-2 – Simulation Results for AES System and Its Components

Module	No. Of input vectors	Output width	Components	Probability of failure	Simulation time	Simulation type
Block I - xS	85435	8	-	9.1250%	33 ms / run	gate level
Block F - S box	3952	8	-	9.0429%	27 ms / run	gate level
Block H - T	85436	32	1 * block F 1 * block I	11.1357%	51 ms / run	RTL
Block G - table_lookup	3560	128	4 * block H	11.2219%	105 ms / run	RTL
Block C - one_round	900	128	4 * block G	33.9910%	140 ms / run	RTL
Block E - S4	1000	32	4 * block F	9.0721%	88 ms / run	RTL
Block D - final_round	100	128	4 * block E	10.0221%	4 ms / run	RTL
Block B - expand_key_ 128	1000	128	1 * block E	12.8349%	5.8 ms / run	RTL
Block A - AES	100	128	10 * block B 9 * block C 1 * block D	50.0625%	93 ms/ run	RTL

Regarding the simulation time, the entire simulation campaign (consisting of the simulation at gate level and RTL) has taken 131 minutes. This represents a reasonable simulation time for the design containing more than 1 million gates.

4.5. Conclusion

We have developed a multi-level SFI methodology for data dependent reliability analysis of RTL descriptions of digital systems. The proposed methodology uses a hierarchical approach: it uses gate level data dependent mutant SFI in order to derive RTL based SFI components; the reliability estimation of system is obtained using RTL SFI. The proposed approach capture the data dependency using gate level simulations for blocks of small complexity, which are then used in the RTL data dependent reliability analysis. The proposed methodology consisted on five phases: hierarchical block decomposition, RTL correct simulation, logic synthesis, gate level SFI and RTL SFI. We have applied the proposed methodology on a complex system, an open-source implementation of a 128 bit AES crypto-core consisting of more than 1 million logic gates. The gate level simulation was not possible on the computing platform used, due to high memory requirements. Using the proposed methodology, the simulations have taken 131 minutes.

5. Cost Effective FPGA Emulated Fault Injection for Probabilistic Faults

Abstract: This chapter presents a cost effective fault emulation technique for circuits affected by probabilistic noise. The problem it addresses is how to efficiently inject faults in many locations within a Circuit Under Test/Design Under Test (CUT/DUT). For this purpose, the Emulated Fault Injection (EFI) components proposed are a trade-off between the desire for speed/performance and the inherent physical board limitations of the Field Programmable Gate Arrays (FPGA). The proposed method also allows exploring the best option for this trade-off with minimal effort. The proposed solution allows enough flexibility to be able to deal with the different EFI architectures selectable by minor code intervention. An analysis of the overhead for the EFI components for various number of fault locations has been provided. A case study of two ISCAS benchmark circuits in order to test our methodologies and to highlight the differences for combinatorial and a sequential circuits is presented. It is shown that the number of fault locations can be increased more than 20 times with similar overhead than other state of the art methods reported in the literature. Furthermore, we investigate the possibility of applying the proposed EFI approach for data dependent probabilistic faults.

Publications: This work has been published in:

O. Boncalo, A. Amaricai, C. Spagnol, E. Popovici "Cost Effective FPGA Probabilistic Fault Emulation", Proceedings of the 32nd NORCHIP Conference, Tampere, Finland 2014

5.1. FPGA Fault Emulation

FPGA EFI represents an alternative to the Simulated Fault Injection (SFI) in order to increase the performance/speed of the fault analysis. There are two different methods for FPGA fault emulation **[Lopez07]**:

- Altering the FPGA configuration file according to the fault model
- Inserting dedicated modules which emulate the fault behavior

Regarding the former approach, it is well suited for permanent faults. For probabilistic errors, altering the configuration file may prove inadequate. It would require multiple configuration files, while FPGA reconfiguration is a time consuming operation.

Regarding the latter approach, the main disadvantage is represented by the resource overhead which it brings. This overhead is dependent on two factors: number of fault locations and desired accuracy. Inserting dedicated modules has been applied for soft errors emulation and probabilistic noise emulation. Regarding the probabilistic noise emulation, this approach requires the usage of Random Number Generators (RNG)**[May12]** , either Pseudo (PRNG) or True (TRNG).

An FPGA fault emulation scheme consists of four components:

- Circuit/design under test (CUT/DUT)
- Fault generation and emulation module
- Stimuli generation module

- Result analysis and observation circuit

Regarding the former, the result analyzer and observation logic may communicate with software running on a host PC **[Civera02][Sauer11]**, or may ran autonomously on the FPGA **[Lopez07][Shirazi13]**. The amount of communication varies amongst solutions from only some commands and configuration values from the host PC **[May13]** to the entire Fault Injection(FI) campaign being prepared on the PC side and data downloaded to the FPGA **[Civera02][Sauer11].** Regarding strictly result analysis, it can be either realized on the FPGA board **[Lopez07]**, or at the cost of more communication overhead it can be computed on the PC **[Civera02]**. All, of these are in fact trade-offs between how much extra logic one can afford on the FPGA, the desired EFI configurability, and how much performance (i.e. emulation speed) is targeted. Communication FPGA-host PC typically decreases emulation speed, while providing better configurability for the emulation campaigns and a more powerful support for results processing **[Lopez07]**.

The fault generation and insertion can be achieved in two different approaches. The first approach uses a fault generation circuitry for each fault location. The advantage of this approach is represented by its high emulation speed, as each clock cycle is obtained a new value for each fault location. For probabilistic faults, this approach has been used in **[May12][May13]**. The probabilistic errors have been generated using Linear Feedback Shift Register (LFSR) as RNG. This approach presents the following disadvantages:

- High resource overhead – This is due two factors. On one hand, one LFSR is used for each fault location. Therefore, the number of LFSRs used in the emulation scheme is equal with the number of fault locations. On the other hand, when having multiple fault locations, dedicated circuit to insert different seeds for each LFSR is required; this circuitry is required in order to avoid strong correlations between different fault locations.
- Correlation between faults – Because LFSR is PRNG, correlation between errors in successive clock cycles exists.

In order to reduce the resource overhead of the approaches which use error generation circuit for each fault location, the idea of using shift registers (similar to scan chains used for testing purposes) has been proposed in **[Civera02][Lopez07][Sauer11]**. This type of fault insertion into the DUT introduces the error bits to their fault location in a serial manner. The approach's advantage is represented by its low cost; the main disadvantage is represented by higher emulation speed.

5.2. Fault Emulation Framework

The proposed EFI approach addresses the injection of probabilistic faults in many locations (several thousands) by the usage of a low cost infrastructure. Furthermore, we target truly uncorrelated fault generation and insertion. Figure 5-1 depicts the developed EFI infrastructure. Besides the CUT, it consists of:

1. EFI Fault Generator and Control – the role of this module is to generate fault bits and to insert them in the corresponding fault locations; the fault insertion is achieved using chains of shift registers.
2. Autonomous testbench – the role of this module is to provide the test vectors (inputs for CUT), the error-free outputs and the result comparison and error rate computation logic

3. Observation logic – this module allows reliability metrics (such as failure rates) monitoring, as well as parameter changes for several EFI campaigns.

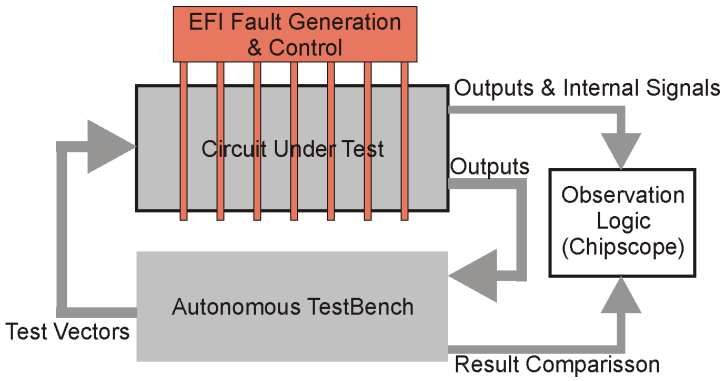


Figure 5-1 – FPGA Emulation Framework

In this subsection, we will describe the autonomous synthesizable testbench and the observation logic. The EFI error generation and control will be detailed in the next section.

The observation logic is based on the Xilinx Chipscope Pro logic analyzer **[Xilinx11].** The dedicated cores for this module are: ILA – Integrated Logic Analyzer –, which allows signal observation and triggers, ICON – Integrated CONtroller –, which provides the communication between Joint Test Action Group (JTAG) interface and the ILA core, and VIO – Virtual Input/Output –, which provides the interface to monitor and to drive signal from the testbench. The main reason behind the Chipscope usage is that it provides cores for monitoring, triggering and driving signals that are optimal in terms of cost and performance for the Xilinx FPGA devices.

The autonomous testbench provides the stimuli generation for the CUT and results processing phase of the EFI campaign. The result processing involves the comparison between the correct gold outputs and the outputs of the fault injected circuit. Regarding the result comparison and analysis, three approaches may be used:

1. Implementing the error-free version of CUT and running in parallel with the injected version **[May12]** – this options is unfeasible for complex circuits, due to the high cost overhead introduced.
2. Duplicating the sequential elements in the design **[Ejlali08]** – this option is valid when the injected faults affect only the flip-flops, while the combinational logic is error-free
3. Storing correct outputs in either FPGA block RAM or external memory (the board memory) – in this case, additional logic for synchronization and memory controller (when external memory is used) is required.

In our FPGA EFI framework, we have used the memory based solution. In the experiments involving the c499 **[Hansen99]** and s1196 **[Brglez89]** circuits, the correct output memory is implemented with FPGA's block RAM.

Conducting an emulation campaign involves the following phases:

1. Setup phase – This phase involves the design of the EFI framework, according to the required fault locations. The following steps are performed: (i) insertion ofXOR gates at fault locations (ii) insertion and configuration of Chipscope modules (iii) autonomous testbench development (iv) insertion of the TRNGs and shift registers.
2. Emulation phase – This phase implies the running of the autonomous testbench on the FPGA device. For probabilistic faults, an EFI campaign consists of at least two orders more than the precision of the smallest probability (e.g. for a probability of 1 in 100 (or 1%), at least 10000 experiments are required).
3. Result processing phase – The injected DUT outputs are compared with the error-free outputs. So far, we have only been interested in accounting the number of failures. In addition to this, signals from several design points have been observed using the ILA core facilities.

The setup phase requires the modification of the DUT and the insertion of the fault emulation infrastructure within the DUT. The actual reliability analysis is performed during the emulation phase.

5.3. Fault Insertion Infrastructure

The proposed fault insertion infrastructure is based on the two major components: error generation module and error insertion module.

The fault generation module is comprised of a RNG and a comparator circuit. The comparator compares the output of the PRNG (on a *p* number of bits) with a constant value. The constant is obtained offline by multiplying the error probability with maximum unsigned integer on *p* bits. We have emulated independent probabilistic faults. In order to accurately emulate these types of faults, one major goal has been to model accurately uncorrelated errors. Therefore, we have used a TRNG for random number generation. Due to the area usage constraints and good randomness we have selected for our implementation the solution proposed in **[Baetoniu08]** by Xilinx. It consists of a XOR based ring oscillator and a Linear Hybrid Cellular Automata (see Figure 5-2). According to **[Baetoniu08]**, the generator has passed most of the DieHard randomness tests used for cryptographic applications.

The fault insertion in the proposed approach is performed using a shift register. This fault emulation scheme is depicted in Figure 5-3. The length of the shift register in this case is equal with the number of fault locations. The simplest implementation is when all fault locations have the same probability of error. Having fault locations with multiple error probabilities requires multiplexing the constants which are compared to the output of the TRNG. This leads to more complicated control logic for the fault insertion. The main drawback of this serial approach (which uses a single shift register for all fault locations) is represented by the large number of clock cycles required to load the fault bits. In order to have uncorrelated errors, the for each cycle of actual simulation, a number of clock cycles equal to the size of the shift register is required in order to perform the fault insertion. In this case, the emulation phase consists of:

- Control phase – which is used for generating the appropriate control signals for the emulation; this control phase takes one or two clock cycles.

- Fault insertion phase – during this phase, the shift register is loaded with the generated error bits; the number of clock cycles required for this phase is equal to the size of the shift register; the DUT operation during this phase is freezed.
- Simulation phase – during this phase, the actual simulation takes place; it takes one clock cycle.

The emulation cycle is depicted in Figure 5-4.

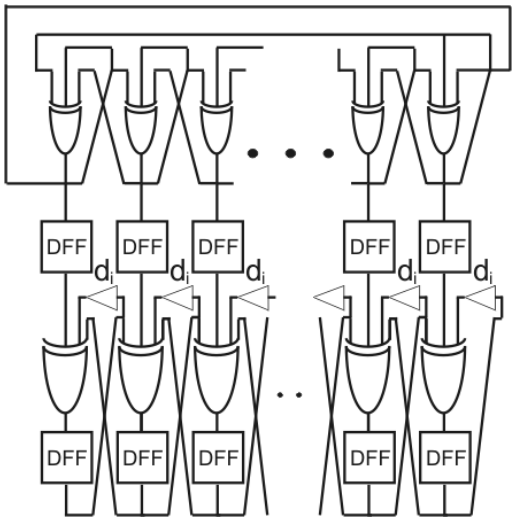


Figure 5-2 – Xilinx Based TRNG (DFF – D Flip-Flop)

Therefore, the proposed serial fault emulation scheme has the advantage of a small cost overhead and an accurate modeling of independent probabilistic errors, while it has the disadvantage of a very large simulation cycle.

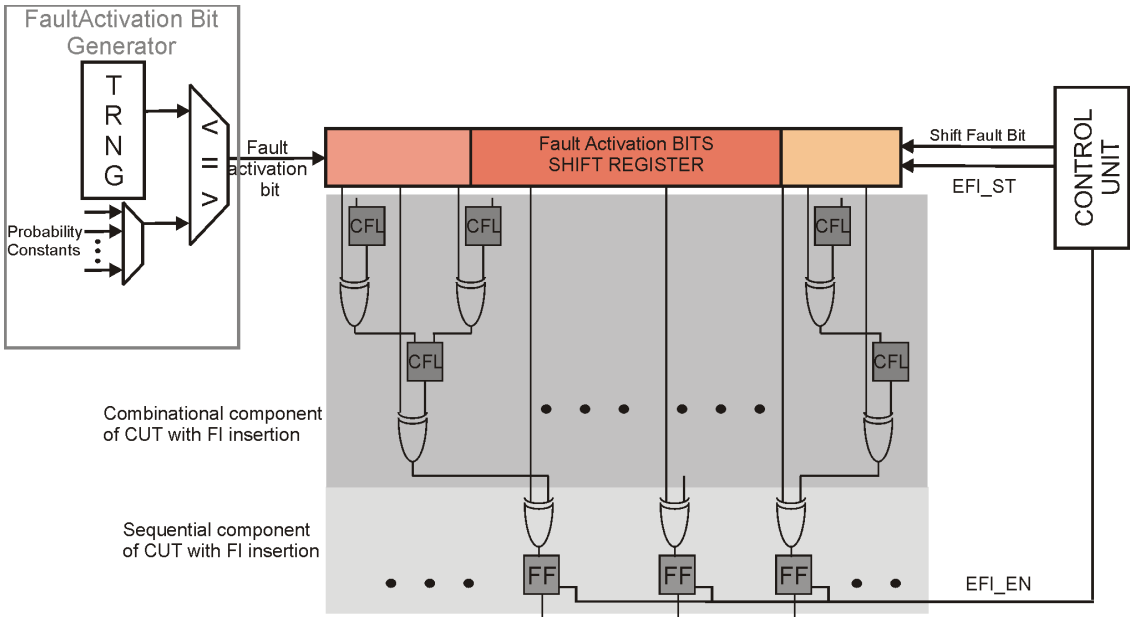


Figure 5-3 – Serial Implementation of the Proposed Fault Generation and Insertion Scheme (CFL – combinational fault location, FF – flip-flop)

The proposed serial fault generator and inserter scheme can also be used when multiple fault locations have different probabilities of error. A multiplexer which selects between different probabilities constants is used in the fault generation module, while the control becomes more complex.

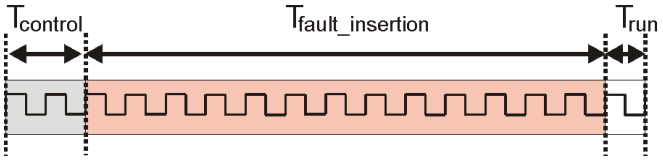


Figure 5-4 – Fault Emulation Phase Components

Regarding the fault insertion, a faulty combinational circuit has been modeled by placing a XOR gate at the output, while a faulty storing element has been modeled by placing a XOR at the input of the circuit (it stores the wrong value).

In order to reduce the number of clock cycles required for a fault emulation campaign, we propose a hybrid serial-parallel approach for error generation and error insertion. Thus, we use *k* number of TRNG-shift register modules in order to generate and insert the error bits to their corresponding fault location (Figure 5-5). The size of a single shift-register in a *k*-layer hybrid EFI scheme is *k* times less than the one required for a full serial approach. Therefore, the fault insertion phase is reduced by k times with respect to the serial EFI scheme. Thus, an improved performance can be obtained using the *k*-layer hybrid approach.

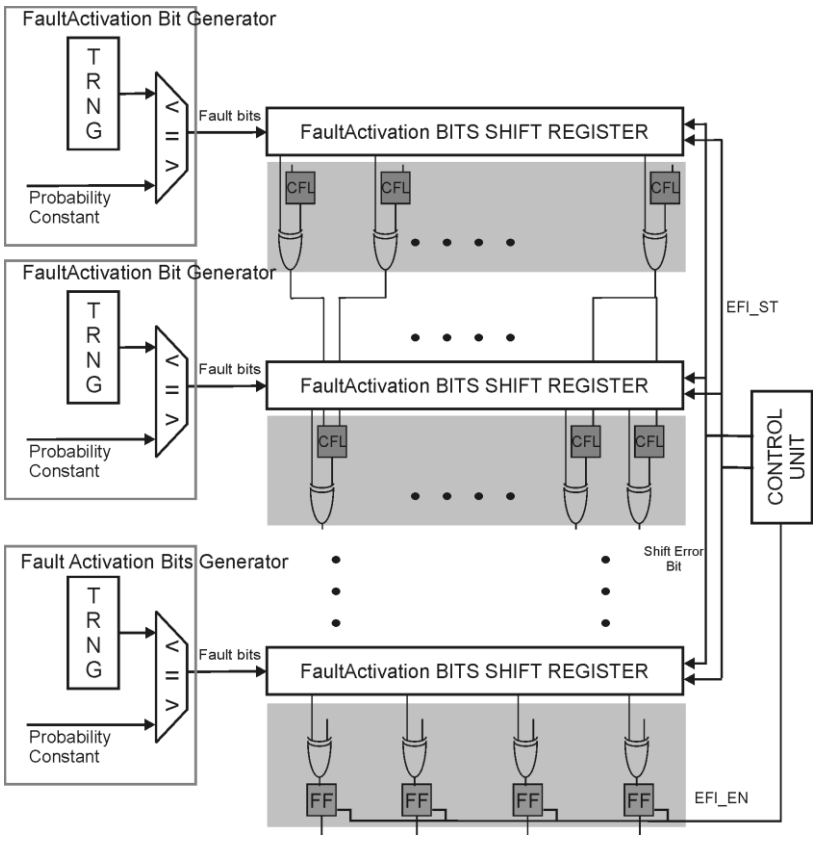


Figure 5-5 – Hybrid Serial-Parallel Implementation of the Proposed Fault Generation and Insertion Scheme (CFL – combinational fault location, FF – flip-flop)

Furthermore, the hybrid approach has simpler control when different fault locations have different probabilities. This is due to the fact that one layer generates and inserts error bits to fault locations which have the same probability.

5.4. FPGA Resource Comparison

Figure 5-6 presents the cost (in Look-Up Tables (LUT) – Flip-Flop (FF) pairs) obtained after synthesis for EFI error generation and insertion module. The estimates have been obtained using the Xilinx ISE 14.4 for Xilinx Virtex-5 VLX-50T FPGA device. The synthesis results show the linear dependence of the serial implementation, 5-layer and 10-layer hybrid implementation with the number of fault locations. Both serial and the hybrid implementations use the 32-bit TRNG. The figure also shows that for a small number of fault locations (e.g. 500), the serial implementation has significant reduced cost with respect to the hybrid (50% less than 5-layer hybrid and 75% less than 10-layer hybrid). For a small number of fault locations, the TRNGs and control logic represent an important cost component, which explain the difference. For a large number of fault locations (e.g. 10000 or 15000), the shift registers cost represent the most important component in the overall cost of the EFI Fault Generation and Control module. Therefore, in these cases, the difference between the hybrid implementations and the serial implementations becomes almost irrelevant compared to the total cost of the EFI fault emulation and insertion scheme. Furthermore, a parallel version (one RNG per fault location) which uses a 24-bit LFSR as RNG has a cost of up to 10 times more with respect to a 10 layer hybrid for 500 fault locations. Thus, we may conclude that for a large number of fault locations, hybrid implementations represent the best cost-EFI performance trade-off.

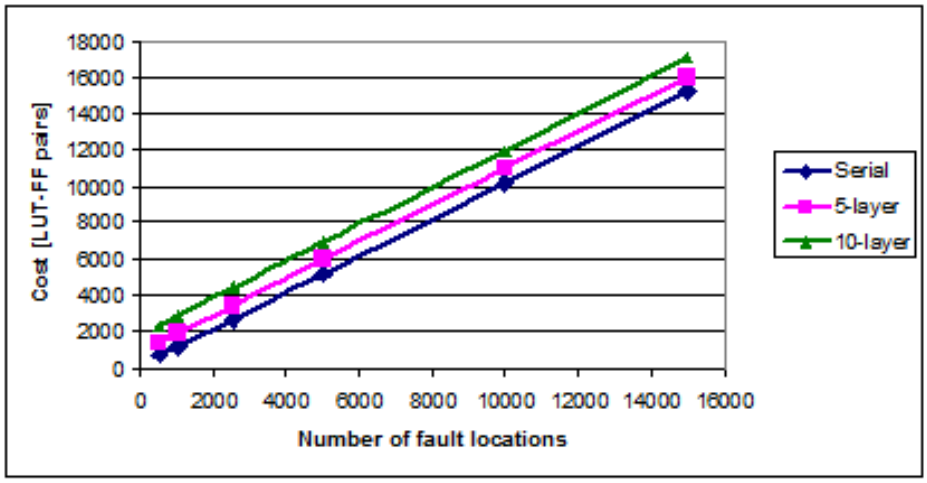


Figure 5-6 – Fault Generation and Control Cost for Various Number of Fault Location

The proposed EFI architectures have been applied for two ISCAS 85 and 89 benchmarks circuits: *c499* (combinational) and *s1196* (sequential). Table 5-1 presents the post place-and-route area estimates for the two circuits. The EFI modules for these two circuits have been implemented on a Digilent Genesys board with Xilinx Virtex-5 FPGA. The synthesis and implementation process has been performed using Xilinx ISE 14.4 software.

Table 5-1 – Cost Estimates for EFI schemes Applied for *C499* and *S1196* Benchmark Circuits

	c499 (LUT-FF pairs)	*s1196* (LUT-FF pairs)	Overhead *(c499)*	Overhead *(s1196)*
Serial	1070	1588	1750%	1185%
Hybrid	1145	2361	1877%	1761%
Parallel	9281	14057	15200%	10400%
No EFI	61	134	0	0

Regarding the *c499* benchmark circuit, a number of 5 layers for the hybrid implementation have been considered. The maximum number of fault locations for a single layer is 40. The number of fault locations for *c499* is 188. For the s1196 benchmark circuit, the hybrid implementation consists of 5 layers, with the maximum number of fault locations for a single layer of 119. The number of fault locations for *s1196* is 406.

The results show that using the full serial implementation, the overall cost overhead (which includes the EFI Error Generator and Control, the autonomous testbench and the Chipscope based observation logic) is 1700% for the *c499* and 1200% for *s1196* benchmark circuits with respect to the two circuits with no EFI. The 5-layer hybrid implementation has an increased area consumption cost of 6.5 % for the *c499* and 48 % for *s1196* with respect to the full serial. The cost of the fully parallel implemented with 24-bits LFSR is around 9 times higher with respect to the fully serial implementation.

Other probabilistic EFI approaches have been presented in **[May2012][May2013]**. Their approach considers the fault locations only the memory elements (the flip-flops); the fault locations considered in their experiments is 18 for *s1196* both in **[May2012]**.The overhead obtained in their approaches is more than 2000% with respect to the basic circuit. In our cases, both the serial and the hybrid present better cost, especially if we consider that in our EFI implementations for *s1196* the number of fault location is more than 22 times higher with respect to **[May2012]**. This difference of the area cost can be explained by the proposed EFI architectures, as well as our autonomous testbench that drives the simulation, and by the usage of ChipscopePro cores that are optimized for Xilinx technology. Furthermore, our experiments also indicate (as in other FPGA EFI implementations) that the cost of the autonomous benchmark and the observation logic represent an important component in the overall cost of the EFI circuitry.

5.5. Data Dependent FPGA Fault Emulation

We have investigated how the proposed EFI scheme can be used for data dependent fault analysis. In the data dependent reliability analysis, each output transition (for output data dependent probabilistic fault models) or each input transition (for input data dependent probabilistic fault models) has associated its error probability.

The fault injection circuitry for a fault location consisting of a 2-input circuit is depicted in Figure 5-7. For each input or output transition probability it uses a dedicated TRNG-shift register based fault generation and insertion module. For each fault location, flips-flops are used for storing

the previous value of the output/input. Based on the previous outputs/inputs and the current outputs/inputs, the appropriate fault bit is selected.

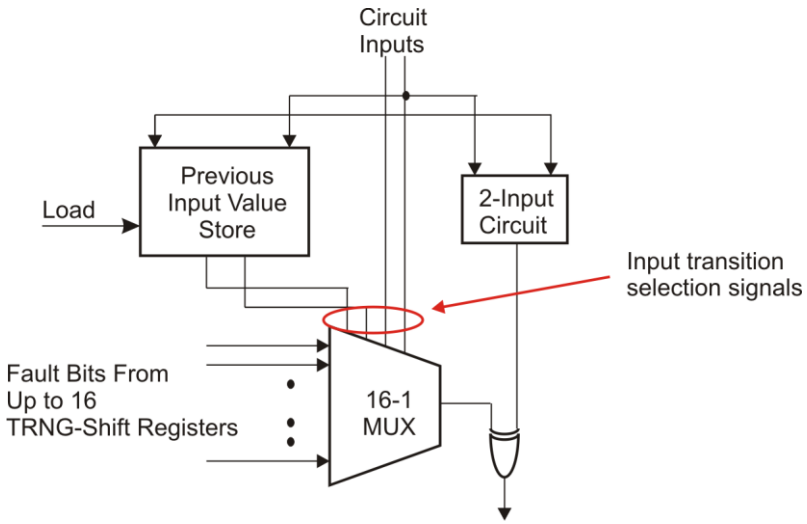


Figure 5-7 – Fault Insertion for Input Data Dependent Model for a Fault Location Consisting of a 2-Input Logic Circuit

Regarding output data dependent, it uses two TRNG-shift register modules: one for the 1-0 output switching and one for 0-1 output switching. Each fault location requires a flip-flop for storing the previous output and a 2-1 multiplexor for selecting the appropriate fault bit.

Regarding input data dependent, the number of TRNG – shift register modules is up to 2^{2n}, where the *n* represents the number of inputs. Furthermore, n flip-flops are used for storing the previous value, as well as a 2^{2n} - 1 multiplexer for selecting the appropriate fault bit. In practice the number of the TRNG-shift register modules, as well as the size of the multiplexer is reduced, due to the fact that there are input switch combinations which do not lead to an output switch.

We have implemented the output data dependent and the input data dependent fault emulation for a Taylor-Kuznetsov (TK) decoding scheme used for error correction in memories. The implemented TK scheme uses a (155,124) Low Density Parity Code (LDPC) and has been designed to perform decoding using faulty logic gates. This scheme has been proposed and analyzed in the Deliverable 4.1 **[D4.1]** of Work Package (WP) 4 **[Brkic13]**. The considered fault locations have been considered the following: the four registers containing the codewords, the 5-input XOR gates, and the 3-input majority logic gates. The 3 input majority logic gates have been implemented using 2 3-input majority logic gates. Simple fault injection (without taking care of data dependency) has also been applied at the main memory block. The main memory block has been implemented using block RAM memory. The error correction is performed in the following way:

- The codewords are stored in the main memory block.
- In case of a read operation, the codewords are buffered in the *dv* registers (where *dv* denotes the column degree in the parity check matrix of the LDPC code).
- Several iterations are performed using the combinational circuitry based on the XOR and Majority Logic; after each iteration, intermediary results are stored in the buffered registers
- After the final iteration, the codeword is available for read; furthermore, the corrected codeword is also re-written in the main memory block.

Table 5-2 – Cost Estimates for TK Decoding Schemes with Output and Input Data Dependent Fault Generation and Insertion

	Cost
TK decoding scheme (no EFI)	1928 LUT-FF pairs 10 BRAM
TK decoding scheme with output data dependent EFI	10389 LUT-FF pairs 10 BRAM
TK decoding scheme with input data dependent EFI	66435 LUT-FF pairs 10 RAM

* - Virtex-5 VLX50T device has a maximum of 28800 LUT-FF pairs

In Table 5-2 synthesis results for TK decoding scheme, EFI for output data dependent analysis for TK decoding scheme as well as EFI for input data dependent analysis are presented. The EFI scheme contains only the fault generation and fault insertion and not the Xilinx Chipscope modules for observation and control. Regarding the input data dependent, only for 32 out of the 64 possible input transitions combinations TRNG-shift registers components have been used. The other 32 possible input transitions combinations do not lead to the output switching.

Synthesis results show that the EFI scheme for input data dependent analysis introduce a very large overhead. For the considered Xilinx Virtex-5 VLX50T device, the input data dependent analysis cannot be performed for the TK decoding scheme based on the (155, 124) LDPC code, as it does not fit FPGA.

5.6. Conclusion

We have proposed a novel FPGA fault emulation scheme for probabilistic error analysis. The proposed implementations rely on a TRNG for fault bits generation and a shift register for fault bits insertion to their according fault locations. The main goals for our EFI approach have been the error correlation avoidance and low cost for the proposed approach. The main contributions of this paper are:

1. Uncorrelated (both spatial and time uncorrelation) probabilistic fault generation and distributing obtained by using TRNG, as well as by loading the entire shift register with generated error bits before a simulation cycle takes place.
2. Hybrid serial-parallel implementation by using multiple TRNGs – shift registers for the error generation and insertion; this way, good performance-cost tradeoffs can be obtained, especially for many fault locations.
3. Good observability and control for EFI by using the FPGA vendor supplied logic analyzer tools (Xilinx Chipscope).

In order to tackle the number of large clock cycles required to load the shift register in the serial implementation, we have developed the hybrid serial-parallel approach. This way, we target better performance a *k*-layer hybrid implementation reduces the number of clock cycles for loading the shift

registers up to *k* times. Regarding the cost of the proposed approach, we have “inserted” a number of 406 fault locations with respect to the 18 fault locations in **[May2012]** for the same area overhead.

Furthermore, we have developed data dependent EFI schemes. These involve the usage of more TRNG – shift register modules, depending on the number of considered outputs/inputs transition probabilities. Regarding the output data dependent fault analysis, the EFI scheme uses 2 TRNG – shift registers components. Regarding the input data dependent model, the EFI scheme may use up to 2^{2n} TRNG – shift register components. As the synthesis results for input data dependent EFI for TK decoding scheme implemented on Virtex-5 VLX50T device, the cost is prohibitive (the fault emulation scheme did not fit the device).

Future work will consist in 2 directions. On one hand, we will target the reduction of the emulation time. This will be achieved by shuffling the bits within the fault insertion component. It is predicted that applying shuffling, correlation between errors will be introduced. On the other hand, we will investigate the cost reduction for data dependent EFI. Furthermore, the proposed methodology will be used for validating designs developed in WP6.

6. Gate-Level/Register Transfer Level Fault Modelling for Probabilistic Sub-Powered Interconnects

Abstract: In this chapter, we introduce Gate-Level (GL)/Register-Transfer-Level (RTL) data dependent probabilistic error models for interconnects. We propose four types of probabilistic fault models: simple probabilistic fault model, switching type probabilistic fault model, the Full Data Dependent (FDD) fault model and the Partial Data Dependent (PDD) fault model. Regarding the FDD fault model, it is based on the fact that the probability of correct switching for one wire within the bus is influenced, through capacitive and inductive coupling, by all the wires within the bus. The PDD fault model is based on the fact the probability of correct switching for a wire is influenced only by the neighbor wires, as the effect of the inductive coupling is considered negligible. We have applied the proposed fault models for the analysis of the open source Wishbone bus.

Publications: Part of this work has been published in:

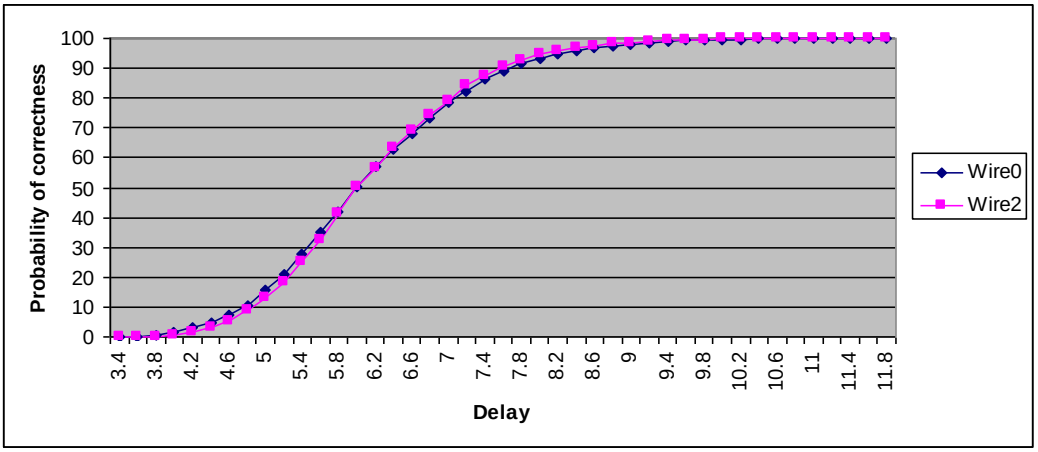
S. Nimara, A. Amaricai, O. Boncalo, M.Popa "Probabilistic saboteur-based simulated fault injection techniques for low supply voltage interconnects" Proc. 10th International Conference on PhD Research in Microelectronics (PRIME), Grenoble, 2014

6.1. Reliability Issues in Interconnects

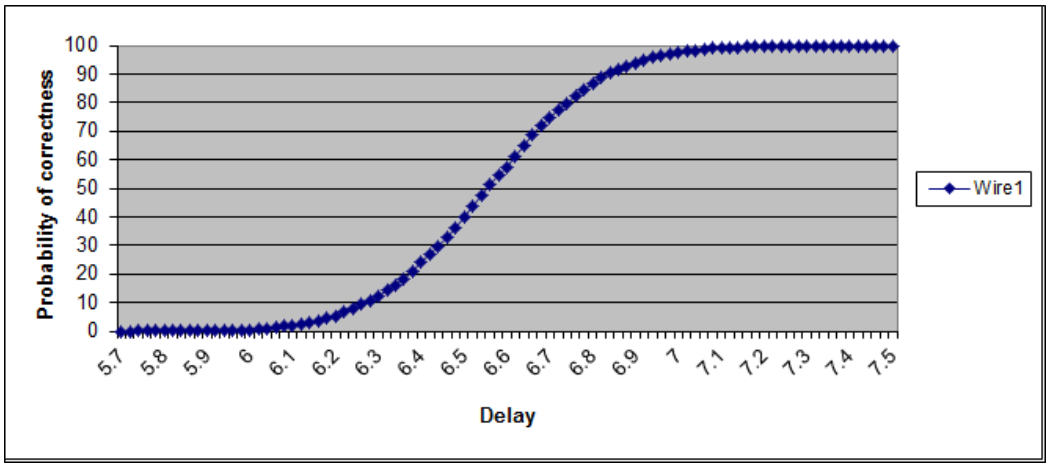
The main factors that lead to reliability issues in interconnects are process variation and crosstalk induced faults. Regarding the process variations, the most frequent forms of it are represented by: device geometry variations, device material and electrical parameter variations, interconnect geometry and material parameter variations **[Agarwal04][Boning99][Nagaraj06]**. These variations will have an effect on the metal thickness or length, dielectric thickness, contact and via size, metal resistivity or dielectric constant. Thus, the resistance, capacitance or inductance parameters of a wire are affected. Process variation in interconnects may alter the timing characteristics of the signals. Thus, an erroneous result at the moment when a certain signal is sampled may appear due to increased resistance or ground capacitance of the wire **[Agarwal04]**.

Crosstalk faults are most probably the result of inappropriate interconnect routing scheme, rather than manufacturing defects **[Sanyal09]**, and they are strongly data-dependent. For the interconnection lines, cross-talk induced faults result from an undesired inductive or capacitive coupling between two or more signal lines, producing both timing alterations and / or noise (like glitches) on those signals **[Favalli04]**. These parasitic couplings determine an energy transfer from one wire to another, depending on the driver strength and they result in crosstalk faults **[Hasan10]**. The authors in **[Hasan10]** realize a classification of crosstalk faults into crosstalk induced glitches and delays. Crosstalk induced glitches appear on a static victim (affected) line when one or more aggressor lines switch their logic value, while crosstalk induced delays occur when aggressor and victim signals change their logic state simultaneously **[Agarwal04]**. The most dominant effect is represented by the capacitive crosstalk: this affect only the neighboring line **[Sanyal09]**. The inductive crosstalk has a smaller influence with respect to the capacitive one; however, the inductive effects may span across multiple lines.

In work covering first year of the project **[D2.1]**, we have performed SPICE Monte-Carlo simulations for buses consisting of 3 wires. We have considered a Resistance, Capacitance, Inductive (RLC) model for interconnects. Process variations have been reflected in the variation of the RLC parameters of the wires. The simulation results show a very strong data dependency in the case of interconnects. The variation in switching delay between different input switch combination sis far greater with respect to logic gates. As Figure 6-1 indicates, the delay for 000-001 transition is three times lower with respect to the 111-010 transition.



a)



b)

Figure 6-1– Correct Switching Probability for 3 Wire Interconnects for Vdd = 0.25V(a - 111 – 010 switching b – 000-001 switching)

Therefore, interconnects pose more reliability problems in sub and near threshold regions of operations, mainly due to the effect of the capacitive crosstalk.

6.2. Probabilistic GL/RTL Fault Models for Interconnects and Their Simulation Methodology

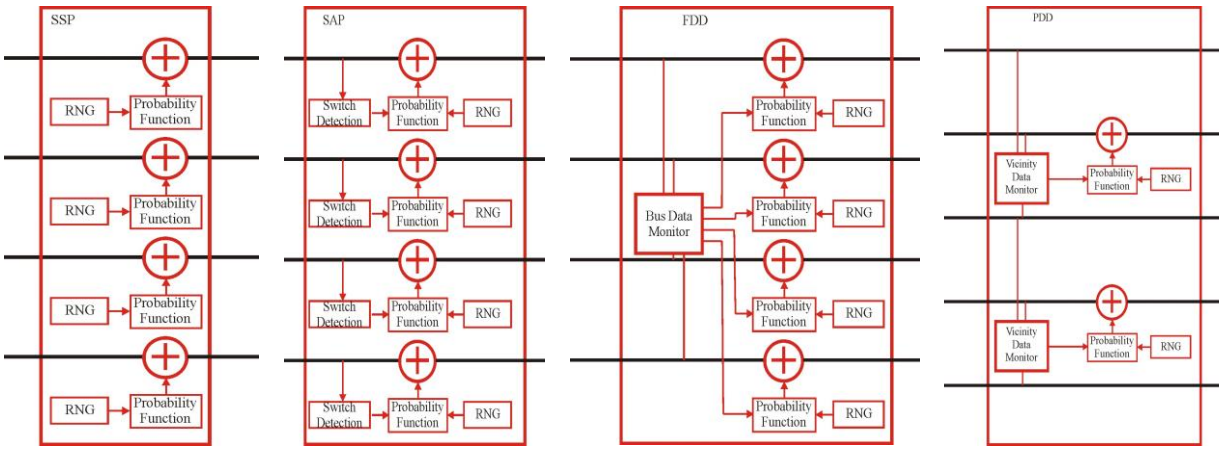
Based on the simulation results presented in the previous section, we have developed four types of fault models for probabilistic interconnects:

1. Standard Signal Probabilistic (SSP) fault model. It represents the simplest one because it only flips the logic value of a certain signal with a given bit-independent probability. It doesn't take into account the last type of transition that took place on that line, nor the data pattern

2. Switching-Aware Probabilistic (SAP) fault model. This model considers probabilistic behavior for a signal only when switching is taking place. It models accurately timing faults: the switching for a line does not respect a given timing constraint. The most simplistic type of SAP considers the same probability for both types of switching; a more accurate considers different probabilities for charging and discharging processes.

3. Full Data Dependent (FDD) fault model. For this model, the probabilities for a line are dependent on the data configuration on the entire bus. This represents the most accurate model, as the timing and value characteristics for a wire are affected by crosstalk (which is data dependent). Although this model is the most accurate, it has very poor scalability: for an *n*-bit bus, 2^n probabilities for a single line are derived (the crosstalk noise manifests when the bus switches).

4. Partial Data Dependent (PDD) fault model. This model represents a simplification of the previous one. The probabilities for a line are dependent on the data configuration on vicinity (1-wire vicinity or 2-wire vicinity). The 1-wire vicinity model is based on the fact that the capacitance effect (which is dominant) manifests only on the neighbor line. With respect to the FDD, for a single line 35 wire switch probabilities have to be used.



a) b) c) d)

Figure 6-2 - Saboteurs' Architectures Corresponding to Proposed Fault Models (a – SSP, b – SAP, c – FDD, d – PDD)

Regarding the simulated fault injection for gate level and RTL description of interconnects, the saboteur represent the natural candidate for implementing the four fault models described above. The proposed SFI methodology has been implemented in Verilog; however, it can be easily adapted to VHDL. Several types of saboteurs have been proposed, such as **[Baraza05][Jenn94]**: serial simple unidirectional saboteur, serial simple bidirectional, serial complex saboteur, serial complex bidirectional saboteur, n-bit unidirectional serial saboteur, n-bit bidirectional serial saboteur, parallel saboteur. According to this classification, the proposed saboteurs can be considered n-bit unidirectional serial saboteurs.

The SSP model-based saboteur contains a fault insertion module, which is triggered according to the desired probability of failure and to the output provided by a random number generator. The

architectures for data dependent saboteurs contains bus data monitors, which extracts the switching activity on the interconnect (according to the corresponding fault models).

Figure 6-2 presents the architectures for four types of saboteurs. All saboteurs consist of a random number generator (which is used to compute the probability of an error). The SAP incorporates a switch detection module, while the PDD and FDD monitor the data on the lines.

6.3. Case Study: Wishbone Bus Analysis

We have performed several simulation campaigns, each of them consisting of 1000 runs and data transmitted was chosen randomly for each run. The simulations have been carried out using Modelsim 10.3 commercial HDL simulator on desktop computer with Intel Core 2 Duo at 2.4 GHz and 2 GB of main memory, with Windows XP OS.

The circuit under test has been the open-source Wishbone bus, designed in Verilog HDL and available on the OpenCores site **[Wishbone10]**. The system was simulated in the particular case of 2 master units and 5 slave units, with 32-bit data and address buses. We have simulated conventional read and write cycles. The sabotaged signals have been grouped into the following:

- Data write signals (the 32-bit unidirectional data bus from master to slave)
- Data read signals (the 32-bit unidirectional data bus from slave to master)
- Address signals – a distinction between the first 3 address bits (the ones used to select the slave) and the rest of the address bits (which are used to address within the slave)
- Master control and handshaking signals (*we, cyc, stb* and *sel)*
- Slave handshaking signals (*ack, rty, err*)

The analyzed system is depicted in Figure 6-3.

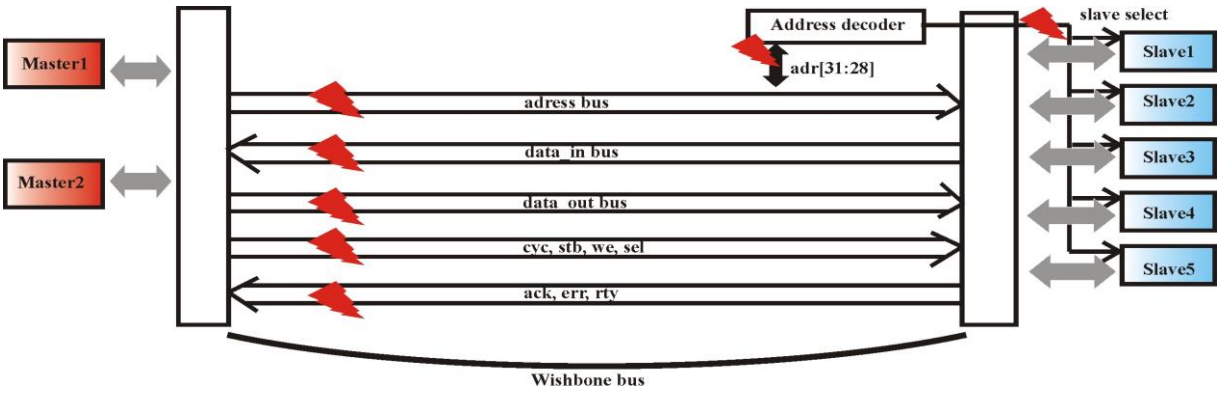


Figure 6-3- Fault Injected Wishbone Bus Signal Groups

The simulation campaigns and simulation times are presented in Table 6-1. Regarding the simulation times, a simulation set consisting of 1000 executions requires less than 2 s. The correct circuit simulation requires almost 1 s.

Regarding the reliability analysis of the Wishbone bus, the following conclusions can be drawn:

1. Faults affecting the most significant signals of the address line have a dramatic effect on the overall signal reliability, as these signals are used for slave selection. Therefore, an error on these signals will result in selecting a wrong slave.

2. Faults affecting master to slave control and handshaking signals (*cyc, stb* and *we*) have the following effects: wrong type of transaction (read instead of write or vice-versa), no transaction is performed (because the bus arbiter cannot grant the bus to the master which had asserted the *cyc* signal or the slave to take into consideration the request from a master), prematurely terminated transactions (due to errors on an ongoing transaction on *cyc* and *stb* signals – these signals are activated throughout an entire transaction);

Table 6-1 – Simulation Results for Saboteur Based SFI of Wishbone Bus

Fault model type	Victim signal	Probability of failure	Runtime [ms]	Fault model type	Victim signal	Probability of failure	Runtime [ms]
SSP during WRITE cycle	Sel	3%	1828	SAP during WRITE cycle	adr[31:28]	5% for 0->1 3% for 1->0	1766
	sel and data	3%	1765		adr[31:28]	10% for 0->1 5% for 1->0	1766
	adr[31:28]	3%	1812		cyc, stb, we, sel	5% for 0->1 3% for 1->0	1750
	adr[31:28]	5%	1750		cyc, stb, we, sel	10% for 0->1 5% for 1->0	1781
	adr[31:28]	10%	1750		data	5% for 0->1 3% for 1->0	1766
	cyc, stb, we, sel	3%	1750		data	10% for 0->1 5% for 1->0	1782
SSP during READ cycle	ack, err, rty	3%	1703	SAP during READ cycle	ack, err, rty	5% for 0->1 3% for 1->0	1703
					ack, err, rty	10% for 0->1 5% for 1->0	1703
PDD during WRITE cycle (1-wire vicinity)	adr[31:28]	[3% ÷ 20%], depending on the transition pattern	1797	PDD during READ cycle (1-wire vicinity)	ack, err, rty	[3% ÷ 20%], depending on the transition pattern	1782
	adr[27:0]		1797		data		1906
	slave select signals		1766				
	cyc, stb, we, sel		1766				
	ack, err, rty		1765				
	Data		1782				
Gold circuit – WRITE cycle	NO fault injection	0%	1078	Gold circuit – READ cycle	NO fault injection	0%	1046

3. Faults affecting the slave to master handshaking have the following effects: the bus may enter into a stand-still, as the master does not de-asserts the *cyc* signals because he has not received any *ack, rty* or *err*; a transaction may be terminated before, as the master receives a wrong *ack, err*, or *rty* – in case an error affects *ack*, the master may read the wrong data; longer transaction when errors appear on the *rty* signal (usually a master restarts the transaction for a *rty*).

4. Faults that affect *sel* lines and data signals affect only the data transmitted on the bus. They do not affect the transaction timing or flow.

Thus, regarding the reliability of the bus, the most critical signals are the most significant bits in the address line and the control and handshaking signals.

6.4. Conclusion

Interconnect face many reliability problems in the context of sub and near threshold computing, due to process variations and crosstalk effects. We have proposed data dependent saboteurs based fault injection for bus reliability analysis, which can be performed at both gate level and RTL. We have developed four types of fault models, of which two are specific to interconnects: the full data dependent model and the partial data dependent model. The full data dependent model is the most accurate one and considers the effect of both capacitive and inductive crosstalk on the reliability of the wires. The partial data dependent fault model has improved scalability, as it considers the probability of correct switching as dependent only on the neighbor wires. The PDD is based on the fact that inductive crosstalk has negligible effects.

7. Energy Models

Abstract: In the era of deep submicron CMOS technology, variability such as On-Chip Variability(OCV) and Process Temperature Voltage(PTV) variability, cause less predictable device and system behavior in terms of energy analysis. Modeling accurately these processes leads to significant simulation time (due to Monte-Carlo (MC) types of simulation); furthermore, there is no unified framework for such models for large circuits. This issue is further deteriorated in near/sub threshold region when low power consumption is required. To tackle the issues related to building an accurate, fast energy model, we pursue a bottom-up approach. We build models related to basic logic gates including AND, Inverter (INV) - used in the various data structures for reliable circuit synthesis in Work Package (WP) 5 -, as well as NAND. Using NAND gates we build 3 input XOR and 3 input majority logic (MAJ) circuits -used in the design of low complexity decoders of WP4. Ongoing work is to expand these energy models to build energy models for the constituent blocks of LDPC decoders as well as for larger circuits using AND-INV gates(AIG) synthetized for improved reliability. Furthermore, we have tackled the issue of energy modeling and estimation of LDPC decoders and some results with real energy measurements are reported.

Publications: Part of this work has been published in :

T. Marconi, C. Spagnol, E. Popovici, S. Cotofana, "Towards Energy Effective LDPC Decoding by Exploiting Channel Noise Variability", Proc. 22nd IFIP/IEEE International Conference on Very Large Scale integration, 2014.

7.1. Design Flow and Energy Model Framework

In order to establish the framework for energy modeling, firstly let's consider the design flow associated with Figure 7-1 developed as part of WP5.

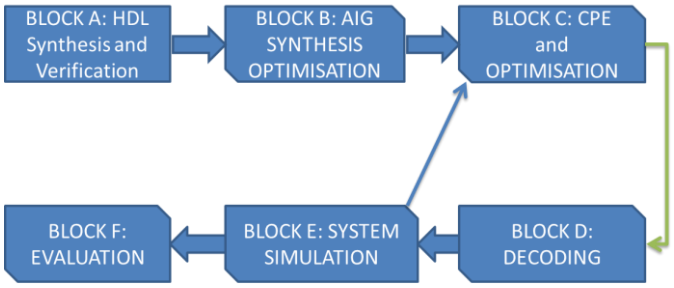


Figure 7-1– Design Flow for Reliable Synthesis

We start by describing a logic circuit by using Hardware Description Languages (HDL) synthesis and simulation - Block A. This circuit is transferred into a selected data structure, such as And-Inverter Graphs (AIG), NAND graphs or MAJ graphs, etc. - Block B. Codeword Prediction Encoding (CPE) and reliability driven optimization of the circuit is performed in Block C, resulting in Shannon annotated Boolean logic. Block D is performing the decoding of the outputs of the CPE logic. Block E analyses the outputs of the decoder and informs Block C for re-optimization (e.g. using different code parameters, etc). Finally, Block F reports on the various metrics associated with the circuits, including

energy, reliability, power, area, delay. Each of the blocks associated with the flow is built hierarchically. Energy and reliability are of prime concern within this flow. From an energy perspective, one needs to evaluate energy in all blocks associated to such a flow. Within Block A, the test/verification vectors have to be decided and these will have a direct relation to energy models. The energy model at this level however does not take into account variability (OCV, PTV, etc) and the resulting energy model might not be accurate. These effects can be taken into account as part of custom and semi-custom tools developed in Block B. Within this block, the circuit is seen as a Boolean network using different gate level representations (such as AIG, NAND graphs, etc) on which the energy models associated to each gate can be superimposed. Using CPE approach will lead to additional circuitry to be added and a new energy model has to be generated to reflect the in-circuit encoding. Regarding the decoder, there are a number of situations to be considered including symmetric decoding - the decoder is on same chip and prone to the same type of faults as the CPE circuit - or asymmetric decoding. These scenarios lead each to different energy models associated with the decoder. The energy models associated with the decoders also depend on the decoder architecture, number of iterations used to achieve certain reliability, code parameters, etc. Such models are extracted within Block E. The synthesis and optimizations results are reported in Block F where metrics such as area, delay, energy, power consumption, reliability are collected and correlated.

7.2. Gate Level Energy Models

The basic gates (AND, INV, NOT, NAND, XOR, MAJ) appear in the synthesis and optimizations tasks presented in the design flow. An extensive modelling and simulation exercise in HSPICE has been performed to get insights and simplified models for energy, delay and reliability associated with these gates. Based on certain voltage level and certain set of variability parameters, we have derived models for delay, energy and reliability associated with these gates.

As two generic building blocks in CMOS circuits, INV and NAND are considered firstly with both charging and discharging events (at the output) taken into account. The parameter variations in all simulations in this section are the same to the ones presented in Chapter 2. Figure 7-2 and Figure 7-3 exhibit the Probability Density Function (PDF) histograms of energy consumption of INV and NAND respectively, as well as the Inverse Gaussian Distribution (IGD) fittings. It is clear that IGD model fits the simulation data quite well although the theoretical explanation is still not well understood.

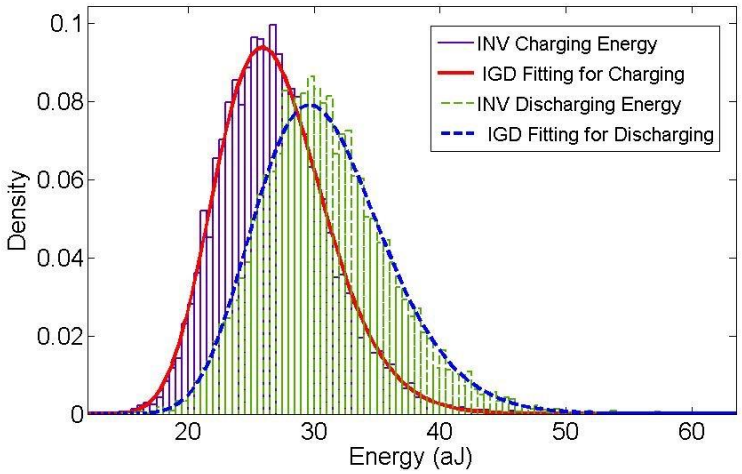


Figure 7-2– IGD Fittings for Energy Consumption of INV

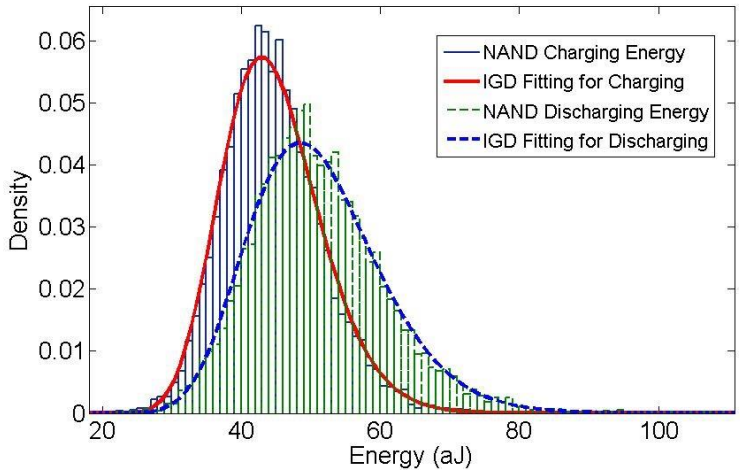


Figure 7-3– IGD Fittings for Energy Consumption of NAND

Other than these two basic gates, MAJ and 3 bit parity check(3 bit input XOR) circuits based on NAND have also been explored. Close match between the simulation data and the IGD fittings have also been found as shown in Figure 7-4 and Figure 7-5. However, each of them having three inputs complicates the study as for different input patterns result in various energy consumption. The two figures indicate that unlike the INV and NAND histograms, the histograms for 3bit-MAJ and 3bit-XOR gates are separated considerably. It is caused by different input patterns, which may trigger different number of gates during the evaluation. Another reason is represented by glitches which make the energy estimation based on simple composition of the basic gates less feasible. Case 1 represents charging, and case 2 represents discharging.

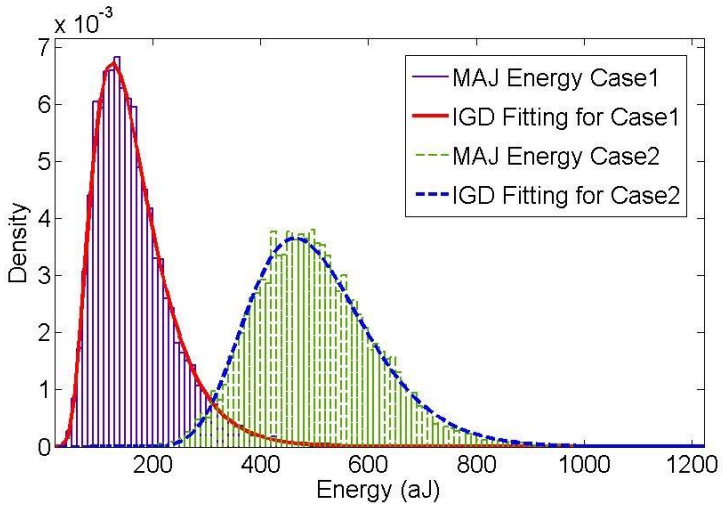


Figure 7-4– IGD Fittings for Energy Consumption of MAJ

(Case1 – Charging; Case2 – Discharging)

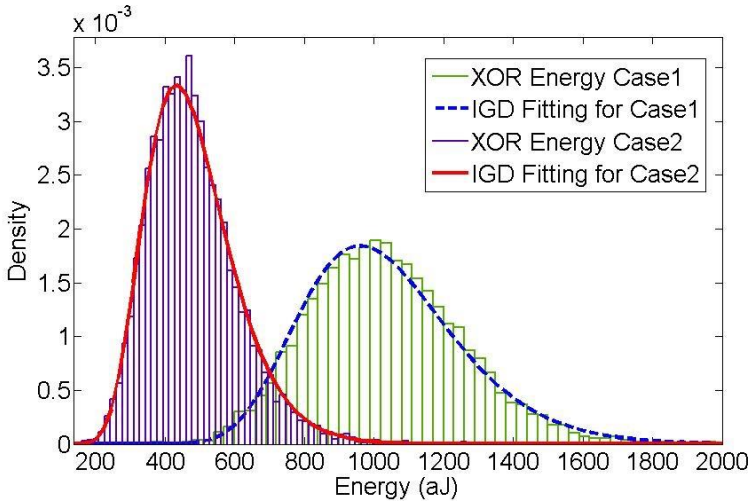


Figure 7-5– IGD Fittings for Energy Consumption of XOR

(Case1 – Charging; Case2 – Discharging)

In the future, we will continue look into this aspect in order to find a model capable of energy estimation/prediction as well as linking it to reliability of the circuits.

7.3. Component Blocks Energy Models

The gates(AND, INV, NOT, NAND, XOR, MAJ) are also building blocks for some of the main component parts of various LDPC encoding and decoding circuits developed in iRISC. Understanding reliability, energy models associated with these computational blocks, would allow an efficient, high level evaluation of energy. The bottleneck is in evaluating energy for large circuits where near/sub threshold voltages are used, and where variability is present due to computational requirements. Activities are ongoing to simplify our analysis while keeping a high degree of accuracy; and linking the developed reliability models with energy.

7.4. Decoder Energy Models

A number of decoders were implemented to date and are currently also evaluated from an energy perspective. The implemented decoders include: probabilistic gradient descent bit-flipping decoders **[Rasheed14, Le14]**, Log Likelihood Ratio Belief Propagation algorithms **[Marconi14],** layered LDPC decoder **[Boncalo14]**, finite alphabet iterative decoders robust to faulty hardware **[Dupraz14]**. At this level, energy depends on the code parameters, architecture of the circuits, number of iterations, voltage levels, clock speed, etc. Developing energy models for these decoders is an ongoing activity. However, a thorough energy evaluation accompanied by energy measurements on FPGA was implemented in **[Marconi14]**.

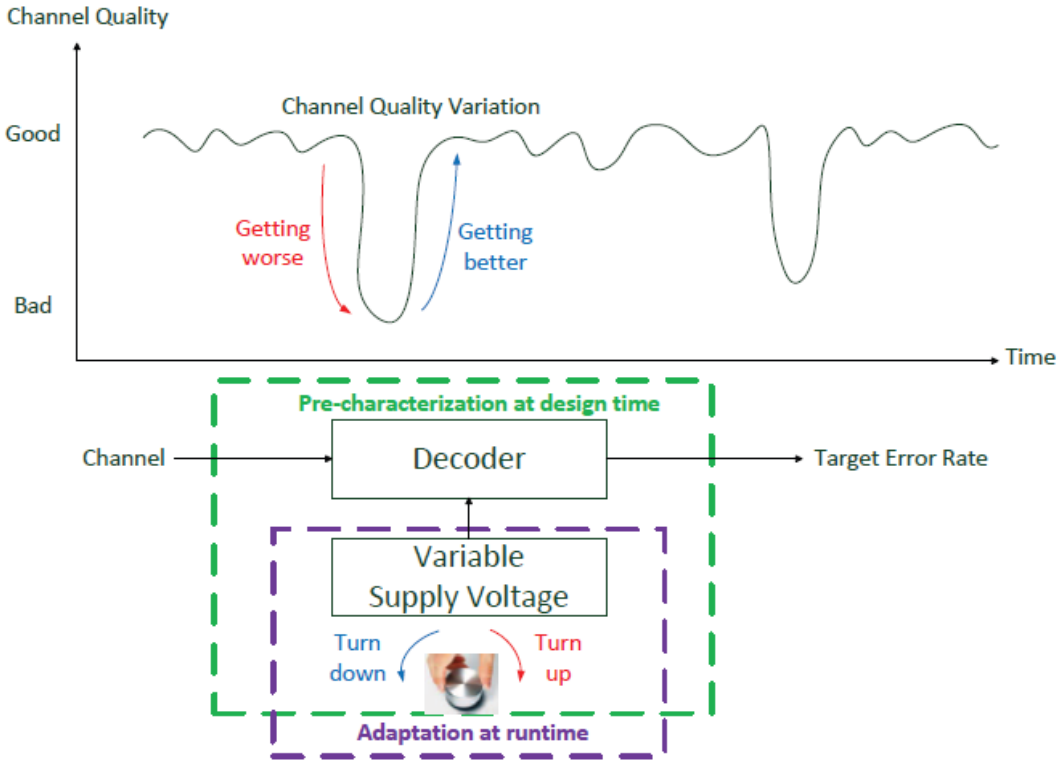


Figure 7-6– Channel Aware Energy Effective Decoding

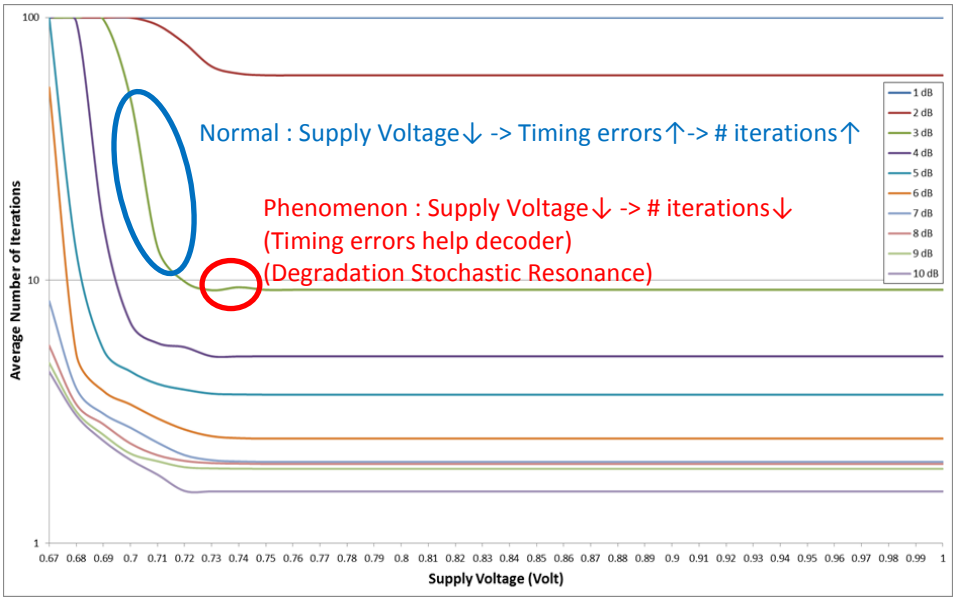
In classical communication systems, channel quality variation is a well-known phenomenon, which fundamentally influences the decoding process. While most of the time, the transmission takes place in good signal to noise conditions, to satisfy Quality of Service (QoS) requirements in all cases, telecom platforms rely on largely overdesigned hardware, which may result in energy waste during most of their operation. We proposed to exploit the channel noise variability and adapt the platform operation conditions such that QoS requirements are satisfied with the minimum energy consumption. In particular, we propose a technique to exploit channel noise variability towards energy effective LDPC decoding amenable to low-energy operation. Endowed with the channel noise variability knowledge, our technique adaptively tunes the operating voltage at runtime, aiming to

achieve the optimal tradeoff between decoder performance and power consumption, while fulfilling the QoS requirements. To demonstrate the capabilities of our proposal, we have implemented it and other state of the art energy reduction methods in conjunction with a fully parallel LDPC decoder on a Virtex-6 FPGA. Our experiments indicate that the proposed technique outperforms state of the art counterparts, in terms of energy reduction, with 71% to 76% and 15% to 28%, while maintaining the targeted decoding robustness. Moreover, the measurements suggest that in certain conditions Degradation Stochastic Resonance occurs, i.e., the energy consumption is unexpectedly diminished due to the fact that unpredictable underpowered components facilitate rather than impede the decoding process.

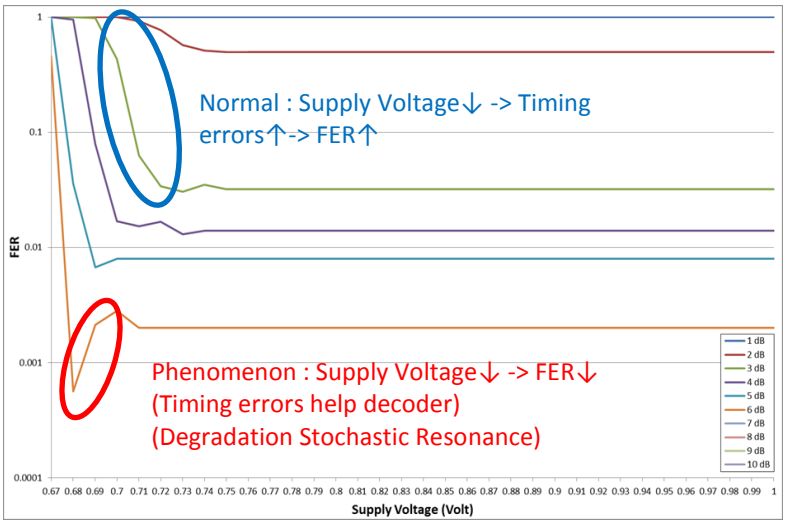
The main concept behind the proposed technique is presented in Figure 7-6. Maintaining the decoding performance (in terms of Frame Error Rater (FER)/Bit Error Rate (BER)) to its required value by actively monitoring the channel noise represents the main way to prevent energy over-consumption. More precisely, we turn the supply voltage up when the channel is getting worse for meeting the target error rate, and vice versa, we turn it down when the channel is in a good condition to save energy.

The question is “How far can we turn the supply voltage down in good channel conditions or up in bad channel conditions?” If we increase the voltage too much when the channel is getting worse, we may spill energy, while if the voltage is not high enough, the target error rate cannot be satisfied. Similar situations may also occur when turning down the voltage in good channel conditions. To determine the appropriate decoder operating voltage at a specific channel condition, we need to know the decoder behavior by means of a pre-characterization process.

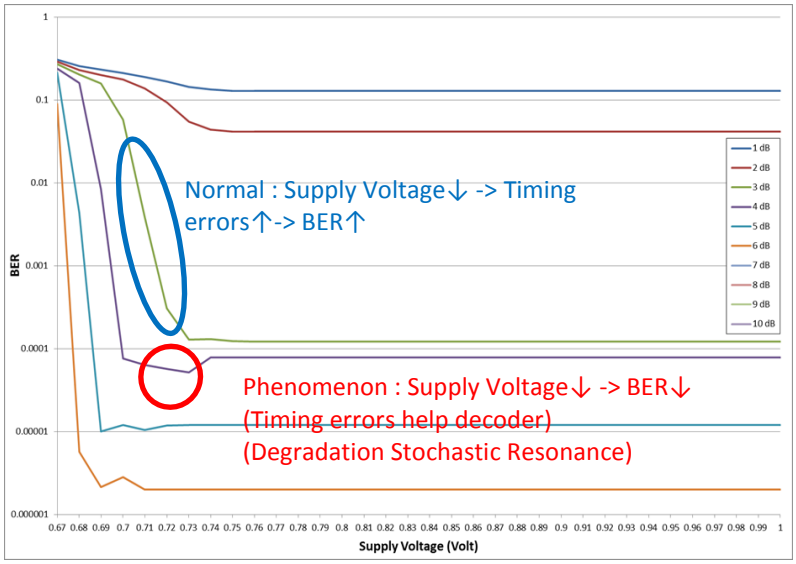
The pre-characterization results are then used to compute the decoding operating voltage for any specific channel condition (Figure 7-7). These values are stored in an LUT and utilized in guiding the decoder to meet the required target error rate while consuming as low energy as possible in the adaptation process.



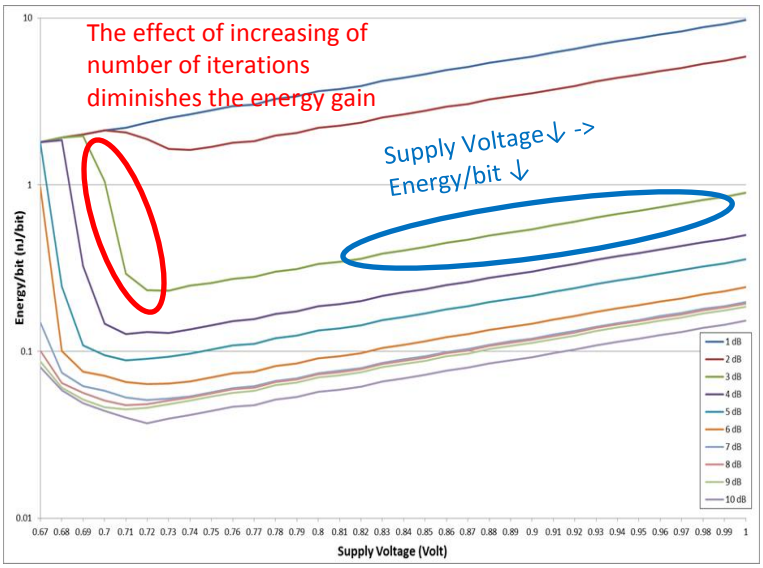
(a) Number of Iterations



(b) Frame Error Rate (FER)



(c) Bit Error Rate (BER)

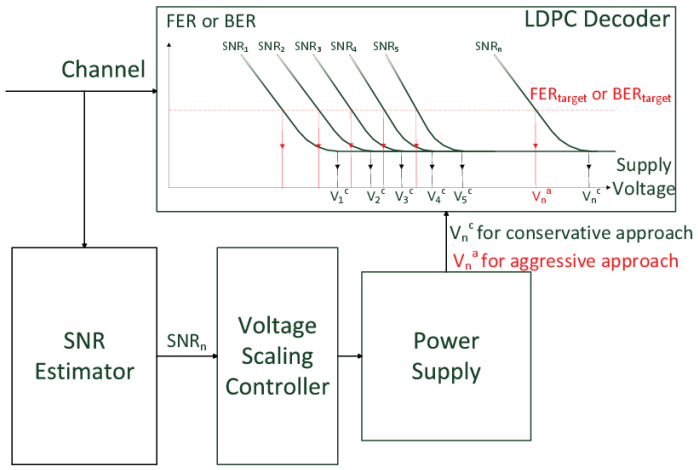


(d) Energy/bit (nJ/bit)

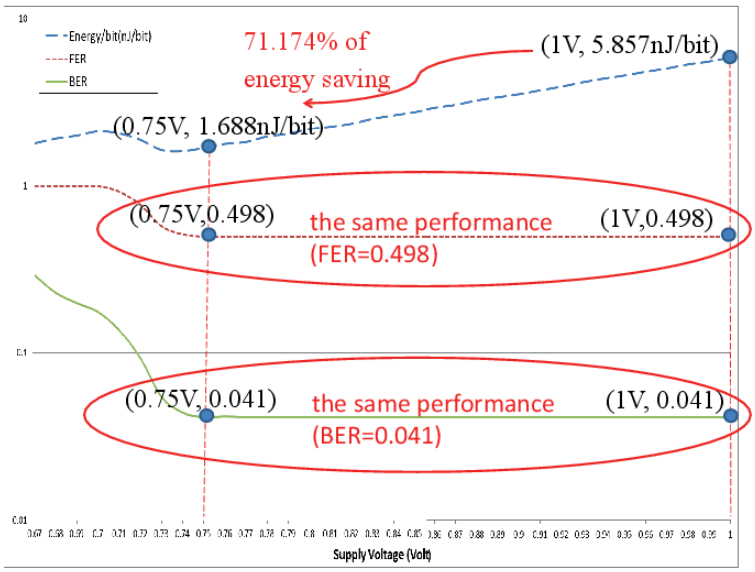
Figure 7-7– Pre-characterization Results

For implementation we target Xilinx Virtex-6 FPGA XC6VLX240T-1FFG1156 and use Xilinx ISE tools. Based on Post Place and Route Static Timing Report of Xilinx tools, the minimum clock period is 19.992 ns (i.e., the maximum frequency is 50.020 MHz). The actual implementation is clocked at 50 MHz. Therefore, the throughput at its maximum iterations is 250 Mbps for all experiments. The pre-characterization results when varying the power supply value from 1V to 0.67V and the channel Signal to Noise Ratio (SNR) from 10dB to 1dB are presented in Figure 7-7 as follows: (a) Average number of iterations, (b) FER, (c) BER, and (d) Energy/bit (nJ/bit). Each SNR has its own minimum supply voltage after which the number of iterations starts to increase sharply as one can observe in Figure 7-7 (a). In general, the increase starts earlier for lower SNR channels and this behavior can be related to the fact that the decoder can do self-correction easier for higher SNR channels where there is not much noise involved. Each SNR has its own specific minimum supply voltage after which its number of iterations goes to the maximum number of iterations which is 100. For the majority of the results it can be seen that when the supply voltage is lowered, the number of iterations stays constant for a while and then increases for the decoder to tackle timing errors. It is unexpected but interesting to note that sometimes, the average number of iterations decreases even if the supply voltage is reduced, which suggests that sometimes the timing errors can help the decoder converging to the correct codeword. This phenomenon is called Degradation Stochastic Resonance **[Aymerich12**] or Stochastic Resonance **[Gammaitoni98]** and it can be also observed in Figure 7-7 (b) and (c) where we present the measured results for FER and BER, respectively. This suggests that voltage reduction can sometimes help improving the decoder performance. Finally, in Figure 7-7 (d), the measured energy/bit for various SNRs is presented. The energy/bit decreases by scaling the voltage, however, after a certain point, an increase in energy/bit is visible as the effect of increasing of number of iterations diminishes the energy gain we get from reducing the voltage, the law of diminishing returns.

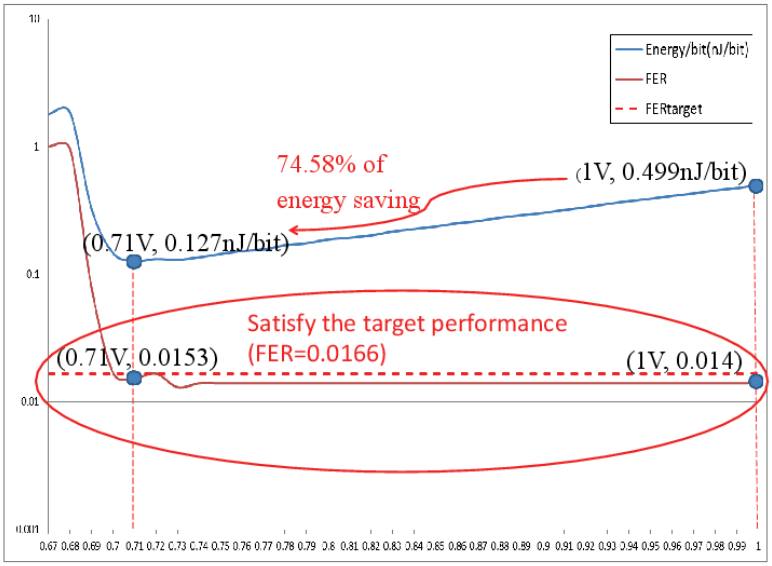
The second part of the proposed process is represented by adaptation. The voltage scaling controller for the targeted LDPC decoder depicted in Figure 7-8 operates as follows. It gets SNR information from the SNR estimator and changes the operating supply voltage at runtime based on the knowledge it has from the measured information gathered during the pre-characterization stage. We note that given that in communication systems with adaptive coding and modulation, the SNR estimator is a standard system component. The basic principle of the adaptation is to trade off over-needed performance for energy saving through active channel quality monitoring. More precisely, we turn down the supply voltage when the channel is in good condition, hence allowing energy saving. However, for preserving target performance, it is required to turn the voltage up when the channel SNR is getting worse. The objective is to minimize the energy not the voltage while ensuring the decoder achieves its needed performance. Note that minimizing the voltage may not always improve the energy efficiency, because the number of iterations of the decoder may increase due to induced timing errors. Thus, this diminishing returns effect needs to be considered when choosing the operating voltage.



a) Design



b) Conservative Approach



c) Aggressive Approach

Figure 7-8– Adaptation Step

We developed two different adaptation strategies as follows:

a. The conservative approach which guaranties that the original decoder performance is always preserved
b. The aggressive approach which only concentrates on achieving the required target performance.

To maintain identical performance while ensuring higher energy efficiency for the channel condition characterized by $SNRn$, the operating voltage Vn is moved towards the point where the energy/bit is minimized and at the same time the FER and BER remain identical to those of the decoder operated at the typical voltage $VTypical$.

By its conservative nature this approach may still sometimes result in energy waste as it tries to mimic the worst case designed decoder and not to just fulfill the target performance requirements. In view of this the aggressive strategy is designed to enable the decoder to adapt itself such that it delivers the required correction capability while minimizing the energy consumption.

To evaluate our technique, we utilize the platform and LDPC decoder as in the pre-characterization stage augmented with the following energy reduction schemes: (i) powering off capability using Early Termination (ET) technique operated at the original supply voltage **[Darabiha08]**, (ii) a Hybrid Early Termination Scheme (HS) which includes the DVS technique in **[Chow05]**, (iii) our Conservative Approach (CON), (iv) our Aggressive Approach targeting FER (AGF), and (v) our Aggressive Approach targeting BER (AGB).

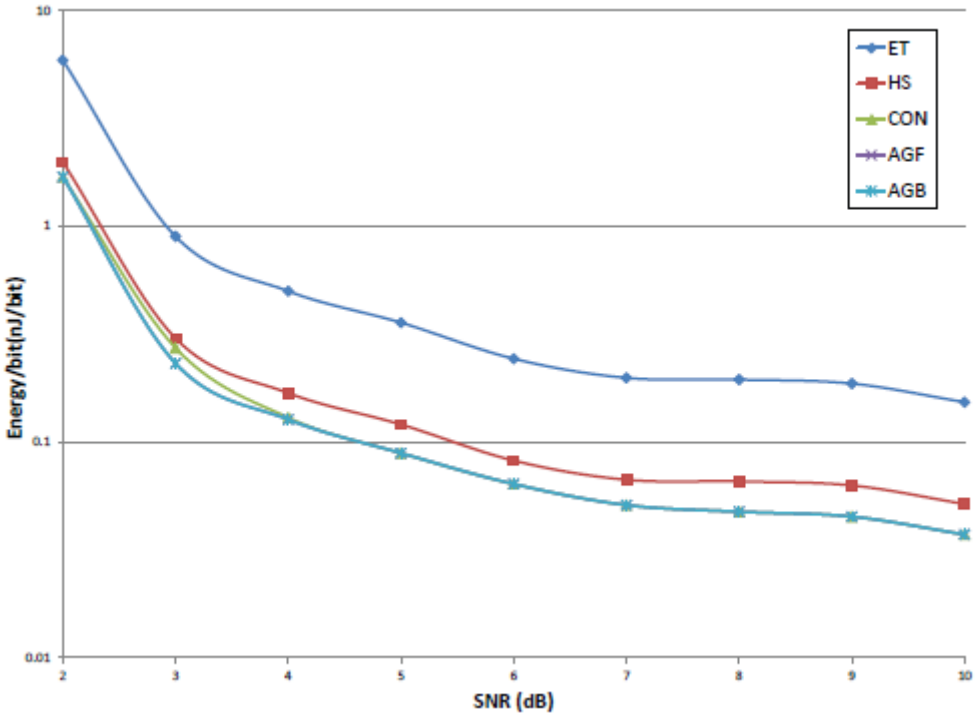


Figure 7-9– Energy Consumption for Different SNRs

We evaluated the energy consumption of all the approaches when changing the channel SNR from 2dB to 10dB and the results are plotted in Figure 7-9. The energy/bit is obtained by accessing Power

Supply Monitor and Controller inside the ML605 board through PMBus. Because AGF and AGB result in identical energy consumptions their plots are overlapped in the figure. One can observe in Figure 7-9 that:

a) Regardless of SNR value ET always consumes more energy than the other approaches and this can be explained by the fact that it has no capability to adapt to channel conditions. Energy consumed by ET decreases when the channel quality is getting better due to its early termination capability. At good channel quality, number of flipped bits decreases. Fewer flipped bits make decoding faster to converge and as a result, ET turns it power off earlier, reducing the consumed energy.
b) Our technique always outperforms both ET and HS. However at low SNRs (2 to 3 dB) the energy reduction is limited (only 10-15% and 15-23% reductions over HS, for CON and AGF/AGB, respectively) by the fact that there is not that much excess performance to exploit. However, for less noisier channels, i.e., SNRs from 3 to 10 dB, more excess performance is available and CON achieves a 22-28% energy reduction over HS thanks to its adaptability to exploit channel noise variability;
c) Because of its additional DVS technique, HS consumes 66% less energy than ET. CON (AGF/AGB) consumes around 71% (73%) and 76% (76%) less energy than ET, for bad and good channel quality, respectively; and (iv) At high SNR values CON, AGF, and AGB consume almost the same energy due to diminishing returns effect, while at low SNR values both AGF and AGB provide 15% energy reduction over CON.

We note that given that our technique does not alter the operating frequency it results in a better performance in terms of decoding throughput when compared to other decoders utilizing dynamic frequency scaling technique.

7.5. System Level Energy Model

Assuming the availability of energy models associated with the CPE and the decoders(both working on faulty and fault free hardware), one needs to determine the energy model of the entire system such that a certain reliability is achieved. The FPGA energy model presented in the previous section, which was validated through measurements, may not be applicable for aggressive voltage scaling in sub-threshold region due to limitations of FPGAs architectures. For such scaling, one needs to resort to models developed in earlier sections which are validated in HSPICE. Linking the channel quality (in our situation the outputs of the CPE), prediction to energy model of the decoder is work in progress which will benefit from the analysis and energy modeling performed at FPGA level. As results suggest, accurate estimation of channel quality at runtime is key in lowering the energy of the system. The various decoders' energy models are also important inputs into the multi-objective system optimization.

7.6. Conclusions

Energy models at different levels of the design hierarchy were considered. It is shown that energy models at gate level can be modelled using IGD. Also, energy models of two simple circuits

built using NAND gates (namely 3bit input XOR circuit and 3bit input MAJ circuit which are key building blocks in the decoders developed in WP4) follow the same IGD distribution. However, it is not clear yet how one can build on compositionality so that the energy model of large circuits can be derived from smaller blocks. Such works are currently being investigated and fed to other activities in other WPs.

Another interesting observation is that we find that one can use same distribution for modelling both energy and reliability (at least for the simple circuits which were studied). Propagating these models through a Boolean network to ascertain the energy model of a larger circuit represents our next challenge.

8. General Conclusions and Next Steps

In the period M13-M21 covered by Deliverable 2.2, we have developed high level fault models, simulation based methodologies and FPGA emulation methods for data dependent probabilistic fault analysis. The main contributions during this period for this WP can be summarized as follows:

1. **Inverse Gaussian Distribution Based Timing Analysis of Sub-powered CMOS Circuits.** An accurate and comprehensive delay model based on IGD considering fan-out effects was proposed and compared with the state of the art. The IGD model is suitable for both combinational and sequential gates. Our model not only provides high accuracy (close match to Monte Carlo Simulation (MCS) results), but more important, exhibits great flexibility against process and voltage supply variations. The calculation of the IGD model key parameters is straightforward, which is helpful for large circuit delay estimation. When compared with MCS data, for a discussion vehicle circuit, the average mismatch introduced by our approach is as low as 1.2% while, on the other hand, the simulation time is diminished by orders of magnitude. Moreover, the proposed IGD based estimation provides even higher accuracy than state of the art Gaussian Distribution based fitting.

2. **Correlated error modeling and degradation quantification for PDF-based circuits reliability assessment.** We proposed to employ a high-level degradation quantifier, i.e., an output voltage based Probability Density Function (PDF), in order to capture a gate (circuit) multiple correlated degradation effects, when being exposed to different aggression profiles. Moreover by propagating such PDFs throughout a larger circuit the correlation between different comprised gates behavior is inherently captured, and thus the correlation of different errors encountered in the circuit is being accounted for. The PDF-based gate reliability design-time pre-characterization was discussed and exemplified for an inverter and a NAND2 gate, as discussion vehicles. We also introduced a practical PDF based simulation framework and evaluated the reliability of a set of ISCAS'85 circuits, based on the prior PDFs that individual gates have accrued by means of a pre-characterization step.

3. **Multi-level simulation technology for reliability analysis of register transfer level (RTL) descriptions**. A hierarchical approach has been developed in order to perform accurate simulated fault injection (SFI) for RTL description. The RTL system is decomposed into simple blocks; data dependent gate level SFI is performed for these blocks; the results of these gate level simulations are used for deriving the SFI components at RTL level; RTL SFI is performed in order to estimate the reliability of the system.

4. **Cost effective FPGA probabilistic fault emulation**. A fault injection scheme based on true random number generators (TRNG) and shift registers has been proposed for the FPGA fault emulation of probabilistic errors. In order to control the emulation flow and to observe the design under test (DUT) response, we have employed FPGA vendor based logic analyzer (in our case Xilinx Chipscope). The main advantages of our methodology are: resource efficient emulation framework, due to the low cost implementation of the fault generation and insertion and of the usage of FPGA vendor based logic analyzer, and high accuracy for probabilistic fault modeling, as the generated and inserted errors are uncorrelated. The main

disadvantage is represented by low performance of the emulation, due to the high number of clock cycles required for shift register loading. In order to alleviate this problem, we have proposed a hybrid serial-parallel approach; the shift register is divided into several smaller shift registers, while using multiple TRNG.

5. **Development of data dependent fault models for interconnects.** We have developed four data dependent fault models for interconnect supplied at sub and near threshold voltages. Two of them, the full data dependent fault model and the partial data dependent fault model, capture in an accurate way the strong data dependency of the reliability characteristics of interconnects, which are due to the process variations and crosstalk effects. Regarding the full data dependent fault model, it is based on the fact that a specific wire is influenced by all the other wires within the bus due to capacitive and inductive crosstalk. The partial data dependent fault models represents a simplification of the previous fault model, based on the fact that the effect of inductive crosstalk is negligible; the reliability for a wire is dependent only on the neighboring wires.

In view of the previously mentioned contributions we have successfully achieved the second objective associated to this WP (Objective 2.2) as well as Milestone 3 (according to the project description of work). The proposed fault models and reliability evaluation methods will be used in the next period for analysis and validation of WP6 proof of concepts, as well as for the development of fault tolerant mechanisms for interconnects within WP4.

The main focus regarding future work within WP2 will consist of deriving energy models for sub-powered probabilistic CMOS circuits, as well as the development of energy-reliability models in the context of sub and near threshold computing.

References

[Agarwal04] K. Agarwal, D. Sylvester, D. Blaauw, F. Liu, and S. Vrudhula "Variational Delay Metrics for Interconnect Timing Analysis" Proceedings of the Design Automation Conference, pp 381-383, 2004

[Amaricai14] A. Amaricai, S. Nimara, O. Boncalo, J. Chen, E. Popovici "Probabilistic Gate Level Fault Modeling for Near and Sub-Threshold CMOS Circuits" Proc. Euromicro Digital System Design, 2014

[Aymerich12] N. Aymerich, S. Cotofana, and A. Rubio, "Degradation stochastic resonance (dsr) in ad-avg architectures," in Nanotechnology (IEEE NANO), 2012 12th IEEE Conference on, Aug 2012, pp. 1–4.

[Baetoniu08] C. Baetoniu "Method and Apparatus for True Random Number Generation" US Patent 7389316, 2008

[Baraza05] J. C. Baraza, J. Gracia, D. Gil, P.J. Gil, "Improvement of Fault Injection Techniques Based on VHDL Code Modification", 10th IEEE International High-Level Design Validation and Test Workshop, 2005

[Baraza08] J. C. Baraza, J. Gracia, S. Blanc, D. Gil, P. Gil, "Enhancement of Fault Injection Techniques Based on the Modification of VHDL code", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 6, 2008

[Bombieri11] N. Bombieri, F. Fummi, V. Guarnieri, "Accelarating RTL Fault Simulation through RTL-to-TLM Abstraction", 16th European Test Symposium, 2011

[Boncalo14] O. Boncalo, A. Amaricai, A. Hera, V. Savin "Cost-efficient FPGA layered LDPC decoder with serial AP-LLR processing" Proc. 24th Int. Conf. on Field Programmable Logic and Applications (FPL), 2014

[Boning99] D. Boning, S. Nassif, "Models of process variations in device and interconnect", Design of High-Performance Microprocessor Circuits, chapter 06, pp. 98-116, 1999

[Brkic13] S. Brkic, P. Ivanis, G. Djordjevic, B. Vasic, "Taylor Kuznetsov fault tolerant memories: a survey and results under correlated gate failures", Proc of 11th International Conference on Telecommunications in Modern Satellite and Broadcasting Services, TELSIKS 2013, Nis, Serbia, 2013

[Brglez89] F. Brglez, D. Bryan, K. Kozminski, "Combinational Profiles of Sequential Benchmark Circuits," Proc. 1989 Int. Symp. on Circuits and Systems (ISCAS), 1989

[Chen14] Chen, Jiaoyan, Christian Spagnol, Satish Grandhi, Emanuel Popovici, Sorin Cotofana, and Alexandru Amaricai. "Linear Compositional Delay Model for the Timing Analysis of Sub-Powered Combinational Circuits." In VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on, pp. 380-385. IEEE, 2014.

[Chow05] C. Chow, L. S. M. Tsui, P.-W. Leong, W. Luk, and S. J. E. Wilton, "Dynamic voltage scaling for commercial FPGAs," in Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on, Dec 2005, pp. 173–180.

[Civera02] P. Civera, L. Macchiarulo, M. Rebaudengo, M. S. Reorda, and M. Violante, "An FPGA-Based Approach for Speeding-Up Fault Injection Campaigns on Safety-Critical Circuits," Journal of Electronic Testing: Theory and Applications 18, Kluwer Academic Publishers, pp. 261–271, 2002.

[D2.1] i-RISC/Deliverable 2.1 "Circuit level fault models for sub-powered CMOS circuits for uncorrelated and correlated errors" online: http://www.i-risc.eu/home/liblocal/docs/iRISC_Deliverables/i-RISC_D2.1.pdf

[D4.1] i-RISC/Deliverable 4.1 "Taylor-Kuznetsov memory architectures using structured LDPC codes", online: http://www.i-risc.eu/home/liblocal/docs/iRISC_Deliverables/i-RISC_D2.1.pdf

[D5.1] i-RISC/Deliverable 5.1, "Data Structures and Design Flow for Fault Tolerant Circuit Synthesis", online: http://www.i-risc.eu/home/liblocal/docs/iRISC_Deliverables/i-RISC_D5.1.pdf

[Darabiha08] A. Darabiha, A. Carusone, and F. Kschischang, "Power reduction techniques for ldpc decoders," IEEE Journal of Solid-State Circuits, vol. 43, no. 8, pp. 1835–1845, Aug 2008.

[Dey00] D.K. Dey, S.K. Ghosh, B. K. Mallick, "Generalized Linear Models: A Bayesian Perspective", Marcel Deker, 2000

[Dupraz14] E. Dupraz, D. Declercq, B. Vasic, V. Savin "Analysis and Design of Finite Alphabet Iterative Decoders Robust to Faulty Hardware" IEEE Transactions on Communications (submitted)

[Ejlali08] A. Ejlali, S. G. Miremadi, "Eror propagation analysis using FPGA-based SEU-fault injection," Microelectronics Reliability, vol. 48 pp. 319–328, June 2008

[Evans13] A. Evans, D. Alexandrescu, E. Costenaro, L. Chen "Hierarchical RTL –Based Combinatorial SER Evaluation" Proc. International On Line Testing Symposium (IOLTS), 2013

[Favalli04] M. Favalli, C. Metra, "TMR Voting in the Presence of Crosstalk Faults at the Voter Inputs", IEEE Transactions on Reliability, vol. 53, no. 3, september 2004

[Feldhofer05] M. Feldhofer, J. Wolkerstorfer, V. Rijmen, "AES implementation on a grain of sand", IEE Proceedings on Information Security, vol. 152, issue 1, pag. 13-20, 2005

[Gammaitoni98] L. Gammaitoni, P. Hanggi, P. Jung, and F. Marchesoni, "Stochastic resonance," Rev. Mod. Phys., vol. 70, pp. 223–287, Jan 1998

[Gil08] D. Gil, L. J. Saiz, J. Gracia, J. C. Baraza, P. J. Gil, "Injecting Intermittent Faults for the Dependability Validation of Commercial Microcontrollers", IEEE International High Level Design Validation and Test (HLDVT) Workshop, 2008

[Hansen99] M. Hansen, H. Yalcin, J. P. Hayes, "Unveiling the ISCAS-85 Benchmarks: A Case Study in Reverse Engineering," IEEE Design and Test, vol. 16, no. 3, pp. 72-80, 1999

[Hasan10] S. Hasan, A. K. Palit, W. Anheier: "Fault Diagnosis of Crosstalk Induced Glitches and Delay Faults", 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010

[Jenn94] E. Jenn, J. Arlat, M. Rimen, J. Ohlsson, J. Karlsson, "Fault Injection into VHDL Models: The MEFISTO Tool", Proceedings 24th Annual International Symposium on Fault Tolerant Computing Systems (FTCS-24), 1994

[Koller09] D. Koller and N. Friedman, *Probabilistic graphical models - principles and techniques*, The Massachusetts Institute of Technology Press, 2009.

[Le14] K. Le, D. Declercq, F. Ghaffari, C. Spagnol, E. Popovici, P. Ivanis, B. Vasic, "Efficient Realization Of Probabilistic Gradient Descent Bit Flipping Decoders", IEEE ISCAS, 2014 (submitted)

[Lopez07] C. López-Ongil, M. García-Valderas, M. Portela-García, L. Entrena, "Autonomous Fault Emulation: A New FPGA-Based Acceleration System for Hardness Evaluation," IEEE Trans. On Nuclear Science, Vol. 54, No. 1, pp. 252-261, Feb. 2007

[Maniatakos09] M. Maniatakos, N. Karimi, C. Tirumurti, A. Jas, Y. Makris, "Instruction-Level Impact Comparison of RT- vs. Gate-Level Faults in a Modern Microprocessor Controller", 27th IEEE VLSI Test Symposium, 2009

[Marconi14] T. Marconi, C. Spagnol, E. Popovici, S. Cotofana, "Towards Energy Effective LDPC Decoding by Exploiting Channel Noise Variability", 22nd IFIP/IEEE International Conference on Very Large Scale integration, 2014

[May12] D. May, W. Stechele, "An FPGA-based Probability-aware Fault Simulator", International Conference on Embedded Computer Systems (SAMOS), 2012

[May13] D. May, W. Stechele, “A resource-efficient probabilistic fault simulator “ Proc. 23rd International Conference on Field Programmable Logic and Applications (FPL), September, 2013

[Nagaraj06] N. S. Nagaraj, “Interconnect process variations: theory and practice”, Proceedings of the 19th International Conference on VLSI Design (VLSID), 2006

[OpencoresAES] 128-bit AES crypto-chip Verilog design, available on Open Cores website: http://www.opencores.org

[PTM45] Predictive Technology Model, http://ptm.asu.edu.

[Rasheed14] O.-A. Rasheed, P. and Ivanis, B. Vasic, "Fault-Tolerant Probabilistic Gradient-Descent Bit Flipping Decoders," IEEE Commun. Letters, vol. 18, no. 9, pp. 1487 - 1490, September 2014

[Sauer11] M. Sauer, V. Tomashevich, J. Muller, M. Lewis, A. Spilla, I. Polian, B. Becker, and W. Burgard, “An FPGA-Based Framework for Run-time Injection and Analysis of Soft Errors in Microprocessors,” IEEE 17th International On-Line Testing Symposium, July, 2011

[Sanyal09] A. Sanyal, A. Pan, S. Kundu, “A study on impact of loading effect on capacitive crosstalk noise” Proc. Int. Symp. On Quality Electronic Design (ISQED), 2009

[Shirazi13] M. S. Shirazi, B. Morris, H. Selvaraj, “Fast FPGA-Based Fault Injection Tool for Embedded Processors,” 14th Int'l Symposium on Quality Electronic Design, 2013

[Sutherland99] Sutherland, Ivan Edward, Robert F. Sproull, and David F. Harris. “Logical effort: designing fast CMOS circuits.” Morgan Kaufmann, 1999.

[Thaker00] P. A. Thaker, “Register-Transfer Level Fault Modeling and Test Evaluation Technique for VLSI Circuits”, Proc. International Test Conference, 2000

[Xilinx11] Xilinx Chipscope, http://www.xilinx.com/tools/cspro.htm.

[Wishbone10] Wishbone B4: Wishbone System on Chip Interconnection Architecture for Portable IP Cores, 2010

[Zaynoun12] Zaynoun, Samy, Muhammed S. Khairy, Ahmed M. Eltawil, Fadi J. Kurdahi, and Amin Khajeh. "Fast error aware model for arithmetic and logic circuits." In Computer Design (ICCD), 2012 IEEE 30th International Conference on, pp. 322-328. IEEE, 2012.

