FP7-ICT / FET-OPEN – 309129 / i-RISC

D5.1

Data Structures and Design Flow for Fault Tolerant Circuit Synthesis

Editor:	Emanuel Popovici (UCC)
Deliverable nature:	Public
Due date:	January 31, 2014
Delivery date:	February 3, 2014
Version	1.0
Total number of pages:	64
Reviewed by:	i-RISC members
Keywords:	Data Structures, Binary Decision Diagrams (BDD), And-Inverter Graphs (AIG), Probability Density Function, Tools for Reliability Evaluation, Codeword Prediction Encoder.

Abstract

This report presents an overview of the work carried out in relation to Work Package 5 (WP5) during the first year of the project. The document reports on the investigation and selection of data structures to be used in the context of efficient fault tolerant circuit synthesis (Task 5.1), on the development of design and validation flows for fault tolerant circuit synthesis, analysis and optimization (Task 5.2), and on activities aimed to develop error coding driven graph augmentation (Task 5.3) a first step towards a Boole-Shannon type bound for circuit design.

List of Authors

Participant	Author
UCC	Emanuel Popovici (e.popovici@ucc.ie), Christian Spagnol (Christian.spagnol@ue.ucc.ie), Satish Kumar (sagrand@ue.ucc.ie), David McCarthy (davidmc@ue.ucc.ie)
TUD	Sorin Cotofana (s.d.cotofana@tudelft.nl), Nicoleta Cucu-Laurenciu (N.CucuLaurenciu@tudelft.nl)
CEA	Valentin Savin (valentin.savin@cea.fr)
UPT	Alexandru Amaricai (Amaricai@cs.upt.ro)

Table of Contents

List of Dissemination Activities

Accepted Papers

D. McCarthy, N. Zeinolabedini, J. Chen, and E. Popovici, "Predictable, Low-power Arithmetic Logic Unit for the 8051 Microcontroller using Asynchronous Logic", *IEEE International Conference on Microelectronics (MIEL)*, Belgrade, Serbia, May 2014.

Submitted Papers

J. Chen, S. Grandhi, C. Spagnol, A. Amaricai, S. Cotofana, and E. Popovici, "Linear Compositional Delay Model for Combinational Circuits Timing Analysis in Sub-Powered Systems", *Great Lakes Symposium on VLSI (GLSVLSI)*, Houston, TX, USA, May 2014, submitted.

S. Grandhi, C. Spagnol, J. Chen, S. Cotofana, and E. Popovici, "A Systematic Approach for Reliability Constrained Logic Optimization", *ACM Design Automation Conference (DAC)*, San Francisco, CA, USA, June 2014, submitted.

J. Chen, D. Mc Carthy, A. Tisserand, and E. Popovici, "Input data independent, low power Montgomery multiplier for safe ciphers", *IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, Potsdam, Germany, May 2014, submitted.

Workshop Presentations

J. Chen, C. Spagnol, S. Kumar, and E. Popovici, "Delay Relevant Reliability of CMOS Circuits" ; Presenter J. Chen, i-RISC Workshop, European Solid-State Circuits Conference(ESSCIRC), Bucharest, September 20, 2013.

C. Spagnol, S. Kumar, J. Chen, and E. Popovici, "A systematic Approach for Reliability Evaluation of Combinational Circuits" ; Presenter C. Spagnol., i-RISC Workshop, European Solid-State Circuits Conference(ESSCIRC), Bucharest, September 20, 2013.

S. Grandhi, C. Spagnol, J. Chen, D. McCarthy, E. Popovici, i-RISC: Innovative Reliable Chip Designs from Low-Powered Unreliable Components, Poster Presenter. S. Grandhi, Research in Action Forum, Microelectronics Week, Dublin, 6 November 2013.

Seminars

E. Popovici, "Systematic Power Estimation and Optimisation for Deep Submicron Digital ICs", CAIRN/INRIA Research Seminar, ENSSAT, 23 July 2013.

List of Figures

List of Tables

Abbreviations

AIG	AND-Invert Graph
BCH	Bose Chaudhuri Hocquenghem
BDD	Binary Decision Diagram
BED	Boolean Expression Diagrams
BSC	Binary Symmetric Channel
CPE	Codeword Prediction Encoder
ECC	Error Correcting Codes
EDA	Electronic Design Automation
IC	Integrated Circuit
LDGM	Low Density Generator Matrix
LDPC	Low Density Parity Codes
NNIG	Nand-Nor-Inverter-Graphs
PDF	Probability Density Function
QC	Quasi Cyclic
RBC	Reduced Boolean Circuits
UEP	Unequal Error Protection
WP	Work Package

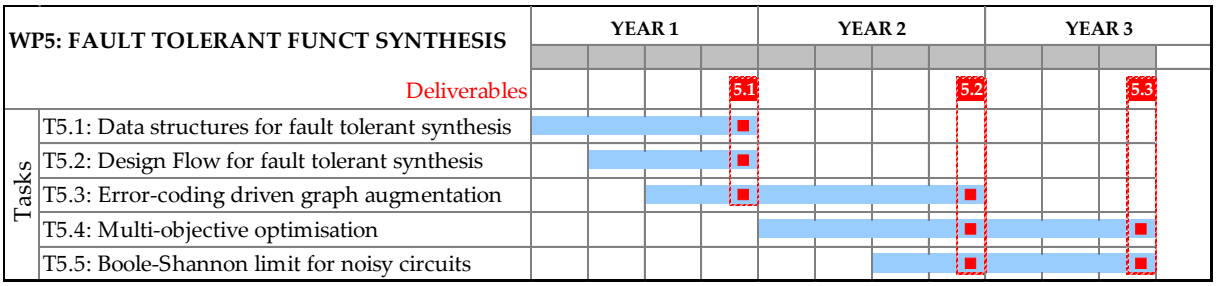
1. Executive Summary

Work Package (WP) 5 has been tasked with two major contributions towards the i-RISC goals. The **first WP5 goal** is the achievement of systematic synthesis and optimisation of reliable circuits, culminating with a multi-objective circuit design optimization, with respect to its size, energy consumption, latency, and all driven by reliability. The reliability measures and models (error/energy/power/etc.) defined in WP2 are central in designing tools and methodologies to achieve this goal. These synthesis tools also rely also on versatile data structures, which can capture these models effectively and accurately and lead to a seamless integration with other tools within the Integrated Circuit (IC) design flow tool chain. In order not to replicate significant body of prior art in circuit synthesis, we selected the data structures based on the availability of tool support (open source or academic tools) and, more importantly, on their capability to integrate i-RISC key concepts related to reliability aware circuit synthesis and optimization.. Consequently, we use a graph representation for the logic functions describing the circuits, and we manipulate/optimise these graphs using combinatorial optimization methods informed by coding theory principles (including some techniques inspired by WP3/WP4).

The introduction of reliability as circuit figure of merit leads to a 4-dimensional (area, delay, power, reliability) solution space, and has a tremendous influence on the complexity of the synthesis process. To limit the large search space, we initially confined our methodologies to the optimization of combinational logic. In the light of design for re-use, the developed tools are integrated within an industry recognized design flow. We propose some design flows which combine i-RISC custom tools together with widely used tools in the circuit design industry together with academic/open source tools. These flows will be used: a) for the systematic synthesis of reliable circuits within a multi-objective optimization framework; b) for the validation, simulation, and characterization of the circuits proposed within i-RISC.

The second WP5 goal is a fundamental study on the effectiveness of integrating error-correcting codes into the structural (Boolean network) implementation of the circuit logical functionality. We developed an initial framework for the synthesis of chips made from unreliable components, through the concept of error correcting codes driven graph augmentation. More precisely, we studied and evaluated some potential links between the logic representation of a digital circuit and error correcting codes in order to generate fault tolerant implementation of the logical functionality of the circuit. For the initial study, which will be expanded in the following two years, we considered two instances. Firstly, we utilize a Codeword Prediction Encoder for the protection of a Boolean function against potential errors. The focus of this approach is not on changing the combinational logic but on augmenting it to enable the retrieval of the correct output even if errors have occurred inside the circuit. While the encoder is embedded in the circuitry (hence prone to errors), the decoder is considered fault-free. For the second instance, we investigate the problem of protecting the functionality of a combinatorial circuit in the scenario where also the decoder is implemented with faulty hardware. In order to understand better the reliability conditions, we initially identify a subset of Boolean functions for which we can apply decoding techniques similar to WP3.

A Gantt chart of WP5 tasks and their time distribution is presented below.



One of the foundations for achieving the WP5 goals is the selection of the most relevant data structure, which allows for the systematic synthesis of reliable circuits (Task 5.1). In this context we analyzed a number of data structures and we selected the And-Inverter-Graph (AIG) structure as being the most appropriate for our goals due to its compactness, versatility to incorporate many parameters of concern (switching activity, delay, reliability, etc.), and scalability to any circuit size. Moreover, the proposed data structure also provides appropriate support to explore systematic multi-objective optimization methodology of fault tolerant circuits (Task 5.4). Another rationale for selecting AIG is that it is supported by the ABC open source academic tool which can be easily integrated in a Verilog/VHDL centered flow. Based on the selected data structure, a first version of an i-RISC tool for computing the reliability of a circuit was implemented. This tool is also integrated within a Verilog/VHDL Hardware Description Language based design flow (Task 5.2). A design flow is being proposed combining custom tools, academic tools with more established/industry accepted tools which will allow evaluation and validation of our designs at various levels (WP6).

Also related to Task 5.2 (and as an initial Task 5.4 step) we introduce a Probability Density Functions (PDFs) based Integrated Circuit (IC) reliability assessment framework. Since the efficiency of reliability driven design-time optimizations and of run-time management frameworks directly depend on the reliability evaluation accuracy, instead of relying on a single probabilistic value to reflect the reliability status of a circuit, we propose to employ a distribution of probabilities for a closer adherence to a faulty circuit stochastic behavior. The proposed framework, which takes an unorthodox approach towards reliability estimation, yields a fast and scalable reliability assessment approach, to be integrated in reliability aware synthesis tools (Task 5.2 and Task 5.4).

Another activity, which was started in the first i-RISC year relates to the systematic synthesis of fault tolerant combinational circuits (Task 5.3). A number of classes of circuits were identified and a first analysis into the encoding of such circuits was performed. Research is ongoing into expanding the classes of circuits towards a generalization of the applied methodology. Also, efficient decoding of such circuits is of paramount importance. Analysis of the complexity for both encoding and decoding for reliable circuits will be done within the frameworks of the synthesis tool and the defined design flow.

2. Data Structures for Fault Tolerant Circuit Synthesis (Task 5.1)

Abstract: The aim of this task is to study the available set of data structures, their benefits and limitations, and the associated open source EDA tools. The emphasis is on data structures, which are: (i) easily expandable in terms of the number of covered parameters, e.g., area, delay, power consumption, reliability, and (ii) easy to manipulate in the context of combinatorial optimization algorithms, i.e., lead to relatively compact circuit representations while incorporating these parameters. Finally the availability of open source tools has been also considered.

Publications: "unpublished work".

2.1. Introduction

Boolean function structural representation, logic synthesis, and technology mapping are important issues in the design of digital circuits. Till date, many data structures to be utilized within Electronic Design Automation (EDA) tools for synthesis and optimisation of digital logic circuits have been proposed **[Meinel98]**. These data structures were particularly tailored for particular metrics of interests such as area/size, delay, or power consumption. Reliability metric is fast emerging as a key design metric in the low power design as well as in the context of deep submicron technologies. Some of the most desirable properties for EDA data structures are the compactness of representation as well as the easiness to convert from and to a logic network. Finally, the data structure should be easily adaptable to a large number of design parameters. These properties may lead to the design of fast EDA tools for both synthesis and simulation. We categorise the EDA tools into Industrial and Academic, with the latter being mostly open source tools. Within the i-RISC project, we attempt to take advantage of the available features in the two EDA domains. At the same time, custom i-RISC tools are developed, and integrated along with the selected tools, to augment the existing software and allowing them to deal with probability of errors and reliability.

Each of the EDA tools has two major blocks: the data structures used to represent the functionality of the logic circuit and the algorithm to synthesize the optimised circuit or to compute a specific metric for the design (area, power, switching activity, etc.). The underlying data structure has an important role in determining the speed and efficiency of the tool. Hence, the first step in fault tolerant circuit synthesis, which is the i-RISC requirement, is to select a compact data structure that can easily be expanded to allow the algorithm to capture the error probability information. The first step to make this decision is to critically consider the various alternatives data structures already present in literature and/or used in the industry.

We considered two of the most commonly used data structures in EDA for digital circuit synthesis namely the Binary Decision Diagrams and the And Inverter Graphs. Some of the basic principles, advantages and shortcoming of each are presented in the following two sections.

2.2. Binary Decision Diagrams

Binary Decision Diagrams (BDDs) were originally introduced in the late 50s to represent a Boolean function being derived from the Shannon expansion. BDDs became of significant importance in EDA after the introduction of the **Reduced Ordered Binary Decision Diagram** (**ROBDD)** by in 1986 **[Bryant 86]**. Since their introduction, the impact of ROBDD has being extensive particularly in the area of formal verification but also in logic synthesis. For long time they have been one of the most used data structures for representing Boolean functions. Two properties of the ROBDD are the cause of such wide acceptance. First, they are canonical (unique), if two circuit have the same ROBDD then they are logically equivalent. It is evident that this property facilitates the development of not just formal verification tool but also in optimization and testing. Second, they are highly effective at representing combinatorial large sets. This again led to several breakthroughs with regards to Finite State Machine equivalence checking and logic minimization **[Coudert89] [Coudert93]**.

At the foundation of BDD is the concept of *Shannon expansion* **[Shannon48].** Given a Boolean function $f(x_1, \dots, x_n)$ the Shannon expansion states that it can be decomposed in $(x_1, \dots, x_n) = x_i f_{x_i} + \bar{x}_i f_{\bar{x}_i}$, where f_{x_i} and $f_{\bar{x}_i}$ are obtained by the function $f(x_1, \dots, x_n)$ by setting the variable x_i to 1 and 0 respectively. The recursive application of the Shannon expansion on the two restriction functions guided by an arbitrary fixed total ordering of variables(e.g. $x_1 < x_2 < x_3 < \cdots$) leads to a fully expanded expression which is unique. Take for example the function: $f(x_1, x_2, x_3) = x_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_3$ the full Shannon expansion is:

$$f(x_1, x_2, x_3) = x_1(x_2 x_3 + \bar{x}_3) + \bar{x}_1(\bar{x}_2 x_3 + \bar{x}_3)$$

$$f(x_1, x_2, x_3) = x_1(x_2(x_3 + \bar{x}_3) + \bar{x}_2(\bar{x}_3)) + \bar{x}_1(x_2(\bar{x}_3) + \bar{x}_2(x_3))$$

Full Shannon expansion is therefore a canonical form. What is of interest is the possibility to represent this expansion in a graph form and the transformation that can be applied to this graph representation to considerably reducing size without losing the canonical form.

Figure 1 shows the BDD graph *G*(*V,E*) corresponding to the Boolean function above. For each layer of the graph a decision on one variable is taken. ('0' dotted line, '1' full line). The final layer shows the values of the function for each set of decision taken.

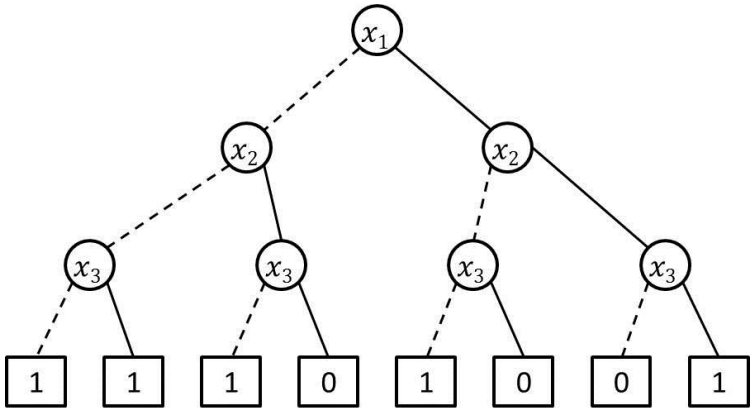


Figure 1: BDD Representation of the function $f(x_1, x_2, x_3) = x_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 x_3 + \bar{x}_3$

A rigorous definition of the BDD states that any Boolean function can be represented as a rooted, directed acyclic graph denoted as *G*(*V,E*), where *V* is the set of vertices and *E* is the set of edges, which link the vertices denoted in *V*. Any $v \in V$ is either a non-terminal node or one of the two terminal nodes namely 0-terminal and 1-terminal. The non-terminal nodes are the decision nodes,

which are labelled by a Boolean variable. These nodes have two child nodes called low child and high child. Any decision node $e \in E$ points to the low-child or high child depending upon the assignment of the Boolean variable labelled on the decision node to be '0' or '1', respectively. The top decision node of the graph is called the root node.

Since it is fundamental that the expansion uses the variable in a fixed order for the representation to be canonical the graph is called an ordered-BDD (OBDD). From this graph representation a reduced size graph can be obtained (ROBDD) with a simple procedure.

1. Replace all leaf with the same value by a single vertex. Redirect all edges incident to the original vertices to this single vertex
2. Process all layers from the bottom up. If in any layer two vertices are found that point to the same children for both 0/1 decisions, then merge the vertices
3. If a vertex exists for which both 0/1 decisions lead to the same child, remove the vertex and point all parent to its child.

Figure 2 shows the three steps applied on the example graph.

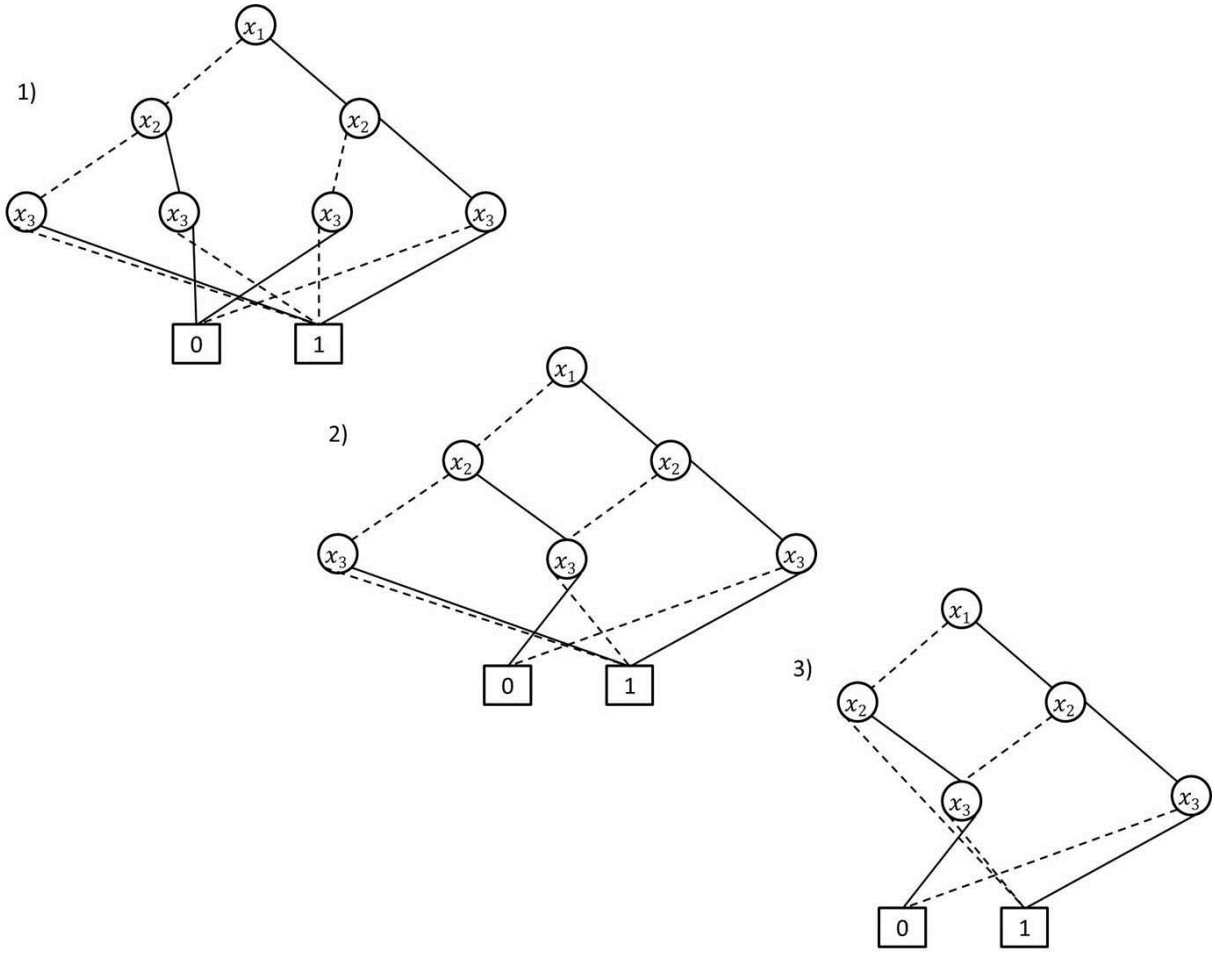


Figure 2: Reducing process applied to the example ODBB

It is possible to construct the ROBDD directly from the logic equation (or other representation) rather than the BDD (since BDD always has size that is exponential on the number of inputs). This allows the ROBDD to be treated as abstract data type on which several procedures can be defined to automate building, merging and manipulating of the ROBDD.

BDDs provide compact representations of Boolean expressions, and there are efficient algorithms for performing different types of logical operations on them. These properties make BDDs widely used in the EDA applications. However despite a sizeable body of related research **[Brayton10] [Balasubramanian07]**, they have reached the limit of their scalability due to the always increasing complexity of modern circuitry. The root cause of this scalability problem is optimal ordering of the variables in the Shannon expansion, and in particular the dependency of the size of the BDD on the chosen ordering. Finding the optimal ordering is an NP-complete problem and it becomes intractable as soon as the number of inputs grows **[Bollig96].** As a consequence, the size of the ROBDD cannot be guaranteed to grow linearly. This dependence is so drastic that in many cases it is impractical to build the BDD at all **[Hachtel00].** Moreover, there exist families of functions for which the number of vertices in their ROBDD grows exponentially and independently of the variable ordering chosen. Unfortunately some of these functions are of common use, for example the multiplication function. This problem lead to the raise of new representation that offer significant speed up and can better deal with the size of modern circuit, the And –Inverted graph, subject of next section.

BDDs have been used extensively in **[Wright00]**, **[Lindgren01],** and **[Ueda95]** for switching activity estimation and optimisation. In these works, the zero-delay model is assumed and reduced switching activity is obtained either by reducing the BDD size or by changing the order of the BDD. The main drawback of these techniques is their computational complexity. Computation time and memory footprint increase rapidly with the circuit size.

The probabilistic technique for switching activity estimation using BDD follows a BDD traversal method **[Ueda99]**. The BDD structure involved in this traversal method increases rapidly with circuit size, hence making it cumbersome for large circuits. Many techniques in **[Tani93]**, **[Felipe08, FelipeY08, Felipe05]** work on reducing the BDD size as used in the BDD traversal method in order to reduce the memory usage and computation time. Several other power-aware BDD-based synthesis techniques are introduced in **[Tinmaung07]**. Most often BDD-based synthesis techniques relate only to one parameter(either area or delay or power consumption). Discussions of BDD-based synthesis techniques which are compared and contrasted through the potential for multi-objective optimisation are presented in **[Mehrotra13].**

2.3. AND-Inverter Graphs

AND-Inverter Graphs (AIG) is another way to represent a Boolean function. AND-Inverter logic gained significant attention in both academia and industry since 1960's through a number of works at IBM and more recently at UC Berkeley. These works considered this data structure for logic and delay optimisation as well as verification. Recently, these data structures gained significant interest in a number of algorithms for design automation tools (for both FPGA and ASIC technologies) as they offer better performance than BDDs or other data structures **[Mishchenko06][Mehrotra13][Machado12]**. It is a novel data structure which unifies equivalence checking, synthesis and technology mapping. The use of AIG nodes is justified as it produces a better correlation with final area and delay once the circuit has been mapped to a target technology **[Figueiro11]**. This advantage with respect to literals comes from (a) the fact that AIGs are multi-level representations allowing sharing of nodes; and (b) the AIG node is a simple structure, which keeps correlation with area as all nodes have homogeneous simple granularity.

An AND-Inverter Graph (AIG) is a directed acyclic graph denoted as $G(V,E)$ where V is the set of vertices and E is the set of edges which link any two vertices. The graph represents the structural functionality of the associated digital logic circuit. Any $v \in V$ either represents a 2-input AND gate or a terminal node labelled with a variable name. The variable names are the names of the input variables of the logic circuit. Any $e \in E$ is the input of a 2-input AND gate.

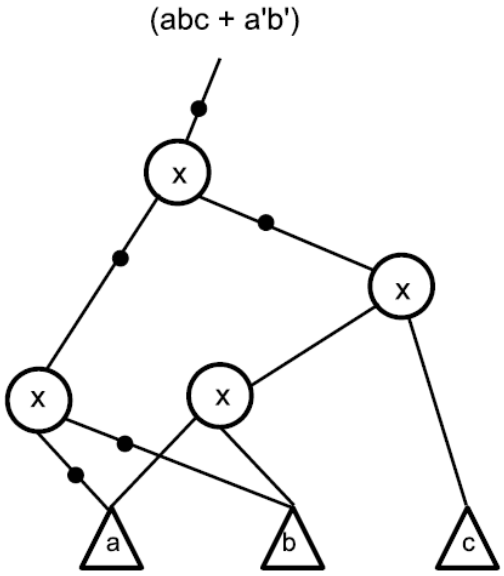


Figure 3: AIG Representation

Figure 3 shows an AIG example representing the *abc* + *a'b'* functionality. The inputs are *a*, *b,* and *c* and are represented as terminal nodes denoted within a triangle. The nodes which represent 2-input AND gates are denoted within a circle. Edges with a dot indicate negation, i.e., the inversion of that input. AIGs provide a more compact representation of digital circuits than BDDs or sum of products, are simple, flexible, scalable and easy to convert from a network of logic gates. Formal verification is also simpler as they can be easily mapped to a netlist. Their size is proportional with the size of the circuit. There are many works on the optimisation of AIG circuits(for power, delay or area) but to our knowledge our work is the first attempt to optimise the circuits for improved reliability.

In addition to synthesis of combinational logic, AIGs were also used for sequential logic synthesis **[Mischenko13][ABC12]**. The simple and uniform structure of AIGs allows rewriting, simulation, mapping, and verification to share the same data structure. It was shown in **[Mehrotra13]** that the data structure is also suited for power driven delay optimisation and delay driven power optimisation resulting in significant power/delay savings. These characteristics along with the existence of an open synthesis tool ABC make AIGs an ideal candidate for a versatile i-RISC data structure.

From the reliability methodology perspective, we believe that a feature of AIG that matches our goals is that it is not canonical. Such a property may allow easier/faster implementation of error coding graph augmentation techniques as the ones proposed in section 5. Figure 4 represents two functionally equivalent representations of a given combinational circuit. The advantage of this particular feature is that it allows the designer to insert/remove redundant nodes that can help in improving the redundancy of the circuit.

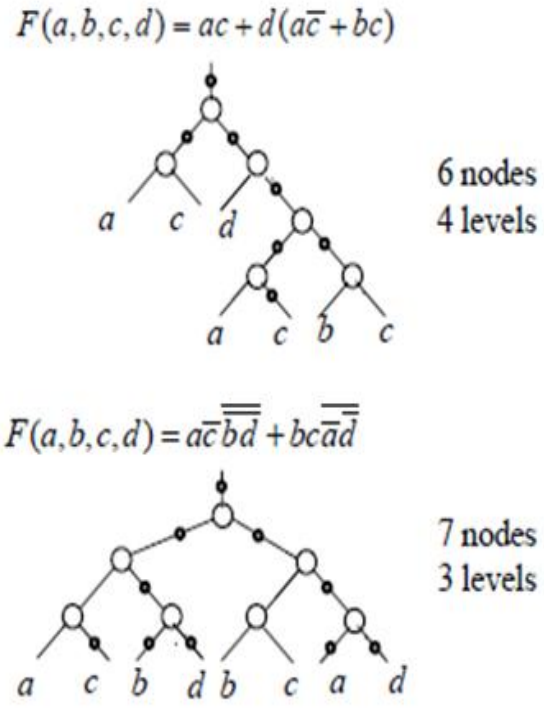


Figure 4: Two functionally equivalent AIGs

Some other related data structures include Boolean Expression Diagrams (BED) **[AndersenHulgard-LICS97]**, Reduced Boolean Circuits (RBC) **[AbdullaBjesseEen-TACAS00]**, Nand-Nor-Inverter-Graphs(NNIG) **[Mehrotra13, Balasubramanian 06]**. However, these data structures are not as widely used at present and more importantly the number of tools for synthesis and simulation are not openly available.

2.4. Summary

Data structures and algorithms largely determine the efficiency of the tool in implementing new applications. Previous tools including SIS, BDD, MVSIS and FBDD get inefficient for large circuits and do not provide enough flexibility for binary synthesis, owing to their type of data structures. ABC deals with simple and flexible data structures, which promise improvements in quality and runtime of several applications which include both sequential as well as combinational logic. Moreover, the BDD are not the best suitable for the aim of the i-RISC project of computing reliability. The major problem in this regards is the fact that the BDD represent only the logic functionality of the combinatorial logic losing any direct link with the physical implementation**.** Technology mapping task is more difficult in BDD AIG are better suited to this task since each node on the graph represents an actual physical port.

For these reasons the AND-Inverter graph has been chosen as the underlying data structure. The data structure has then been enhanced to store information regarding the reliability of the circuit it represents. Once the AIG graph is chosen as the most suitable data structure the ABC tool is the natural choice. The ABC tool has several advantages: a) it is open source b) It is actively maintained

and c) it has several routing suitable for synthesis/mapping and verification. As part of Task 5.2 a reliability synthesis and optimisation sub-packages has been developed. The ABC tool (including the sub-packages) is then incorporated in a design flow which involves also industry tools, namely from Synopsys. Although AIG data structure promise significant advantages over other data structures including BDDs, for some specialised tasks it is possible to use various procedures associated to specific data structures(particularly BDDs) through various packages included in the tool.

3. Design Flow for Fault Tolerant Circuit Synthesis, Analysis, and Optimization (Task 5.2)

Abstract: In this section an integrated design flow comprising of industry and academic tools is proposed. Design compiler from Synopsys is used to synthesize a high level description of a circuit specified in Verilog. The resulting gate level netlist is then translated into an AIG data structure, which is then analysed/optimized/manipulated within the ABC tool (or an i-RISC custom tool). The resulting synthesized netlist is then technology mapped and analysed using standard industry tools from Synopsys. Such a flow makes use of a number of state of the art tools within Synopsys complementing them with the custom i-RISC tools for ultra-low power and reliable circuit design. We show that a heterogeneous flow (using both academic and industry tools) is necessary. First, we analyze the reliability by using a simulation based flow, which relies entirely on industry tools using a high level of abstraction. Another methodology for reliability analysis is presented in WP2 when HSPICE and Monte Carlo methods are extensively used. Both these methods lead to an infeasible solution for accurate reliability computation. Hence, two approaches are followed. One approach considers a design flow incorporating hybrid description of designs with different components described at different levels of abstraction (Verilog and HSPICE). In order to scope the performance of our results, we also use the HSIM Plus tool from Synopsys, which allows a hybrid code (Verilog and HSPICE) to be simulated and analyzed. Such an approach allows us to scope and analyze a particular part of the design in HSPICE while the rest of the design is written and simulated in Verilog. This flow has been evaluated and refined on a number of arithmetic circuits and will be a basis for further evaluation within WP6 in particular. The second approach considers custom tools in conjunction with reliability models developed in WP2.

Publications: One submitted paper at DAC2014; one accepted paper at MIEL2014;

3.1. Academic EDA Tools

A number of open-source EDA tools provide a programming environment and a solid platform for research in logic synthesis, technology mapping, power and delay estimation and optimisation (Figure 5). These academic tools represent the Boolean functionality of any digital circuit using one of the data structure detailed in the previous sections. Manipulation for logic synthesis, optimisation and technology mapping are done on these data structures.

- The CUDD **[Yanushkevich05]** package provides functions to manipulate Binary Decision Diagrams (BDDs). The package also provides a large assortment of variable reordering methods, to reduce the number of BDD nodes.

- FBDD **[Wu05]** is an open-source logic synthesis package based on BDDs. The package employs several new techniques, including folded logic transformations and two-variable sharing extraction.
- SIS **[Sentovich92]** is another interactive tool for synthesis and optimisation of sequential circuits. Given a state transition table, a signal transition graph, or a logic-level description of a sequential circuit, it produces an optimised netlists in the target technology while preserving the sequential input-output behaviour.

Although many techniques applied in the tools demonstrate robustness and optimality in EDA, the tools, other than SIS, have not found application in industry and can only be considered theoretical frameworks. But due to the limitation associated with BDD as detailed before, this tool has been replaced with a more scalable tool called 'ABC'.

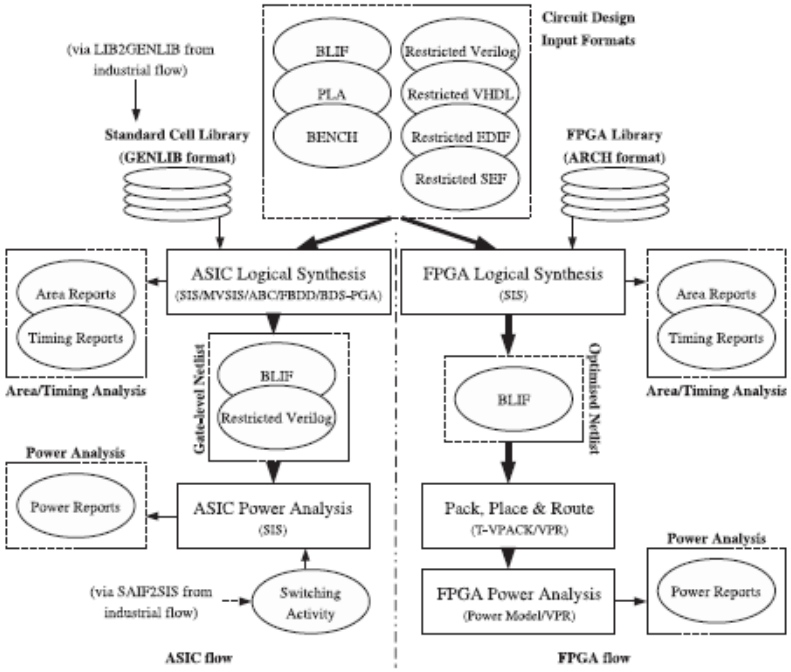


Figure 5: A Typical Academic Design Flow

ABC **[Brayton10]** is a logic synthesis and verification tool which performs scalable logic optimisation based on AND-Inverter Graphs (AIGs) **[Mishchenko06]**. In all of these academic tools, data structures and algorithms largely determine the efficiency of the tool in implementing new applications. Previous tools like SIS, BDD, and FBDD get inefficient for large circuits and do not provide enough flexibility for binary synthesis, owing to their type of data structures. ABC deals with simple and flexible data structures, i.e. AIGs, which promise improvements in quality and runtime of several applications which include both sequential as well as combinational logic. Our major work is based on AND-Inverter graphs, manipulated in ABC using some of i-RISC custom synthesis and optimisation sub-packages. The tool allows for a variety of functional representations including BDDs, Sum of Products to solve specific tasks while switching to AIG for the mainstream network manipulation. This versatility allows us to integrate custom i-RISC packages within the ABC tool. Then, the ABC tool (including the sub-packages) is incorporated in a design flow which involves also industry tools, namely from Synopsys for a thorough evaluation of the synthesis results.

3.2. Industry EDA Tools

An overview of a typical design flow appears in Figure 6. The green arrow indicates the path from RTL through to a power analysis on the final placed and routed netlist. The flow turns RTL and a set of constraints into placed-and-routed gates. It is divided into several stages, namely:

• RTL Simulation

RTL simulation for development and functional verification, under the control of a top-level testbench;

• Synthesis

Synthesis of RTL and constraints to a gate-level netlist in TSMC standard cells;

• Placement

Construction of a floorplan and automated cell placement;

• Routing

Construction of a clock tree and automated wire routing;

• Gate-level Simulation

Gate-level simulation with annotated delays using the final IC compiler netlist and the same testbench as used for RTL simulation;

• Equivalence Checking

Formal equivalence check between placed-and-routed gates and original RTL using Synopsys Formality;

• Timing Analysis

Static timing analysis with back-annotated parasitics, and

• Power Analysis

Analysis of average and time-based power using gate-level activity and back-annotated parasitics.

A typical digital IC design flow from Industry combines a number of tools from providers including Synopsys, Cadence and Mentor Graphics. The front end of the design flow is dominated by Synopsys tools while the backend is typically made of Cadence/Mentor Graphics tools. Within the i-RISC project, most of research is at higher level (frontend) with some lower end models in the form of HSPICE or SPICE. The i-RISC designs which will be evaluated as part of WP6 require a design flow which can handle both high level constructs as well as lower level models for accurate and fast evaluation of performance. HSIMplus is a powerful tool which is capable of circuit check, reliability assessment associated with physical effects in nanometer IC designs, etc.

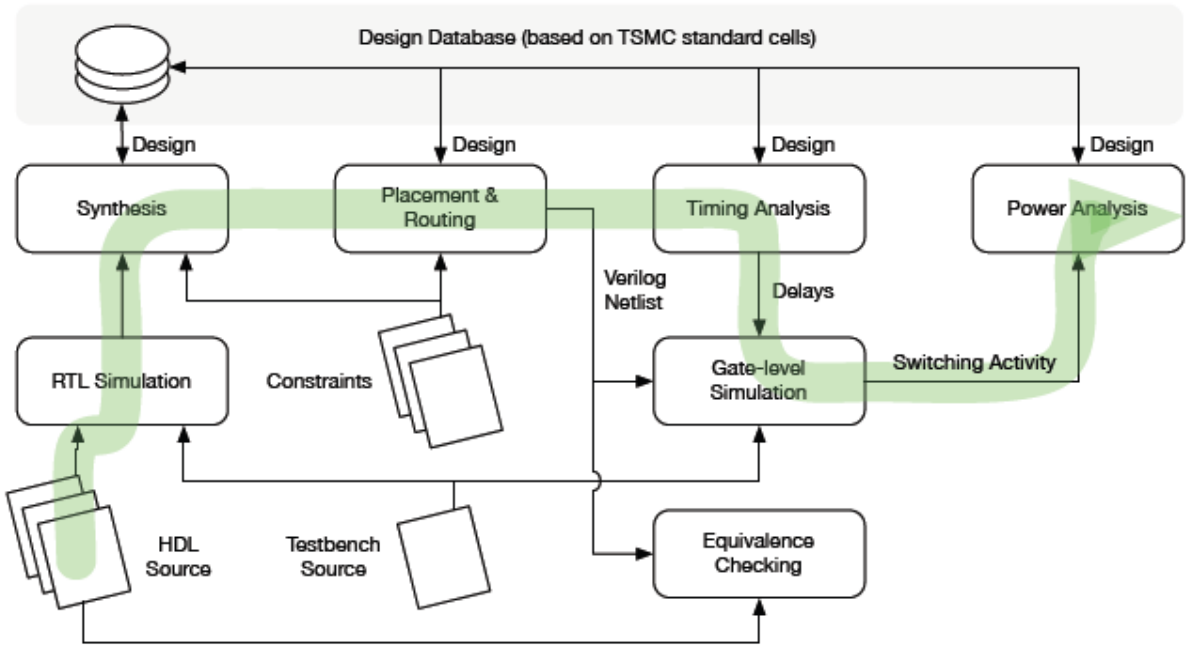


Figure 6: Implementation Flow overview

We use both the academic and the industry tools together to form an integrated design flow. This also includes some custom scripts and developing wrappers on top of existing tools in order to facilitate the reliability analysis and optimization flow. Figure 7 describes the integrated flow in the simplest possible fashion. We use design compiler to flatten the RTL of the IP and use some custom developed scripts to make necessary modifications. This is because the 'ABC' tool cannot understand the direct output of the design compiler. We modify the circuit using the wrappers developed in-house on top of the 'ABC' tool. Then, we use various industry standard tools to perform formal verification and analyse the performance improvement.

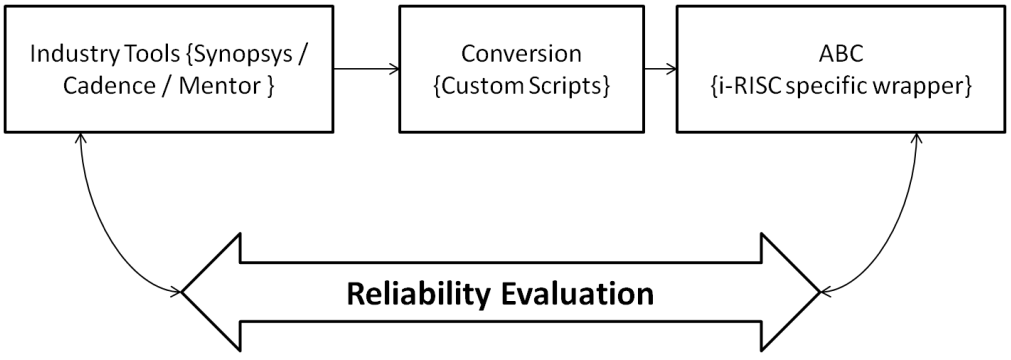


Figure 7: Integrated Flow

The structure of HSIMplus is presented in Figure 8 . It offers good accuracy and impressive simulation time reduction when compared with other SPICE simulation tools.

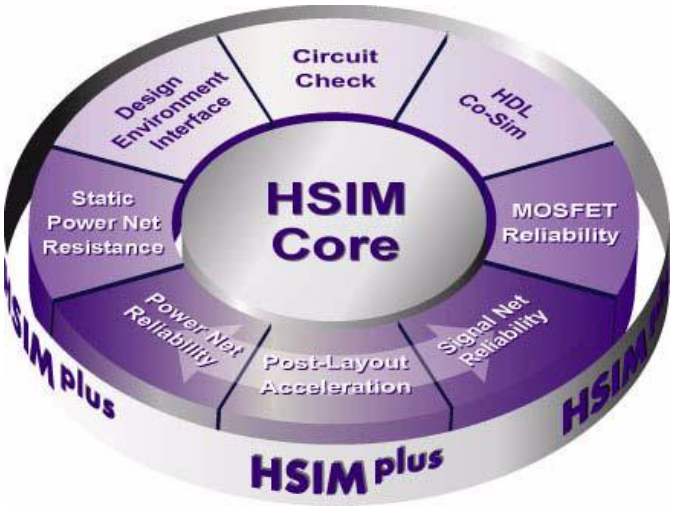


Figure 8: HSIMplus Features

As a complete transistor-level simulation and analysis platform, HSIMplus also provides the capability to simulate designs of various sizes and complexities, described even at different levels of abstraction, thanks to its HDL Co-Sim capability which is compatible with several tools from Synopsys and Cadence.

This attractive feature allows the integration of both Verilog/VHDL and SPICE netlists during the simulation. SPICE simulation is famous for its custom adjustable controllability and high accuracy while it results in long simulation time for large circuits. On the other hand, HDL simulation could easily handle big and complex digital circuits but with little or no control on parameters such as power supply, temperature, variations, etc. With HDL Co-sim function, we could easily take advantage of both approaches by manipulating the parameters in SPICE netlist where high accuracy is necessary and keeping other parts in HDL. This unique solution is of interest for both verification and optimization.

3.3. Tools for Circuit Reliability Computation

One of the keys for developing an efficient optimization tool is the availability of accurate reliability information as well as efficient/fast algorithms for computing the reliability of logic functions representing partial solutions during the optimization process. A pure reliability analysis based on HSPICE Monte Carlo simulations is not feasible due to a prohibitive computation time and excessive resource requirements. In this section we address the problem of computing reliability information investigating three different approaches and presenting the tools that has been developed for each approach. The first approach consist on simulating the combinatorial circuit under test and injecting error with specify characteristics to gather statistical information on how these error propagate through the circuit . The second method builds a model of the combinatorial circuit and calculates the reliability of the circuit based on mathematical analysis of the model. Finally a reduce complexity bounding approach is presented.

3.3.1. Simulation Based Tools

HDL based Simulated Fault Injection (SFI) represents a good trade-off between time, cost, and accuracy for reliability evaluation of circuits **[Jenn94] [Gil08]**. Fault injection techniques have been classified in two main categories **[Jenn94]**: techniques that do not require code modification (based

on simulator commands **[Sheng08]**) and those based on HDL code modifications (mutants and saboteurs **[Jenn94] [Gil08]**). The latter require more coding effort; however, they present higher fault modeling capability. Saboteurs represent an entity/module, which alters signal timing or values. Mutants represent modified architectures/modules which replace the correct one.

During the i-RISC project, we have developed SFI methodologies for probabilistic circuits, which can accurately model the probabilistic switching behavior of sub-powered circuits. Our approach for combinational circuits relies on mutants. Four types of mutants have been developed, based on the desired fault modeling:

1. Gate output probabilistic mutant – it alters the gate output with a given probability.
2. Gate output switching probabilistic mutant – the mutated gate switches correctly with a given probability.
3. Gate output switching type probabilistic mutant – different probabilities are taken into account for the two types of switching (charging and discharging).
4. Gate input switching type probabilistic mutant – the input switching is taken into account for this type of mutant.

The proposed SFI methodology relies on two phases: (i) the setup phase and (ii) the simulation and data analysis phase. In (i), the fault parameters are set, the gates are mutated according to a parameter set and the fault model, the input vectors are selected, the ideal (error free) circuit is simulated with the input vectors set and a testbench is generated. The (ii) phase consists of the simulation of the not ideal circuit and comparison with the ideal circuit results.

In the context of reliability evaluation within the synthesis process, simulated fault injection has two major disadvantages:

1. Requires a large number of simulations for probabilistic circuits; because probabilities were obtained using random number generators, accurate results can be only obtained when performing at least one or two orders of magnitude more simulations then the desired accuracy, e.g., for a failure rate of 1 in 1000, a number of at least several tens of thousands of simulations are required. This leads to large execution time overhead.
2. Relies on external HDL simulators; the SFI processes are performed using dedicated HDL simulators, e.g., Modelsim from Mentor Graphics, VCS.

Therefore, reliability evaluation of probabilistic circuits using HDL based SFI within synthesis process is unfeasible. Hence alternative methodologies have been researched and included within the proposed design flow.

3.3.2. Gate Error Model Based Tool

As AIG's are used to represent combinational circuits, we need to develop probabilistic gate error models for AND & inverter logic gates. Consider an unreliable AND gate, such a gate can be modelled as an ideal (error free) AND gate followed by a faulty buffer that models the stochastic behaviour of the errors. This model moves the entire error statistics on the gate output and so it implicitly assumes symmetrical error behaviour in relation to the inputs. This assumption holds true for many physical gate implementations and simplifies somewhat the analysis. The two nodes Z* and Z, as shown in Figure 9 are named as internal and the external output node, respectively. To analyse this model behaviour, let's consider that errors on the output of a combinational gate can be mainly due

to two reasons: (i) errors on the gate input nodes and (ii) intrinsic errors within the gate. In this model, the circuit under evaluation primary inputs are assumed to be error free and statistical independent.

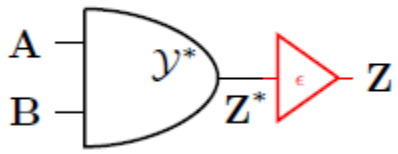


Figure 9: Unreliable AND Gate Model

3.3.2.1. Ideal *2-Input AND Gate*

As AIG's comprises of only 2-input AND gates, we restrict our analysis to 2-input AND gates and note that its extension to multi-input AND gates is straightforward. The static probability values of the internal output node Z* can be expressed as:

$$P_{Z^*}(1) = P[A = 1, B = 1] \quad (1)$$

$$P_{Z^*}(0) = 1 - P_{Z^*}(1) \quad (2)$$

First, the error on the internal output node (Z*) due to errors on the input nodes of an ideal AND gate is computed. We note that an error on an input need not necessarily result in a wrong output value. For example an error free '0' on one of the gate inputs would mask the error on the other input node from being propagated onto the output. Similarly, consider the scenario where the inputs are set to '0' & '1' and both are erroneous. This event of double error mutually negates each other and results in a correct output value. Hence, due to masking and double error events, there are no simple rules to predict the gate output state. Tab. 1 presents an exhaustive enumeration on all the possible cases.

Table 1 : Ideal AND gate with unreliable inputs

State Of Nodes Under Reliable Conditions			**State Of Nodes / correct value**			**Error Probability**
A	***B***	***Z****	***A***	***B***	***Z****	
0	0	0	c/0 c/0 ε/1 ε/1	c/0 ε/1 c/0 ε/1	c/0 c/0 c/0 ε/1	$P[A_0,B_0, A_\varepsilon, B_\varepsilon]$
0	1	0	c/0 c/0 ε/1 ε/1	ε/0 c/1 ε/0 c1	c/0 c/0 c/0 ε1	$P[A_0,B_1, A_\varepsilon, B_c]$
1	0	0	ε/0 ε/0 c/1 c/1	c/0 ε/1 c/0 ε/1	c/0 c/0 c/0 ε/1	$P[A_1,B_0, A_c, B_\varepsilon]$
1	1	1	ε/0 ε/0 c/1 c/1	ε/0 c/1 ε/0 c/1	ε/0 ε/0 ε/0 c/1	$P[A_1,B_1, A_\varepsilon, B_\varepsilon]$ $P[A_1,B_1, A_\varepsilon, B_c]$ $P[A_1,B_1, A_c, B_\varepsilon]$

To explain the table, consider the case when A=0 and B=0. Then, the internal output Z* should be ideally 0. Now, each of the inputs can assume an error (ε) or a correct (c) state. The state of the inputs determines if Z*=0 is correct or not. It is clear that, for these inputs values, the internal output node is in error if and only if both the inputs are in error. It is evident from Tab. 1 that error on one or both inputs need not necessarily translate into an output error. Only 6 of the possible 16 cases result in an internal output error. The probabilities for each of these events to occur are

presented in the last column. In order to arrive at a close form representation of AND gate output node error probability, we assume that the gate inputs are independent, i.e. there is no reconvergence, in order to simplify analysis. This allows for the utilization of simple formulas to compute reliability and reduce the overall algorithm execution time. The internal node error probability can be expressed as the sum of all the six terms in Tab. 1 that result in an erroneous output and evaluates to:

$$P_\varepsilon(Z^*) = P_\varepsilon(A)P_\varepsilon(B)P_A(0)P_B(0) + P_\varepsilon(A)[1 - P_\varepsilon(B)]P_A(0)P_B(1) + [1 - P_\varepsilon(A)]P_\varepsilon(B)P_A(1)P_B(0) + [P_\varepsilon(A) + P_\varepsilon(B) - P_\varepsilon(A)\,P_\varepsilon(B)]P_A(1)P_B(1) \quad \textbf{(3)}$$

3.3.2.2. Intrinsic Gate Error Effects

Tab. 2 presents the AND gate output behaviour in the presence of both input and internal errors. We employ the Binary Symmetric Channel (BSC) **[Ryan09]** technique to model the erroneous buffer behaviour.

Table 2 : Faulty AND gate with unreliable inputs

*Actual Value of Z**	*State of Z**	*Gate Fault*	*Value of Z*	*Correct Value of 'Z'*	*State of 'Z'*	*Error_Probability*
0	c	C	0	0	c	
	c	f	1	0	ε	$P[Z^*=0, Z^*_c, P_f]$
	ε	c	0	1	ε	$P[Z^*=0, Z^*_\varepsilon, P_c]$
	ε	f	1	1	c	
1	c	C	1	1	c	
	c	f	0	1	ε	$P[Z^*=1, Z^*_c, P_f]$
	ε	c	1	0	ε	$P[Z^*=1, Z^*_\varepsilon, P_c]$
	ε	f	0	0	c	

The gate output static and error probabilities can be defined as:

$$P_Z(1) = P_{Z^*}(1)(1 - P_F) + P_{Z^*}(0)\,P_F \quad (4)$$

$$P_Z(0) = P_{Z^*}(0)(1 - P_F) + P_{Z^*}(1)\,P_F \quad (5)$$

$$P_Z(Z) = P_F + P_\varepsilon(Z^*) - 2 * P_F * P_\varepsilon(Z^*) \quad (6)$$

3.3.3. Reconvergent Fanout and Error Bounding

Based on the mathematical modelling presented in the previous section, we introduce an algorithm which, given a certain circuit and the error probabilities of the primary inputs and internal gates, can evaluate the error probability of the outputs. As indicated in Alg. 1, on every node, it checks for the state of the two children nodes. In case, if they are inverted, the error due to the inverter is accounted. Using Eq. (2) and (3) the error due to the AND gate is computed both on the internal and external output nodes. This process is continued until the primary outputs are reached. At present, we have not taken the Reconvergent Fanout (RFON) into consideration. A tool, called as Reliability Evaluator, is developed based on Alg. 1. It is integrated into the open source tool 'ABC' to automate the error probability computation.

Algorithm 1: Generic Method for Reliability Evaluation

INPUT: N, total number of nodes in the AIG network;
Error probability of Individual Gates
Switching Activity P_{SA} on the primary input nodes {PI's}
OUTPUT: Output error probability of the circuit
1. **For all** nodes I = 1 to N **do**
2. **if** *left_node* is inverted **then**
3. Use Eq. 3 to compute new static node error probability
4. **end if**
5. **if** *right_node* is inverted

6. Use Eq. 3 to compute new static node error probability
7. **end if**
8. **Compute** Internal node error probability using Eq. 2
9. **Compute** Output node probability using Eq. 3
10. **End for**

To evaluate the tool performance we considered a set of MCNC benchmark circuits. In Tab. 3, Columns 1 and 2 give the name and number of gates in the benchmark circuit. Column 3 captures the accuracy of the method when compared with Monte Carlo (MC) simulations while Column 4 highlights the significant time savings the proposed algorithms achieves when compared with MC simulations. From the Tab. 3, it is clear that the proposed algorithm is accurate enough and can be used to compare two logically equivalent configurations. As the analysis is AIG based, we have not performed any kind of comparative study with any of the previous works **[Han11] [Mohanram09]**. VSIM simulator was used to implement a Monte Carlo framework for reliability analysis under fault injection. The sample size used for reliability analysis was 100k, 50k, and 10k for 0.001, 0.01, 0.05 error scenarios, respectively.

Table 3: MCNC Benchmark Circuits Based Accuracy & Performance Evaluation for different gate errors (ε)

Benchmark	Gate Count		Avg. Error Deviation on all outputs %			Runtime {s}	
	AND	Inverter	ε = 0.001	ε = 0.01	ε = 0.05	MC_Sims	Tool
Cu	55	29	5.75	2.78	9.48	7051.89	0.393
x2	55	32	6.06	7.24	9.24	5356.85	0.924
Parity	45	61	3.51	6.38	7.55	10215.41	1.042
cm150a	61	71	3.36	2.81	9.43	16477.93	1.558
Cordic	82	84	1.57	2.24	9.57	22749.73	0.966
Mux	85	92	2.32	3.13	6.25	22019.47	0.295
b9	104	78	3.79	3.38	6.57	43578.04	0.827
Count	128	130	5.34	7.84	9.84	52890.57	4.691

3.3.3.1. Bounding Node Error Probability

This section presents a method to deal with the RFON issue when computing circuit error probability. Computing the node error probability is more complex in the presence of RFON because Eq. (3) does not hold true. This is because each of the terms in Eq. (3) cannot be factorized due to dependencies between the four probabilities. There is no closed form solution to resolve terms of the type $P[A_x,B_x,A_y,B_y]$, where $x \in \{0,1\}$ and $y \in \{\varepsilon,c\}$ in an exact manner. Iterative approaches could be developed but their complexity would grow exponentially for each of the 6 terms.

To simplify the propagation algorithm, a bounding procedure is presented. Tab. [2] lists all the possible scenarios that would result in an output error. Bounding each of these scenarios singularly results in loose bounds because the error committed in each term is accumulated. In turn, this result in the quick convergence of the upper/lower bound to 1/0, respectively. To avoid this scenario, only the total gate output error probability is bounded. To facilitate the bounding process two cases are considered, a pessimistic and an optimistic scenario. Upper/lower bounding always resulted in over/under estimated error probability. The advantage is in the fact that the formulas for the considered cases offer easy analytic solution.

PESSIMISTIC RULE: A fault occurring on any of the inputs can be propagated onto the output if and only if the second input node is set to '1'. To make the computation more pessimistic, we take the effect of both the input nodes. This is represented by Eq. (7).

$$P_\varepsilon(Z^*) \leq Max\{P_\varepsilon(A)\}P_1(B) + Max\{P_\varepsilon(B)\}P_1(A) \quad (7)$$

OPTIMISTIC RULE: A fault occurring on any of the inputs can be propagated onto the output if and only if the second input node is set to '1'. To make the computation more optimistic, we take the effect of only one node into consideration. This is represented by Eq. (8).

$$P_\varepsilon(Z^*) \geq Max[\{P_\varepsilon(A)P_1(B)\},\{P_\varepsilon(B)P_1(A)\}] + P(A_\varepsilon B_\varepsilon A_1 B_1) \quad (8)$$

Having developed bounds to apply on the nodes that are point of reconvergence, it is now possible to update the algorithm to propagate them:

Algorithm 2: RFON based Reliability Evaluation

Require: N, total number of nodes in the AIG network
1. **Initialize** Static Probability (P_{SP}) on all nodes computed using the generic algorithm
2. **Compute** Static probability (P_{SP}) using Eq.(1-2)
3. **for** I = 1 to N **do**
4. **if** *fanout_of_current_node > 1*
5. Identify all the nodes in the cone
6. **for** J = 1 to *No of nodes in the cone* **do**
7. **if** *left_child* is inverted
8. Use Eq.(3-6) to compute new static & node error probability
9. **end if**
10. **if** *right_child* is inverted
11. Use Eq.(3-6) to compute new static & node error probability
12. **end if**
13. **If** *node_under_consideration converges* **do**
14. Use Eq.(8,9) to compute bounds
15. **end if**
16. Use adaptive based algorithm to compute new static probabilities on each node
17. **End for**
18. **End for**

A methodology is developed which addresses the issue of the reliability computation of any circuit for a give gate error probability. Based on this tool, we have developed a wrapper around the open source tool 'ABC'. This enables the node error propagation on a complex circuit represented in the AIG format. This methodology makes use of the error models developed in WP2. Further, this tool plays a vital role in the reliability aware synthesis framework that we introduce in the next section.

3.4. Reliability Driven Synthesis

Traditional logic synthesis methodologies and tools are centred on fulfilling timing, power, and area constraints or on achieving acceptable trade-offs among those **[Pedram96] [Mehrotra11]**. However, as the CMOS technology entered the nanometer era it cannot cover any longer all the relevant aspects. Nanotechnology specific issues, e.g., VDD reduction, higher impact of process parameter and temperature variations, result of increased device failure rates, making CMOS ICs less reliable **[Borkar05] [Constantinescu03]**. This tendency is not CMOS specific, as even the most promising post silicon devices, e.g., Carbon Nanotube Field-Effect Transistors (CNFETs) that are considered to eventually replace CMOS suffer from various amounts of statistical variation in device behaviour, leading to a lack of reliability. As a result, reliability is turning out to be a major design metric sharing

equal importance with the other existing design metrics and design time reliability assessment and optimization is becoming a mandatory step in the IC design flow. To overcome the reliability related concerns, a variety of techniques have already been proposed. Negative Bias Temperature Instability (NBTI) and its mitigation techniques have received significant focus **[Vrudhula06]**. Dynamic Reliability Management (DRM) techniques, which try to hide the inherent pessimistic reliability while maintaining the system performance and lifetime expectation within the desired range were previously proposed **[Rivers04]**. Another novel idea is the concept of inexact computing **[Palem12]** where circuits are designed with erroneous gates.

Logic synthesis traditionally is classified into two broad categories, local rule-based transformations (or rewriting) and technology independent/dependent algorithms **[Mishchenko 06-b]**. Rewriting is based on employing a set of local transformation techniques on a small sub-section of the graph in order to improve area, power or timing. Algorithmic based approaches work on the observation that there exists certain set of operations, which are inherently good irrespective of technology. Our work explores the possibility of reducing output error probability by employing local transformation techniques. Though reliability driven logic optimization is in its infancy when compared to power and delay driven optimization, the method presented here is still based on the popular and successful concept of local transformations **[Darringer81] [Brayton87].**

The basic idea behind our proposal is to implement reliability aware transformations. We introduce set of local transformation rules for logic optimization from a reliability perspective. The proposed transformation rules (i) maintain the logical equivalence of the new circuit with the original one and (ii) provide a set of standard rules that when applied in a guided fashion would result in improved circuit reliability. We then study the impact of the gate error probability on equivalent logic configurations to determine the best realization. The transformation rules are built upon the application of Boolean algebra logical equivalence laws such as swapping and reduction of variables. We have evaluated our logic transformation rule set on a test circuit and results show a reliability improvement in the order of 10%.

Given a combinatorial circuit implementing the Boolean function f lets call S_I the initial AIG network prior to rule application and S_F the final AIG network after rule application. The aim of the reliability aware synthesis is to find a sequence of transformations leading to an AIG network $S_F=S_{opt}$ that minimise the cost function Err(f). It is well known that logic synthesis based on local transformation is an NP hard problem. Thus, we rely on a heuristic approach to find an acceptable solution, i.e., an AIG providing reliability higher than a certain threshold. The strategy chosen is to apply local optimization based on a transformation rule set. The challenge is in deciding whether to transform a sub-graph to the new state S_0 based on the impact of the transformation on the PO reliability.

3.4.1. Local Transformation Rules

In this section, we present the set of transformation rules utilized in our quest for the reliability optimized implementation of Boolean functions.

Rule 1: The first transformation is modelled based on the Boolean algebra distributive law. The rule decreases the node count by one and also improves the circuit reliability.

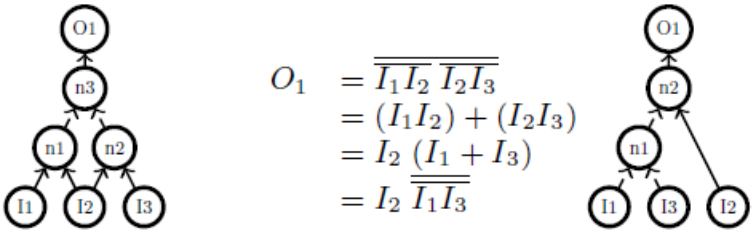


Figure 10: Logic Equivalence Rule 1

Rule 2: This transformation is based on the associativity low. The underlying assumption is that reducing the length of the longest path will improve the reliability of the circuit.

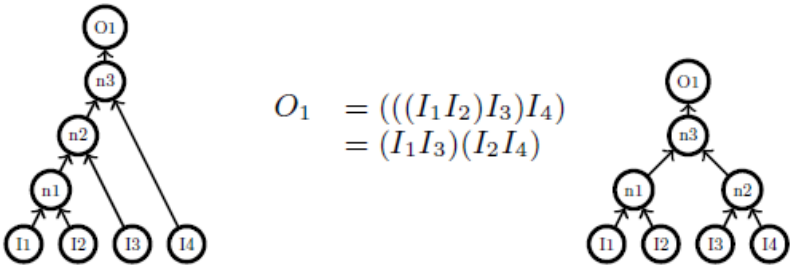


Figure 11: Logical Equivalence Rule 2

Rule 3: This rule is also based on the associativity low. It suggests the fact that inverters distributed equally on both legs will improve the circuit reliability. This rule specifically targets the configuration for a 2-input XOR gate. Its application can be seen predominantly in circuits like priority encoders, Cordic processors, adders, etc.

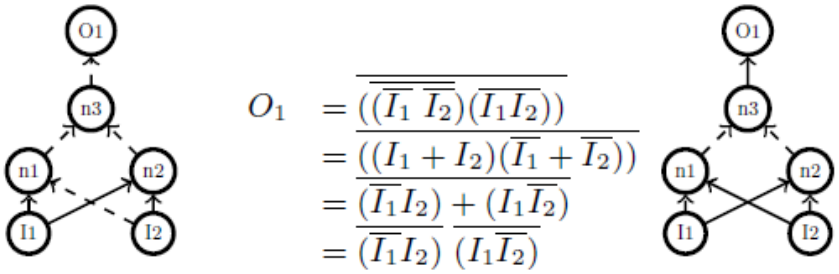


Figure 12: Logical Equivalence Rule 3

Rule 4: This rule is derived based on principles of associativity and insertion and defines the best representation for 3-bit majority voter. Figure 13 depicts the three possible configurations of the majority voter. This rule in principle applies rule1 to reduce node count and then applies rule3 to place the inverters equally on both the legs of the output. The reliability analysis is performed on all the three circuits and the results are plotted in Figure 15(d). An improvement of more than 15% is achieved by this transformation.

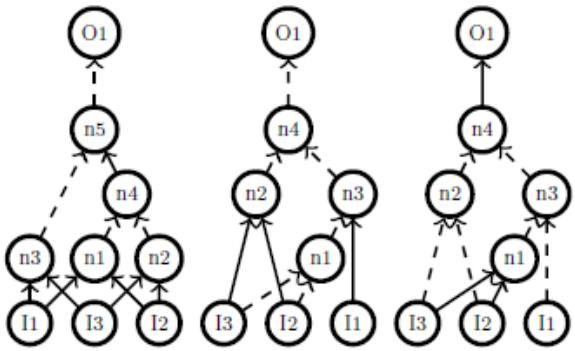


Figure 13: Logical Equivalence Rule 4

Rule 5: The fifth transformation rule is based on the commutative law. The rule states that the signals with the lowest static probability of '1' in an AIG tree should be closer to the output, or closer to the root node of a sub graph. Intuitively, this rule is maximizing the masking effect of the AND gate to minimize the effect of any error coming from the left side of the graph. The reliability improvement is strongly dependent on the static probability of the gate inputs.

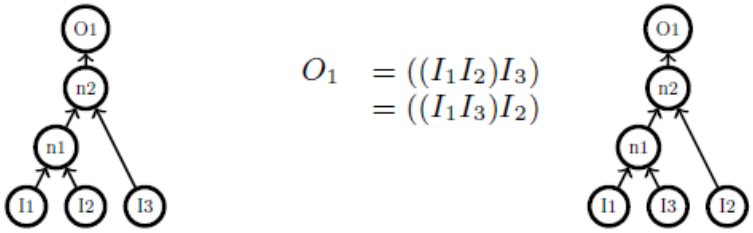


Figure 14: Logical Equivalence Rule 5

3.4.2. Simulation Results

Direct mathematical analysis is not feasible to compute the improvement achieved as the number of variables in the equations used to compute the output error probability are too high. Instead, we set the input node static probability to 0.5 and the input error probability to 0.01 and perform simulation using the tool previously described in section 3.1.3. All the rules have also been investigated for other generic patterns such as Gaussian. Simulation results for different input patterns confirm that rules [1-4] improve the reliability of the circuit and hence suggest that they are applicable in any general scenario. In contrast, rule5 is applicable only under certain circumstances as detailed out later in the section. The improvement achieved by employing the local transformations is plotted in Figure 15.

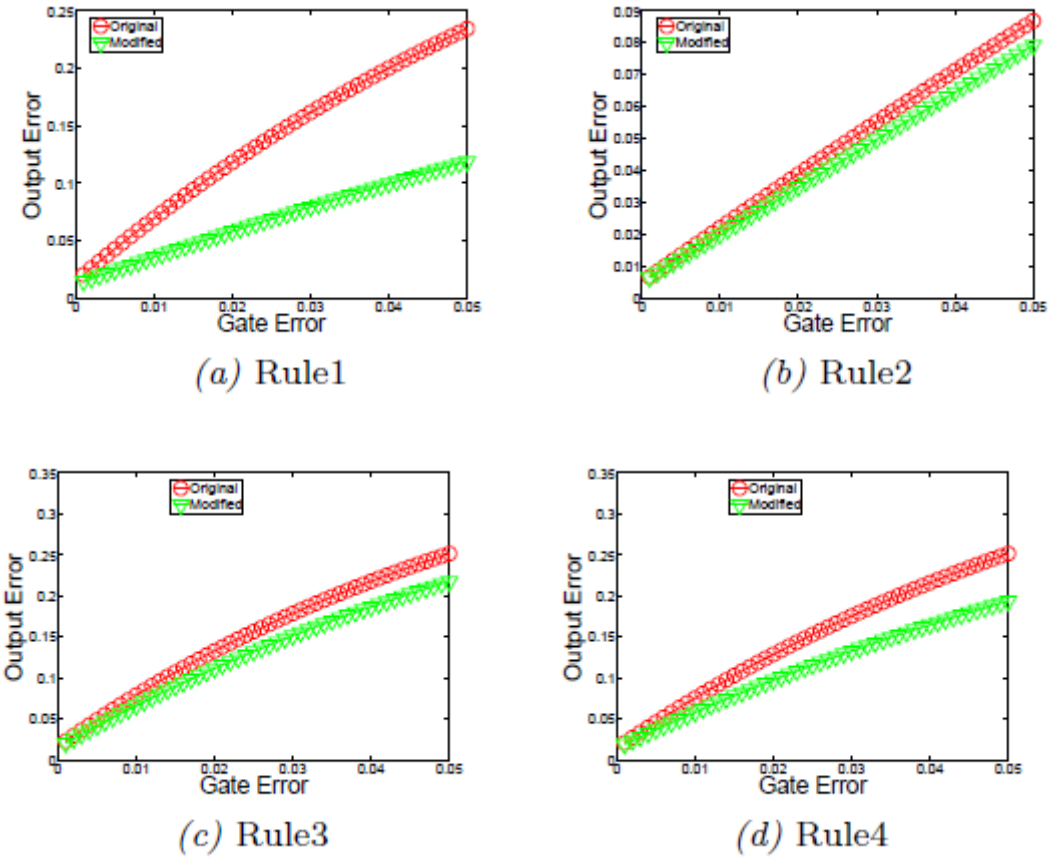


Figure 15: Simulation Results for rules [1-4]

3.4.3. Search Algorithm

The previously presented rules have been tested in various conditions and the simulation results show reliability improvement in every scenario. However, due to the high variable count, it is impossible to derive a rigorous proof and/or to test them for all input scenarios. Hence, one cannot generalize the improvement in all circumstances. As a result, a local optimization search algorithm is used to confirm the reliability improvement before its application on the circuit. Alg. 3 details the process adopted for performing local transformations. Starting from the initial circuit configuration, we traverse through the graph to see if any of the rules are applicable on the given node. For every possible transformation, the new circuit reliability is computed. The configuration that yields the highest circuit reliability improvement is selected and the new topology is generated. This process is continued on every node on the graph until we reach the primary outputs where no more transformations are applicable.

Algorithm 3: Reliability Aware Optimization

Require: N, total number of nodes in the AIG network
Require: RN, total number of transformation rules
1. **For** I= 1 to N **do**
2. **For** J = 1 to RN **do**
3. **if** Rule R_j is applicable on $Node_i$ then
4. Implement the transformation and calculate new reliability W_{ij}
5. **End if**
6. Switch back to normal configuration
7. **End for**
8. Select R_j s.t. $W_j = \min(W_{ij})$

9. **Implement** R_j
10. **End for**

3.4.4. A Case Study

As a case study, the proposed reliability aware synthesis algorithm is applied on the AIG depicted in Figure 16(a). As the circuit is simple, only two local transformations are applicable. Rule2 is applicable on node n1. The new error probability of the circuit after applying this rule is presented in Figure 17. It is clear that applying Rule2 on node n1 results in higher reliability. No other rule is applicable on node n1. So, this would be the new reference topology of the circuit. Also, Rule2 can be applied on node n2 of Figure 16. After this transformation, the reliability of the circuit reduces and hence this transformation is not applicable. Further, rule2 can also be applied on node n3 of Figure 16(c). Simulation results presented in Figure 17 show that such transformation also results in improvement in the reliability of the circuit. No further rules can be applied on any of the nodes and this would remain as the most optimized version of the reference circuit. It is clear that even for such a small circuit, the application of the reliability aware synthesis algorithm can achieve a marked improvement of 25% over the initial configuration and about 10% over the netlist synthesised by the ABC tool.

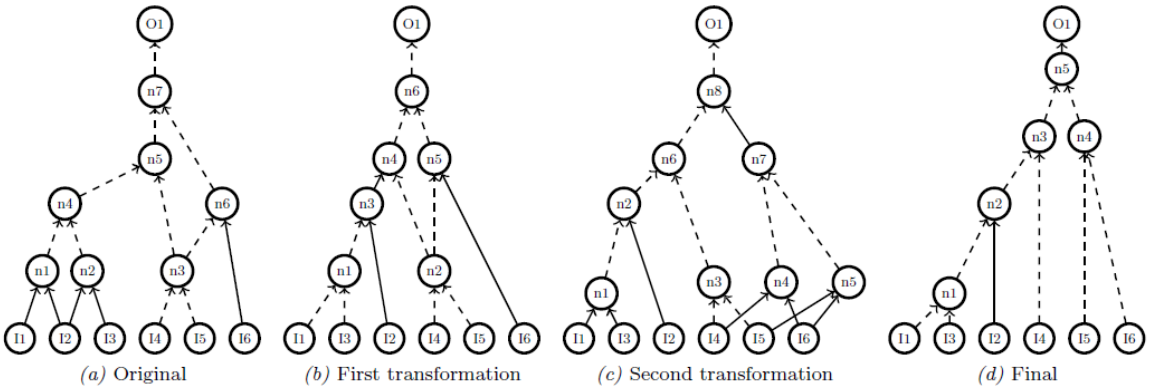


Figure 16: Application of logic transformation rule set

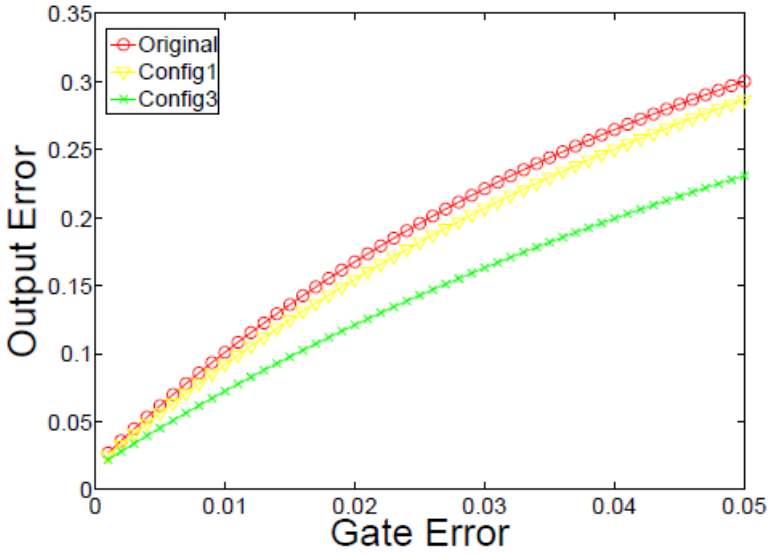


Figure 17: Case Study Simulation Results

4. Probability Density Functions (PDFs) Based IC Reliability Evaluation (Task 5.2 & Task 5.4)

Abstract: In this section we introduce a Probability Density Functions (PDFs) based Integrated Circuit (IC) reliability assessment framework. Since the efficiency of reliability driven design-time optimizations and of run-time management frameworks directly depend on the reliability evaluation accuracy, instead of relying on a single probabilistic value to reflect the reliability status of a circuit, we propose to employ a distribution of probabilities for a closer adherence to a faulty circuit stochastic behavior. To the extent of our knowledge, the proposed approach describes the first attempt to assess the reliability of a circuit based on PDFs. The proposed framework is based on a variational inference method, and exploits the geometry of the statistical manifold to yield a fast and scalable reliability assessment approach, which can be potentially integrated in reliability aware synthesis tools and Dynamic Reliability Management frameworks.

Publications: Unpublished Work (a Journal paper is in preparation).

Traditionally, reliability begins to be evaluated usually late in the life cycle of an Integrated Circuit (IC) **[Pecht09]**, i.e., shortly before manufacturing release, via well-defined qualification tests, e.g., compliance tests or zero-failure tests **[Crowe98]**. As a result, the preponderant percentage of all reliability and qualification IC life cycle costs is attributed to correcting the IC design inadequacies (intent misalignment between design and reality) and defects after their occurrence, and not to making the IC right to begin with **[Cranwell07]**. However, for the current technology, neglecting the reliability concerns during the initial IC life cycle phases, is no longer a feasible approach for a highly competitive semiconductor industry which emphasizes short time-to-market, reduced Non-Recurring Engineering (NRE) costs associated with mask spins, first-pass success and long-term reliability goals, e.g., extended useful lifetime. In the realm of variability and higher failure probabilities expected for the emerging nano-devices and their afferent interconnects, reliability should be addressed upstream of full scale IC life cycle phases, from the early design inception phase to the in-field operation phase. Therefore, fault and defect tolerant techniques those enable an IC to recover from manufacturing and operational errors, have to be considered up-front, at design-time (pre-Si).

Building architectures from prohibitively unreliable emerging nano-devices, expected to exhibit increased susceptibility to variations, e.g., manufacturing, permanent and transient failures, suggest including reliability as an optimization goal (besides power, area, and time) in the forthcoming EDA tools. Considering this context, an accurate yet fast reliability analysis is needed at design time, to allow a gate-level comparison of different logic circuits architectures and enable a reliability driven synthesis process. Moreover, such an evaluation method can be embedded in a Dynamic Reliability Management (DRM) framework **[Wang13]** to allow for fast and accurate reliability evaluation based on aging sensor collected data.

4.1. Previous Work

Various probabilistic analytical approaches to evaluate the circuit reliability have been proposed. The **Probability Element (ProxEl)** method was introduced in **[Horton02]** to alleviate the typical problems encountered by Monte Carlo simulation (i.e., finding good quality pseudo-random number

generators) and partial differential equations (i.e., more difficult to set up and solve). Each probability element carries enough information to probabilistically determine the future behavior of the model and generate all the succeeding probability elements **[Molnar05]**. The method describes every probabilistic configuration of the model in a minimal manner, and it is based on a discrete-time Markov chain as underlying stochastic process, which is constructed by on-the-fly inspection of all possible probabilistic behaviors.

In **[Patel03]**, the authors introduced the **Probabilistic Transfer Matrices (PTM)** formalism, whose underlying principle dates back to year 1964 **[Levin64]**. The PTM method relies on an exhaustive listing of the gate inputs/outputs, allowing simultaneous and exact reliability evaluation over all possible input combinations. It employs a matrix representation of gate errors. For instance, the PTM afferent to a NAND2 gate is given by:

$$PTM_{NAND} = \begin{bmatrix} \varepsilon & 1-\varepsilon \\ \varepsilon & 1-\varepsilon \\ \varepsilon & 1-\varepsilon \\ 1-\varepsilon & \varepsilon \end{bmatrix} \quad (9)$$

where ϵ is the gate failure probability.
The gate output probability is obtained as:

$$[P_0 \ P_1] = [P_{00}\, P_{01}\, P_{10}\, P_{11}] * PTM_{NAND} \quad (10)$$

with P0 and P1 the NAND2 output probabilities of obtaining a logic '0' and '1', respectively, and P00 , P01 , P10 , and P11 the gate input probabilities of logic '00', '01', '10', and '11', respectively. A circuit PTM is based on logic dependent composition of individual gates or sub-circuits PTMs **[Krishnaswamy05]**. The PTM-based approach is applicable to any type of logic gate and thus allows different probabilities of failure for different logic gates. While allowing for the derivation of the exact circuit probability of failure, the method involves massive storage (i.e., $O(2^{m+n})$, for a circuit with *m* primary inputs and *n* primary outputs) - albeit matrices compression - and a big overhead associated with handling the stored data. Furthermore, as the circuit size increases, the circuit modularization process, which is not automated, may become intricate and error prone. Thus, the high computational complexity in terms of runtime and memory usage renders the PTM-based approach useful for reliability estimation of small circuits only.

Another analytical reliability estimation approach relies on the **Probabilistic Gate Model (PGM) [Han05], [Taylor06], [Han11]**. The method entails expressing the probability of each logic gate as follows:

$$P_{GATE_{out}} = \begin{bmatrix} P_{GATE_{in}} & 1-P_{GATE_{in}} \end{bmatrix} . \begin{bmatrix} \varepsilon \\ 1-\varepsilon \end{bmatrix} \quad (11)$$

where ϵ is the gate failure probability, P_{GATEin} is the probability of the fault-free gate to generate a logic '1' at its output, and $P_{GATEout}$ is the probability of the correct gate output. The circuit is modularized (with logic gates as the indivisible units) and its overall reliability is derived via an iterative procedure by multiplying the individual gate output reliability figures. While being applicable to potentially model any type of gate and failure, the method assumes that the gate input/output signals are statistically independent, which leads to approximate reliability results. In **[Han11]**, the correlations in the input signals or caused by reconvergent fan-outs are addressed by sequentially decomposing and treating each fan-out in a recursive manner, at the expense of

increasing the computational time exponentially with the number of reconvergent fan-outs. While exhibiting similar reliability estimates but a lower computational and time complexity (i.e., O (*n*) and O ($n \cdot 2^m$) respectively, for a circuit with *m* gates and *n* primary inputs) when compared to the PTM approach, the PGM-based model is still not scalable to large circuits.

In **[Choudhury09]**, the authors propose three scalable algorithms for reliability assessment. Particularly, the **single-pass reliability analysis** algorithm is able to: (i) accurately evaluate the reliability of circuits without convergent fan-out and (ii) approximately evaluate the reliability of circuits exhibiting spatial correlations, by computing pairwise correlation coefficients of dependent signals. The algorithm is based on expressing the error at a gate output as the cumulative effect of the intrinsic, local gate error component and an error component attributed to the failures of the gates in its fan-in cone. The gates are topologically sorted and the circuit output reliability is derived in a single pass from the primary inputs to the primary outputs. The algorithm is fast and requires a smaller memory footprint. However, the reliability estimates may be less accurate since the circuit reliability is assessed based on a single probabilistic value. The single pass method is extended in **[Mahdavi09]** to multiple passes for reliability evaluation of sequential circuits.

The **Signal Probability Reliability Analysis (SPRA)** method was proposed in **[Franco08-a]** and **[Franco08-b]** embeds the cumulative effect of multiple, simultaneous errors in a circuit, in the form of a bit-flip error at the output of a faulty gate. The probability of a signal is represented as a matrix consisting of four possible states that a signal probability can attain, i.e., a correct logic '0', a correct logic '1', a faulty logic '0', and a faulty logic '1' **[Ercolani89]**:

$$P_{signal} = \begin{bmatrix} P_{signal=correct0} & P_{signal=faulty1} \\ P_{signal=faulty0} & P_{signal=correct0} \end{bmatrix} \quad (12)$$

The output probability of a gate is obtained by multiplying the gate transfer function (expressed using PTM **[Patel03]**) with the Kronecker product of the gate input signals probabilities, as follows:

$$[P_0 \;\; P_1] = \left(P_{GATE_{in1}} \otimes P_{GATE_{in2}}\right) X \; PTM_{NAND} \quad (13)$$

The SPRA-based method yields exact results for circuits without reconvergent fan-outs, but approximate reliability estimates when dealing with the exponential complexity of signals probability correlations. Since SPRA stores the information related to each signal independently, the memory footprint is smaller when compared to the PTM approach which stores the probability information of the entire circuit in one PTM. Another advantage resides in the fact that only one matrix multiplication is required to compute the reliability of a gate, which makes the SPRA-based approach fast and scalable.

Probabilistic graphical models as an approach for logic circuits reliability assessment, benefit of flexibility and power of representation as well as increased ability to effectively learn and perform inference in large networks as is the case of large logic circuits **[Koller09]**. They provide a principled approach to deal with uncertainty through the use of the probability theory and an effective manner to cope with the complexity associated with the circuit correlations through the use of graph theory. A graphical model can be regarded as consisting of a collection of probability distributions which factorize according to the underlying structure of the graph. The two most common types of probabilistic graphical models are Bayesian Networks and Markov random fields.

Recently, **Bayesian Networks (BNs)** have been applied in the context of circuit reliability evaluation **[Rejimon05], [Rejimon05], [Ibrahim11]**. BNs, whose underlying semantics are based on directed graphs, allows one to capture both the temporal and spatial circuit dependencies in a comprehensive

manner, providing an exact and minimal probabilistic model for reasoning and inference in causal logic networks. BNs encode the joint probability distribution over a set of variables and decompose them into a product of the conditional probability distributions over each variable given, its parent nodes in the graph. The conditional probability of an output signal given the input signals probabilities reflect the propagation of errors through the circuit. Thus, given the circuit inputs probabilities of failure, it is possible to predict the output probability of failure. Furthermore, due to the conditional probability symmetric nature, BNs can be employed not only for prediction (i.e., infer the effects of known causes), but also for diagnosis (i.e., infer the potential causes of known effects). While the probability of failure is exact as in the PTM-based approach, the BNs-based approach suffers from massive storage requirements associated with the underlying large conditional probability tables, and thus manipulating BNs for large circuits is potentially intractable. Improvement was reported in **[Ibrahim11]**, by taking into account the dependence of the probability of failure of different gates on each gate internal structure (i.e., the gate probability of failure is obtained from the probabilities of failure of the individual transistors encompassed by the gate).

The **Markov Random Fields (MRFs)** based reliability evaluation approach presented in **[Bahar03]**, employs the Gibbs distribution to characterize the reliability in terms of entropy and the noise in terms of thermal energy. While being suitable for reliability assessment of small circuits or of regular redundant architectures, such as NAND multiplexing and triple modular redundancy **[Bhaduri05]**, for arbitrary multilevel logic circuits the MRF-based approach becomes computationally intensive as a consequence of the involved minimization of a Gibbs distribution function with a large number of variables. Furthermore, MRFs model dependencies via undirected graph structures, which might be less suitable for the causal computing, characterized by a flow of information from the input to the output, and widely present in traditional and future nano-computing platforms. Specifically, MRFs cannot capture the induced and non-transitive dependencies, which are two important dependencies exhibited by causal networks **[Pearl88]**. Another aspect is that errors are indirectly modeled through the polynomial coefficients of the energy potential expressions, which for nano-CMOS circuits are complex functions of the underlying errors causes and hence less amenable to a direct specification (e.g., uniform faults across logic gates will not correspond to uniform errors over the coefficients).

The previous discussion, and given the i-RISC ambitions on reliability aware design and logic synthesis, clearly indicates that state of the art solutions do not fulfill our requirements and constraints. In view of that we introduce in the following a highly accurate reliability evaluation framework, which do to its small amount of required resources can be integrated in the i-RISC reliability aware synthesis tool chain and in Dynamic Reliability Management (DRM) frameworks.

4.2. Proposed Approach

Hitherto reliability evaluation approaches customary posit a single value for a gate probability of failure. While benefiting from a relative simplicity of implementation, the single-probability approach may not suffice for accurate reliability assessment. A more appropriate approach to model the faulty circuit's stochastic behavior would be to consider a range of failure probabilities of a gate output, for given gate errors. Such a Probability Density Function (PDF) of a gate output can be constructed based on a large sample of different fault patterns, and environmental aggression profiles.

Another aspect is related to the fact that the majority of previous approaches evaluate the reliability of a circuit starting from the gate level and furthermore, most of them rely on the assumption that all

gates have the same reliability. However, the reliability accurate computation at gate-level is of foremost importance, since very small reliability estimation errors at the gate-level can severely impact the reliability evaluation of circuits comprising large numbers of gates **[Roelke07], [Han02]**. For instance for VLSI circuits, a very small estimation error at the gate-level conducts to circuit reliability estimation errors of a few orders of magnitude, e.g., for a circuit composed of N = 10^{10} gates, and using the most simplistic model to derive the circuit reliability:

$$P_{failCIRCUIT} = 1 - \left(1 - P_{failGATE}\right)^{N} \quad (14)$$

a gate-level reliability estimation error in the order of 10^{-10}, translates into an estimation error of the circuit reliability in the order of 10^{-2} **[Ibrahim11]**).

Thus, the above considerations motivate us to develop a reliability assessment framework, which aims to provide probabilistic inference that is accurate enough and suitable for fast and large-scale circuit settings.

4.2.1. Model Definition

The probabilistic query that we would like to solve, can be stated as follows:

Given, (i) a circuit with known topology and possibly layout, its workload (known primary inputs vectors and their associated probabilities/PDFs), and (ii) an aggression profile (e.g., T, VDD, fault scenarios - fault types and their expected probabilities), determine the PDFs of obtaining the correct circuit primary outputs.

Assume we have a probabilistic model of the form

$$p(E, X, Y) = p(X|Y)\, p(X|E)\, p(E) \quad (15)$$

with the following sets of nodes:

- ***E***, the set of evidence nodes, which are associated to the known PDFs of the circuit primary inputs;
- ***Y***, the set of latent (hidden) nodes, which are associated to the PDFs of the circuit primary outputs. These are the PDFs that we would like to infer.
- ***X***, the latent (hidden) intermediary nodes, which correspond to the remaining gates and wires of the circuit.

We shall assume throughout the remainder of the section that we are concerned with exponential family's representations. The restriction to the exponential family of probability distributions is a mild and reasonable assumption, since most of the important distributions, e.g., Gaussian, multinomial, exponential, Dirichelet, Poisson, Gamma, belong to the exponential family. We consider a conjugate prior in the exponential family, which implies that the posterior and the prior distributions have the same functional form. The convexity properties, and in particular the associated conjugate dual relation, of the exponential family have immediate inferential implications, with desirable algorithmic consequences.

The exponential family of distributions **[Wainwright08]** over a random variable x, parameterized by given θ, is expressed as:

$$p(x, \theta) = \exp\{\left(\theta, \phi(x)\right) - \psi(\theta)\} \quad (16)$$

taken w.r.t. an underlying measure ν on the space {x}.

Here, $\psi(\theta)$ is the log-normalizer which ensures that $\int p(x,\theta)v(dx) = 1$, and is a convex function defined by the integral:

$$\psi(\theta) = \log \int \exp\{\theta, \phi(x)v(dx)\} \quad (17)$$

The vector functions θ and $\phi(.)$ are the natural (canonical) parameter and sufficient statistics, respectively.

4.2.2. Variational Inference – General Framework

Standard approaches to inference over the probability simplex include variational inference **[Beal03]**, **[Wainwright08]**, and Markov Chain Monte Carlo methods, such as Gibbs sampling **[Gilks96]**. In the proposed framework, we concern ourselves with variational inference since it serves better for developing fast, potentially on-line algorithms in large-scale settings.

The general idea of variational inference is to recast the probabilistic inference problem into an optimization-based formulation, and express the posterior distribution of interest as the optimization problem solution. In order to provide a means of approximating the posterior distribution of interest, the optimization problem is relaxed. Such relaxations can be carried out in various ways, either by approximating the function to be optimized, or by approximating the set over which the optimization is performed. One common approach is to approximate the posterior distribution of interest (i.e., the true posterior) with a family of distributions from which the distribution that is closest to the true posterior is sought. The measure of closeness between two distributions is commonly the Kullback-Leibler (KL) divergence. The optimization of the KL divergence is then usually cast as an optimization of a lower bound on the logarithm of the likelihood (i.e., the marginal probability of the observations). The resulting optimized distribution constitutes the approximated posterior of interest. Figure 18 illustrates coarsely the variation inference principle.

In the following, we shall employ the following notations:

- $E_{g(x)}[f(x)] = \int g(x)f(x)dx$ denotes the expectation of distribution f under distribution g;
- $KL\,|f(x)||g(x)| = -\int f(x)\log\frac{g(x)}{f(x)}dx$ denotes the KL divergence between distributions f and g.

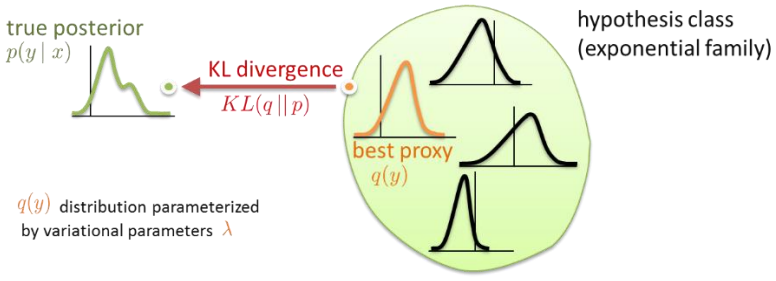


Figure 18: Schematic representation of the variation inference concept.

4.2.3. Evidence Lower Bound

In variational inference, we would like to minimize the KL divergence from the approximated distribution (i.e., the variational distribution over the latent variables/parameters X and Y), q(X, Y) and the true posterior, p(X, Y |E). To this end, we derive a lower bound on the logarithm of the marginal likelihood (i.e., the model evidence E), log p(E), by employing Jensen's inequality ($\mathbb{E}_g|f(a)| \geq \mathbb{E}_g[\log f(a)]$, ∀a random variable, f and g distributions), as follows:

$$\log p(E) = \log \int p(E,X,Y)\, dXdY$$
$$= log \int p(E,X,Y) \frac{q(X,Y)}{q(X,Y)} dXdY$$
$$\geq \int q(X,Y)\, log \frac{p(E,X,Y)}{q(X,Y)} dXdY$$
$$= \mathbb{E}_{q(X,Y)}[\log p(E,X,Y)] - \mathbb{E}_{q(X,Y)}[log\, q(X,Y)]$$
$$\triangleq \mathcal{L}_{XY}(q) \qquad (18)$$

Since

$$\log p(E) = \mathcal{L}_{XY}(q) + KL\,|q(X,Y)||p(X,Y|E)| \qquad (19)$$

minimizing the KL divergence (which is < 0) is equivalent to maximizing the lower bound $\mathcal{L}_{XY}(q)$, as illustrated in Figure 19.

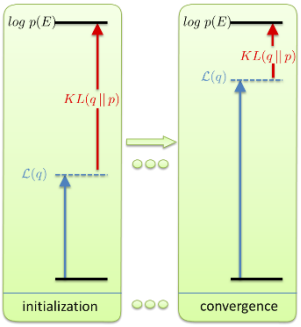


Figure 19: The KL divergence optimization.

In the most straightforward variational inference (Mean Field (MF), also known as Variational Bayes (VB)) framework, the family of approximate distributions are restricted to a tractable family, by positing that the variational distribution q factorizes over the latent variables, e.g., q(X, Y) = q(X) q(Y) (i.e., X and Y are conditionally independent), where each factor of q has a free functional form. The objective is to determine the variational distribution, which maximizes the evidence lower bound $\mathcal{L}_{XY}(q)$. The obtained optimized distribution q* constitutes the approximation of the true posterior distribution p over the latent variables/parameters. Such factorization enables for instance, the analytical optimization to be performed iteratively, with respect to each factor of q, while holding the remaining factors of q fixed, in an analogous manner to the Expectation-Maximization (EM) algorithm (i.e., coordinate gradient ascent on the lower bound). Specifically, in the E-step, q(X) is assumed fixed and the posterior over Y is updated by setting $\partial\mathcal{L}(q)/\partial q(Y) = 0$ which results in the optimal distribution given by:

$$q^*(Y) \propto e^{E_{q(x)}[p(E,Y|X)]} \qquad (20)$$

In a similar manner, in the M-step, computing $\partial\mathcal{L}(q)/\partial q(X) = 0$ for $q(Y)$ fixed (from the E-step), results in the following expression for the optimal posterior over variables X:

$$q^*(Y) \propto p(Y)\, e^{E_{q(y)}[p(E,Y|X)]} \qquad (21)$$

While, the variational EM approach constitutes a convenient solution to the intractability associated with the high dimensional integrals, its convergence speed can be very slow, prohibiting its utilization in large scale settings. Furthermore, the MF strong independence assumptions between the latent variables may place unrealistic or questionable factorizations, disregarding important correlations among the graph nodes and as a consequence resulting in significant bias of the posterior distribution estimate.

Rather than assuming the q factorization over the latent variables, an approach for a closer adherence to reality would be to marginalize analytically (i.e., to integrate out) a subset of the latent variables, and thus to perform the optimization of the evidence lower bound only with respect to the remaining latent variables. Besides the improvement in convergence speed, and hence its feasibility in large scale settings as required in our case, this approach also achieves a more accurate estimate of the true posterior distribution with a tighter lower bound. In **[Teh07-a]**, the authors applied the collapsed variational approach for the latent Dirichelet allocation problem and in **[Teh07-b]** for the hierarchical Dirichelet process. In **[Sung08]**, the authors studied the collapsed approach in the general context of conjugate-exponential family. In **[King06]**, the authors developed the collapsed inference in the context of Gaussian processes, for which the proposed KL-corrected bound - a lower bound on the model evidence, which is also an upper bound on the original variational bound - is shown to radically improve the convergence speed. The same KL-corrected bound was studied in **[Gredilla11]** for Gaussian processes under input independent noise profiles, and referred to as the marginalized variational bound. In our framework, we shall follow the collapsed approach, and adapt the KL-corrected evidence lower bound for the current required context.

Integrating analytically the latent variables Y, a lower bound on the model evidence with respect to the variables X can be derived by applying the Jensen inequality, in a similar manner to the derivation in (18):

$$\begin{aligned}\log p(E) &= \log \int q(X) \log \frac{\int p(E,X,Y)dY}{q(X)} dX \\ &= log \int q(X) log \frac{\int p(E,X|Y)p(Y)dY}{q(X)} dX \\ &= log \int q(X) log \frac{\mathbb{E}_{p(Y)}|p(E,X|Y)}{q(X)} dX \\ &= \mathbb{E}_{q(x)}\left[\log E_{p(Y)}[p(E,X|Y)]\right] - \mathbb{E}_{q(X)}[\log q(X)] \\ &\triangleq \mathcal{L}_{q(X)} \end{aligned} \quad (22)$$

For tractability reasons, a first-order (linear) approximation [31] of $\mathbb{E}_{q(Y)}[g(X)]$ with $g\left(\mathbb{E}_{q(Y)}[X]\right)$ is applied to (22), yielding the following expression for the bound $\mathcal{L}_{x(q)}$:

$$\mathcal{L}_{q(X)} = log \mathbb{E}_{p(Y)}\left[e^{\mathbb{E}_{q(X)}\left[\log \frac{p(E,X|Y)}{q(X)}\right]}\right] \quad (23)$$

Using (20), the expression of the optimal approximated posterior is then given by:

$$\begin{aligned} q^*(Y) &= \frac{e^{\int q(X) log \frac{p(E,X|Y)p(Y)}{q(X)} dX}}{C} \\ &= \frac{p(Y)e^{\mathbb{E}_{q(X)}\left[log \frac{p(E,X|Y)}{q(X)}\right]}}{C} \end{aligned} \quad (24)$$

where C is the normalization constant, and is obtained by integrating the numerator in Eq. (24) with respect to Y.

We index the distribution q(X) by a set of variational parameters θ, and seek the configuration of θ which optimizes the lower bound $\mathcal{L}_X(q)$, rendering therefore q* which is the closest to the true posterior p. The algorithm convergence can be monitored by evaluating if the difference between the previous lower bound (for the previous θ) and its current value (for the current update of θ) is sufficiently small. The optimal distribution q* for the parameters θ at bound convergence is given by (9), with p(Y), the prior distribution of Y. We note that, in the variational EM method, the lower

bound $\mathcal{L}_{XY}(q)$, depends on two sets of variables, whose updates in the E-step and M-step are interlocked (the bound optimization with respect to each set is performed while holding the other set fixed). In our case, the lower bound $\mathcal{L}_X(q)$, is expressed only as a function of one set (the variational parameters) and as such the lower bound optimization is performed only with respect to one set of variables. However, the lower bound dependence on the other set q(Y) is accounted for and absorbed when the optimization w.r.t. q(X) (specifically w.r.t. θ) is performed, q(Y) being expressed in terms of q(X) and its prior as given by (9).

One may further note that the only assumed factorization in this approach is among the output latent nodes Y, conditioned by the evidence nodes E and the approximated nodes X. This can be easily determined using an independence criterion. The d-separation topological criterion **[Geiger90] [Verma98]** determines whether a set of nodes A is conditionally independent of another set of nodes B given a set of evidence nodes V. In particular the set of nodes A is d-separated from the set of nodes B by the set V if and only if every undirected path from a node in A to a node in B is *blocked* by V, i.e., at least one of the following three axioms - depicted in Figure 20 – holds true: (i) every undirected path contains a sequential node in V $(\longrightarrow V_i \longrightarrow)$, (ii) every undirected path contains a divergent node in V $(\longleftarrow V_j \longrightarrow)$, and (iii) every undirected path contains a convergent node $(\longrightarrow T_k \longleftarrow)$ such that neither the convergent node, nor any of its descendants are in V.

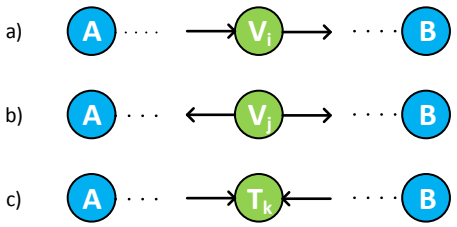


Figure 20: The configurations for the d-separation criteria.

One can observer that for circuit specific scenarios, as is our case, the independence among the output nodes conditioned by the rest of the circuit nodes (input nodes - E, and the inferred nodes X) is always satisfied, since case a) from Figure 20 always holds true.

4.2.4. The Lower Bound $\mathcal{L}_X(q)$ Gradient

Subsequently we concern ourselves with finding the optimal configuration of the variational parameters θ, which optimizes the objective function $\mathcal{L}_X(q)$. In **[Sato01]**, the authors proved that the coordinate ascent algorithm is equivalent to the natural gradient method. Therefore, an update via taking a step in the steepest direction in the space of variational parameters θ, using a Riemannian metric (e.g., the natural gradient **[Amari98]** which accounts for the space information geometry), is equivalent to performing a coordinate ascent update (equivalent to the E- step of the variational EM approach). To this end, we evaluate the gradient of the lower bound with respect to the variational parameters, which gives us the direction of the coordinate ascent update for the variational parameters θ.

The space of all probability distributions $S = \{q(X|\theta)\}$ is not Euclidian with an orthonormal coordinate system θ, but a curved space, namely a Riemannian manifold. In such spaces, the shortest distance between two points does not correspond anymore to an Euclidian line, but to a geodesic (i.e., a curve) following the space curvature. The immediate consequence is that the steepest descent direction along a manifold path (as given by the iterative updates of the parameters θ) is different than the steepest descent direction in the classical Euclidian parameter space. In the case of

statistical manifolds, the commonly employed Riemannian metric is the Fisher information matrix [Amari85] [Amari00]. The rationale behind the choice of the Fisher information matrix as Riemannian metric tensor, can be under- stood from a statistical perspective, as follows: The natural gradient corresponds to the direction which maximizes the objective function $\mathcal{L}_{x(q)}$, such that the KL divergence $KL[q(X|\theta)||q(X|\theta+\delta\theta)]$ is not changed through the optimization (otherwise stated, the natural gradient gives the direction of the highest increase in the objective function, for the smallest change in the KL divergence):

$$\tilde{\nabla}\mathcal{L}_X(q(\theta)) = \underset{\delta\theta}{\operatorname{argmax}}\ \mathcal{L}_X(q(X|\theta+\delta\theta))$$
$$s.t. \quad KL\left[q(X|\theta)\,||\,q(X|\theta+\delta\theta)\right] \leq \epsilon. \tag{25}$$

It follows then from (10), the expression of the natural gradient:

$$\nabla\mathcal{L}_X\big(q(X|\theta)\big) = G^{-1}(\theta).\nabla\,\mathcal{L}_X\big(q(X|\theta)\big) \tag{26}$$

where G is the Fisher information matrix (i.e., the Hessian of the KL divergence) with

$$g_{ij}(\theta) = \mathbb{E}_q\left[\nabla_{\theta_i} logq(X|\theta)\,.\nabla_{\theta_j}^T logq(X|\theta)\right]$$
$$= \mathbb{E}_q[-\nabla^2 logq(X|\theta)] \tag{27}$$

A chief advantage is the KL-invariance with respect to the re-parameterization of the family of variational distributions $q(X|\theta)$, i.e., the update direction depends only on $q(X|\theta)$, and not on a particular transformation of the θ parameters; the followed optimization trajectory in the parameters space is the same, regardless of the re-parameterization of θ. As a result, as opposed to the vanilla gradient, the natural gradient exhibits fast isotropic convergence properties. Additionally, it circumvents the slow or early convergence proneness of the vanilla gradient, avoiding over-aggressive steps on ridges and too small steps on plateaus, and hence being able to cope in an efficient manner with ill-shaped $\mathcal{L}_X(q)$. Furthermore, when the Riemannian manifold is an exponential family of distributions (with ψ a convex function), (27) reduces to:

$$g_{ij}(\theta) = \nabla^2\psi(\theta) \tag{28}$$

For the exponential family, the Riemannian manifold has a dually flat structure, induced by the convex function $\psi(\theta)$. There are two systems of coordinates, {θ} and {η} - the latter also referred to as the expectation parameters, $\eta = \nabla\psi(\theta))$ - and the Riemannian metric is defined by the mutually inverse metric tensors, G(θ) and G(η). The immediate implication is that the natural gradient w.r.t. the expectation parameters η is given by the vanilla gradient w.r.t. the canonical parameters θ and vice versa:

$$\nabla_\theta\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\eta} \quad and \quad \nabla_\eta\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\theta} \tag{29}$$

avoiding therefore the expensive (prohibitively for high dimensionality variational parameters as in our setting) computation of the matrix inverse $G^{-1}(\theta)$.

4.2.5. The Optimization Algorithm

In the following, we denote by G_k the expression $\tilde{\nabla}\mathcal{L}_X(q(\theta_k))$. As concerns the nonlinear optimization techniques, we opt to employ the nonlinear Conjugate Gradient (CG) method, due to its algorithmic simplicity, supralinear (at least quadratic) convergence, and suitability for large scale optimization scenarios. In this method, one determines first the search direction, H_k, then computes a step size, α

(as a result of a line search, or set adaptively), and finally one updates the parameters in the search direction using $\theta_k = \theta_{k-1} + \propto H_k$.

In the flat Euclidean space, evaluating the search direction amounts to computing:

$$H_k = -G_k + \gamma_k \propto H_{k-1} \qquad (30)$$

where $\gamma_0 = 0$ and γ_k is suitably defined, with commonly employed variants such as Fletcher-Reeves (FR), Polak-Ribiere (PR), and Hestenes-Stiefel (HS) **[Nocedal06]**:

$$\gamma_k^{FR} = \frac{\nabla_k^T \nabla_k}{\nabla_{k-1}^T \nabla_{k-1}} \qquad (31)$$

$$\gamma_k^{PR} = \frac{\nabla_k^T (\nabla_k - \nabla_{k-1})}{(\nabla_{k-1}^T \nabla_{k-1})} \qquad (32)$$

$$\gamma_k^{HS} = \frac{\nabla_k^T (\nabla_k - \nabla_{k-1})}{(\nabla_k - \nabla_{k-1})^T \gamma_{k-1}} \qquad (33)$$

The majority of previous gradient-based statistical inference approaches employ the flat space approximation of the conjugate gradient **[Kuusela09] [Honkela10] [Hoffman13]**, based on the rationale that the minimization of functions on a Riemannian manifold is locally equivalent to the minimization on an Euclidian space (since every Riemannian manifold can be isometrically embedded in an Euclidean space). However, as the statistical space is a curved manifold, most of the Euclidean space operations become undefined. For instance, the minimization of $\mathcal{L}$ is not performed any longer along straight lines but along geodesics. Another example is the additive rule employed for updating the search direction, which makes no sense in the Riemannian manifold - it has no geometrical meaning, and as a consequence, the resulting sequence of parameters updates can significantly deviate from the true geodesic whose end-point is the optimal solution. Since H_{k-1} does not reside in the same tangent space as G_k, it follows that it needs to be transported to the tangent space of x_k in order to enable the addition of the two vectors.

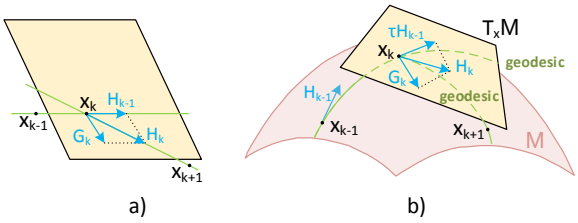


Figure 21: Conjugate Gradient in: a) flat (Euclidean) space, and in b) curved (Riemannian) space.

Figure 21(a) illustrates the CG algorithm in Euclidean space, and Figure 21(b) depicts its counterpart in Riemannian space. When compared to the flat (Euclidean) space approximation, the Riemannian conjugate gradient provides significant advantages in terms of convergence speed and solution accuracy.

Subsequently, after conceptually introducing the basic notions of geometry on a manifold **[Amari85] [Amari00] [Absil08]**, we shall address the adaptation of the Euclidean conjugate gradient optimization method on the Riemannian space. For notation brevity, in what follows we shall denote x_k as variational parameters θ_k.

Tangent space and geodesics

A tangent space $T_x\mathcal{M}$, can be defined as the space composed out of all vectors tangent to the manifold curves which pass through a point x on the manifold $\mathcal{M}$. A graphical illustration of such a tangent space is presented in Figure 22.

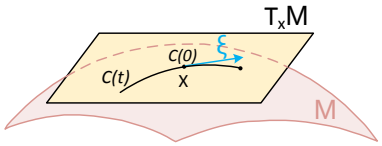


Figure 22: A tangent space Tx $\mathcal{M}$ and a tangent vector ξ, given the point x = C (0) on the manifold curve C (t).

A geodesic between two manifold points is the shortest curve on the manifold which connects the two points. Its velocity is constant (i.e., the tangent vectors along the geodesic are parallel) and its acceleration is zero. Geodesics on manifold can be seen as a generalization of the Euclidean concept of straight lines.

Exponential map and retraction

Conceptually, given a point x on the manifold $\mathcal{M}$, the exponential mapping $\mathrm{Exp}_x : T_x\mathcal{M} \longrightarrow \mathcal{M}$ assigns to a tangent vector ξ ∈ $T_x\mathcal{M}$, the unique point (Exp_x(ξ)) on the geodesic, whose distance from point x is given by the length of the tangent vector ξ. Note that Exp_x(tξ) is the geodesic which emanates from point x on the manifold, in the direction of tangent vector ξ. The exponential map concept is graphically illustrated in Figure 23.

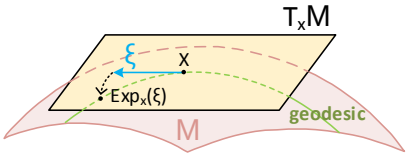


Figure 23: Illustration of an exponential mapping Exp_x at point x, which maps the tangent vector ξ ∈ Tx$\mathcal{M}$ to the point Exp_x (ξ) on the manifold.

Generalizing, retractions serve a similar purpose, namely mapping vectors in the tangent plane of point x to manifold points situated near point x. In particular, the exponential mapping can be regarded as the ideal retraction; however, unfortunately it requires the geodesic curve calculation, which is computationally expensive in practical scenarios (one being required to solve a second-order nonlinear ordinary differential equation, which generally does not admit a closed form solution). One way to alleviate the computational challenges associated with the Riemannian exponential map, while retaining the convergence properties of the optimization method, is by using *any* retraction instead. While the exponential map satisfies the constraint of zero-acceleration associated with a geodesic (which is hard to ensure in practical situations), a general retraction imposes no second (i.e., non-zero acceleration) or higher order requirements. For instance, while an exponential map corresponds to moving along the true geodesic, a first order retraction corresponds to moving along a curve on the manifold that initially moved in the specified direction (non-zero acceleration).

Parallel translation and vector transport

Parallel translation, as depicted in Figure 24, of a tangent vector ξ from point x to point y along a manifold curve C, can be defined as the affine connection ∇, where an affine connection is an (infinitesimal) linear relation between the tangent spaces of two neighbor manifold points. For a Riemannian manifold there exists a unique affine connection ∇ compatible with the Riemannian metric, namely the Levi-Civita connection (which is isometric, preserving the angles and the norm of the tangent vectors). In Euclidean spaces, the Levi-Civita connection is reduced to the classical directional derivative.

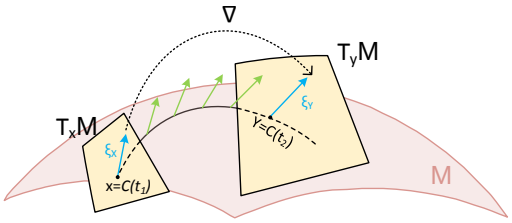


Figure 24: Illustration of parallel translation of tangent vector ξx from point x ∈ Tx $\mathcal{M}$ to point y ∈ Ty $\mathcal{M}$ along the manifold curve C.

The parallel translation induced by the Levi-Civita connection is usually along the geodesic defined by the exponential map of a tangent vector, and thus faces similar computational problems as previously discussed. As exponential mapping is a particular case of retraction, in a similar way, parallel translation along a geodesic is a particular form of the more general concept of vector transport (vector transport can be regarded as a first-order approximation of parallel translation). Roughly speaking, a vector transport τ specifies how to transport a tangent vector between two tangent spaces.

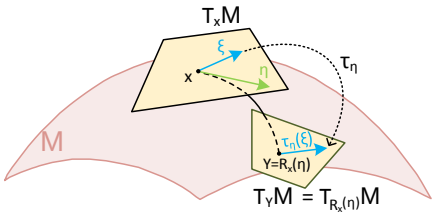


Figure 25: Illustration of vector transport τη, which transports vector ξ ∈ Tx$\mathcal{M}$ to vector τη (ξ) ∈ Ty $\mathcal{M}$.

Figure 25 presents the transport vector τ_η, i.e., the application mapping the tangent vector ξ from point x (which defines the tangent space $T_x\mathcal{M}$) to point y = $R_x(\eta)$ (which defines another tangent space, $T_y\mathcal{M}$), w.r.t. the tangent vector η ∈ $T_x\mathcal{M}$.

The computational efficiency of an optimization algorithm in a Riemannnian manifold depends to the largest extent on the choices of the affine connection ∇ and of the retraction R. In our framework, we employ the Levi-Civita connection. As far as the retraction is concerned, we follow the approach proposed in **[Absil08]**, specifically, we employ vector transport by differentiated retraction as a computationally tractable relaxation of parallel translation.

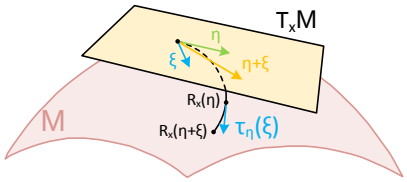


Figure 26: Illustration of differentiated retraction as vector transport.

Illustrated schematically in Figure 26 the vector transport is formally defined as:

$$\tau_\eta(\xi) \triangleq DR_x(\eta)[\xi] = \frac{d}{dt} R_x(\eta + t\xi)\Big|_{t=0} \qquad (34)$$

Having presented intuitively the basic Riemannian geometry concepts which serve our purpose and their practical, computationally efficient relaxations, we are now in position to address the adaptation of the CG optimization algorithm in the Riemannian manifold. In the case of statistical Riemannian manifolds, (29) are generalized to:

$$H_k = -G_k + \gamma_k \tau H_{k-1} \qquad (35)$$

where $G_k = \tilde{\nabla}\mathcal{L}_X(q(\theta^{(k)}))$, denotes the Riemannian gradient of the objective function, and τH_{k-1} defines the parallel translation of H_{k-1} (see Figure 21(b)). One iteration of the CG algorithm can be outlined as follows: at iteration *k*, the Riemannian gradient G_k of the objective function is evaluated, and the new search direction H_k for geodesic minimization is conjugated to the gradient and is evaluated to be a combination of the previous search direction H_{k-1} and the current Riemannian gradient at step G_k; finally a step is made in the direction of H_k to obtain x_{k+1}, which is the minimum of the objective function in the direction of H_{k-1}. In Algorithm 1, we present the formalism of the CG method on Riemannian manifolds.

Algorithm 4: Riemannian conjugate gradient method for optimizing the lower bound £

Input: the objective function $\mathcal{L}$: $\mathcal{M} \rightarrow$ R
Output: global minimize of $\mathcal{L}$
Set an initial point x_o on $\mathcal{M}$
Set the initial search direction $H_o = -G_o$
k=0
repeat
- Calculate the step length ak > 0 satisfying (30) and (31)
- Take a minimizing step by setting :
  $X_{k+1} = R_{xk}(\propto_k H_k)$
- Calculate γ_{k+1}^{FR}
- Compute the new search direction
  $H_{k+1} = -G_{k+1} + \gamma_{k+1}^{FR}\, \tau_{\propto_k H_k}^{R}(H_k)$

Until sufficiently minimizes the objective $\mathcal{L}$

In Algorithm 1, the employed vector transport $\tau^{R}{}_{\propto_k H_k}(H_k)$ is defined as:

$$\begin{cases} \dfrac{\|H_k\|_{x_k}}{\left\|\tau_{\propto_k H_k}(H_k)\right\|_{R_{x_k(H)}}}\, \tau_{\propto_k H_k}(H_k), for \left\| \tau_{\propto_k H_k}(H_k)\right\|_{x_k} \leq \|H_k\|_{x_k} \\ \tau_{\propto_k H_k}(H_k), \quad otherwise \end{cases} \qquad (36)$$

where $\tau_{\alpha_k H_k}(H_k)$ is the vector transport associated with the differentiated retraction R **[Absil08]**, which is scaled when the norm of the previous search direction is increased **[Sato13]**. The step size α_k is chosen such that it obeys the following two relations, known as the strong Wolfe conditions **[Nocedal06] [Ring12]**, i.e., the sufficient decrease and curvature conditions which prevent the step length to be excessively short:

$$\mathcal{L}(a) \leq \mathcal{L}(x_k) + C_1 \propto_k \langle G_k H_k \rangle_{x_k} \qquad (37)$$

$$|\langle grad\, \mathcal{L}(a), D_a[H_k]\rangle_a| \leq C_2 |\langle G_k H_k \rangle_{x_k}| \qquad (38)$$

where for compactness of notation, we employed a instead of $R_{x_k}(\propto_k H_k)$.

The global convergence of the Riemannian conjugate gradient was proved in **[Ring12]**, under the assumption that the vector transport does not increase the norm of the tangent vectors. Note that while for parallel translation, a vector is moved along a geodesic, staying parallel to itself and preserving its magnitude, vector transport imposes no constraint of vectors norm preservation. In

[Sato13], the authors relax the norm constraint, and rescale the transported vector only when necessary (i.e., in the cases when the tangent vector norm was increased during transport), while ensuring the global convergence property. As concerns the speed of convergence, using the rescaled vector transport proposed in **[Sato13]**, results in supralinear convergence of the sequence $\{x_k\}$ to the global minimizer.

4.2.6. Conclusion and Future Work

In this section we introduced a Probability Density Functions (PDFs) based Integrated Circuit (IC) reliability assessment framework. Since the efficiency of reliability driven design-time optimizations and of run-time management frameworks directly depend on the reliability evaluation accuracy, instead of relying on a single probabilistic value to reflect the reliability status of a circuit, we propose to employ a distribution of probabilities for a closer adherence to a faulty circuit stochastic behavior. To the extent of our knowledge, the proposed approach describes the first attempt to assess the reliability of a circuit based on PDFs.

The framework is based on a variational inference method, and exploits the geometry of the statistical manifold to yield a fast and scalable reliability assessment approach, which can be potentially integrated in reliability aware synthesis tools and Dynamic Reliability Management frameworks.

Ongoing work consists in the numerical assessment of the proposed approach. As avenue to future work, we aim to introduce the time component in the proposed framework and extend its area of applicability to sequential circuits. Other potential directions of research continuation, include exploring a measure different than the KL divergence for assessing two distributions dissimilarity **[Gibbs12]** **[Minami02]**, and if found appropriate, further developing the afferent inferential mathematical apparatus.

5. Error Coding Driven Graph Augmentation (Task 5.3)

Abstract: In this section we present a novel method to design fault tolerant circuitry. In particular we focus on improving the fault tolerance capability of combinatorial logic by means of error correction codes. Our method is completely general and can be applied to any combinatorial logic without prior knowledge of the circuit functionality. The idea is adapted from the field of forward error correction for telecommunications. The study presented here focuses mainly on encoding aspects for protecting fault prone Boolean functions.

Publications: Unpublished work (work in progress, a conference paper planned).

5.1. Codeword Prediction Encoder (CPE) for Fault Prone Boolean Functions

The approach presented in the previous section attempts to reduce the error probability on the output of a combinatorial circuit by choosing an optimal realization of the Boolean function under investigation.

A different approach to improve fault tolerance is based on the use of methods derived from Error Control Coding (ECC) theory to protect the combinational logic that implements a particular Boolean Function. The focus of this approach is not on changing the combinational logic but on augmenting it

to enable the retrieval of the correct output even if errors have occurred. The process carries remembrance with the process of adding redundancy to the transmitted signal in a communication system, and is described in details in the next section.

The main challenges include:

- How to add minimum amount of redundancy to correct the given number of errors (see Task 5.6)?
- How implement a system capable to take advantage of the added redundancy?

In relation to the second problem, two scenarios are under investigation. In the first, we consider the case where the device/logical block that needs to "receive" the augmented output if fault free. In the second both logic blocks are assumed error prone.

The first scenario presents an evident asymmetry between the "encoder" and the "decoder". An example of unbalanced system where this scenario is a realistic assumption is any system where some of the modules have harder power constraints compared with others. Examples vary from wireless communication between battery powered host and plugged in master, satellite and space exploration communication systems or even in-chip voltage islands. The "perfect decoder" assumption allows us to consider ECC codes for which a faulty decoder is not available. In particular it makes possible to use coding schemes other than Low Density Parity Check Codes, the only known ECC for which fault tolerant decoder exist. In doing so, it allows the comparison of traditional error correction codes and Low Density Parity Check codes. This scenario together with implementation results for several ECC schemes are presented in section 5.2.

The second scenario assumes both "encoder" and "decoder" to be fault prone. Given such situation only LDPC codes are valid candidates. Moreover to allow strict mathematical analysis assumptions are made on the characteristic of the Boolean function to be protected. Preliminary results and investigation on size and feasibility of the proposed method in this scenario are presented in section 5.3

5.2. Ideal Decoder/Generic Function

The approach presented here can be seen as an expansion of the Check Symbols Generation **[Touba97] [Mohanram03]** and the Parity Prediction Function **[Sogomonjan93] [Manich96] [Ko01]**, where circuitry is added to a combinatorial network to generate extra bit to ensure parity. We formalize these approaches and extend them to take full advantage of the power of error correction codes to enable correction of the faults not just detection.

5.2.1. Proposed Scheme

Forward error correction methods generate a codeword $C \in \boldsymbol{C}$, with $\boldsymbol{C}$ being the particular code being used, by encoding the information message u. We call the encoder $\mathbb{E}$. Assuming without loss of generality that the encoder is systematic the codeword c is composed by u and several extra symbols called the parity P (Figure 27).

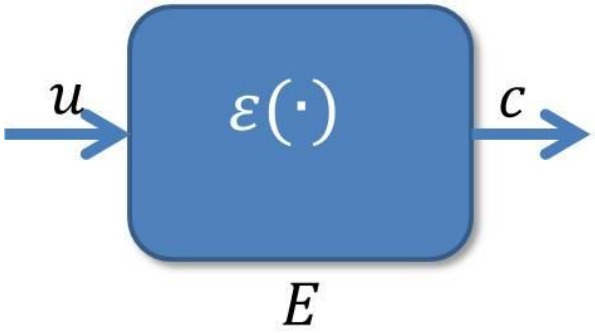


Figure 27: Normal Encoder

We look for a way to apply such procedure on an unreliable combinatorial circuit $\mathbb{CC}$ with inputs I and outputs O. The simplest solution would be to encode the outputs o (Figure 28), however this approach suffers from the fact that if a fault incurs in $\mathbb{CC}$ then $\tilde{O}$ is the input of the encoder that gives $\tilde{c}$. $\tilde{c}$ may or may not be a codeword of $\boldsymbol{C}$, depending of the presence of faults in the encoder logic, however no decoder will be able to retrieve I from it. Another disadvantage of the scheme is that from hardware prospective the concatenation of $\mathbb{CC}$ and $\mathbb{E}$ increases the critical path and hence limits the throughput.

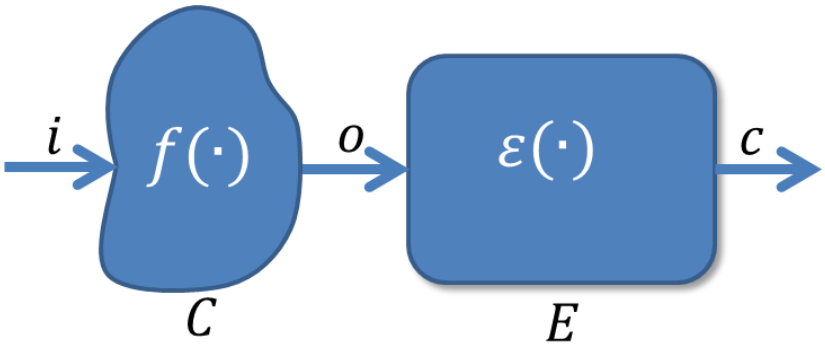


Figure 28: Erroneous architecture

We propose to combine the encoding process in the combinatorial circuit to ensure that the outputs O of the new combinatorial logic $\mathbb{CE}$ is a codeword as depicted in Figure 29.

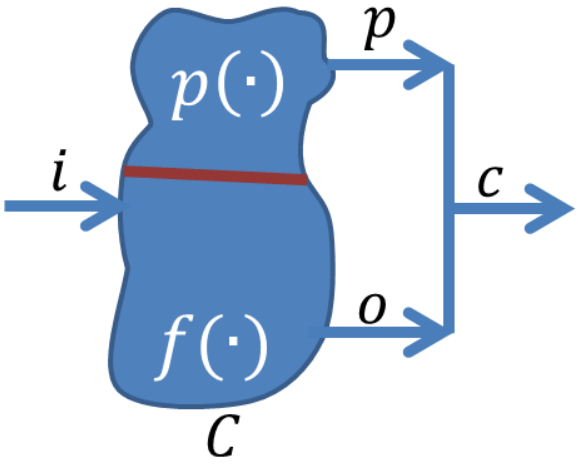


Figure 29: Proposed Encoder

In this new scenario $\mathbb{CE}$ computes not only the outputs o but also a set of parity p that guarantee$[O|P] \in \boldsymbol{C}$, (systematic encoding is used).

To understand how the new combinatorial logic is obtained lets define as $\mathcal{E}(\cdot)$ the function computed by E and let's assume that we are working with binary codes, then $\mathcal{E}: GF(2)^k \rightarrow GF(2)^n$ with (k, n) the dimensions of the code $\boldsymbol{C}$. The functionality of $\mathbb{CC}$ can be described as a function $\mathcal{F}(\cdot)$ with $\mathcal{F}: GF(2)^l \rightarrow GF(2)^o$ where l is the number of binary inputs and o is the number of binary outputs. The function $\mathcal{P}(\cdot)$ that maps the inputs directly to the parity is $\mathcal{P}: GF(2)^l \rightarrow GF(2)^m$ and is the composition $\mathcal{E}(\mathcal{F}(\cdot))$. This can be seen simply by considering how its result must be equivalent of computing $\mathcal{F}(\cdot)$ and then $\mathcal{E}(\cdot)$ on the results.

It is evident that even if the operation is equivalent to serially concatenating the two blocks it computes the parity on an independent path then the original combinatorial logic hence it does not suffer from the fault propagation scenario discussed above.

In a sense the function $\mathcal{P}(\cdot)$ **predicts** the parity of the outputs O from the inputs I as if a standards encoder were implemented with o as inputs.

For all linear FEC codes the encoding process can be expressed as a matrix multiplication of the input message and the generator matrix $\boldsymbol{G}$, the composition $\mathcal{E}(\mathcal{F}(\cdot))$ is then equivalent to:

$$\mathcal{P}(j) = \sum_{b=0}^{o} \mathcal{F}_{b(j)} \boldsymbol{G}(b, j) \quad j \in \{0, \dots, m\} \tag{39}$$

The combinatorial logic that compute the parity is a linear combination of the various functions that compute the output bits, as such the resulting logic may be costly, both with regards with area consumption and in term of critical path.

5.2.2. Hardware Impact Investigation

This section presents several applications of this encoding technique to two IP cores. The use of different ECC codes is investigated. For reference the results are compared with the original size/timing and with the Triple Modular Redundancy (TMR) scheme **[Taylor68-a]**.

- **IP cores**

To make the study meaningful for real application we applied the encoding scheme on two IP cores commonly used in telecommunication systems. In particular both the cores presented are used in

Optical Transport Network (OTN) and both have been tested with throughput 100Gb/s. The cores have been chosen for the fact that they are both purely combinatorial and of considerable size.

Moreover, the codes are parametric and the number of inputs/outputs can easily be modified to match the size of the ECC code.

I. The Scrambler;
II. The Chien Search block for Reed Solomon decoding.

- **ECC codes used in our analysis**

Several codes are investigated to span a large variety of possibilities in terms of codelength, error correction capability, encoding/decoding complexity, etc. These include:

- Hamming code;
- BCH codes;
- Low Density Parity Check (LDPC) codes;
- Unequal Error Protection (UEP) LDPC codes;
- Low Density Generator Matrix (LDGM) codes.

- **Implementation**

Each of the two IP cores has been implemented using all the proposed codes as fault correction option. All implemented CPE logic have $k \sim 1000$ Inputs and both cores have been designed to match the code information dimension. For ECC with dimension $k_1 < k$, the input signals have been grouped in $\frac{k}{k_1}$ groups and the ECC applied for each group.

- **Hardware Implementation Results**

In this section the area and timing results from implementing the schemes on ASIC technology(TSMC 45nm) are presented. Area and Delay results for the application of the proposed fault protection scheme for the Scrambler and Chien Search cores are presented in Table 3 and

Table 4, respectively. Figure 30 and Figure 31 show a graphical representation of the area consumption while Figure 32 and Figure 33 present charts of the delay comparison for the two cores.

It can be seen how the implementation of the CPE protection can have a significant impact in term of area and delay. It is also evident how the cost of the scheme is dependent on the combinatorial logic to be secured.

In the case of the Scrambler core (Table 3, Figure 30, Figure 32) the logic consists of long chains of XORs. Appling Eq. (14) to it results again on a simple chain of XOR. Moreover due to cancellation effects (XORing the same input twice is equivalent to not consider the input) the average length of the XOR chain does not increase. As a result most of the investigated CPE schemes have similar delay and similar area consumption. A considerable variation of this trend is the case of the Hamming code, for which the implementation has a smaller area consumption and delay. This is due to the simplicity of the code used that protects only against single error events.

The combinational logic of a Chien Search core is considerably less structured and more complex than the Scrambler circuit. The implementation of a Chien Search block requires the implementation of finite field arithmetic operations together with multiplexing and adders. These varieties of operations make the combinatorial circuit considerably less structured and more complex. As a consequence, the parity calculation logic does not simplify and it results in higher area consumption and longer delay (Table 4, Figure 31, Figure 33). This is particularly true for LDPC codes since they require high dimension to achieve good performances that means more terms in Eq. (14).

Table 3: Scrambler-core Synthesis results comparison.

Code Scheme	Area	Overhead (%)	TMR comp. (%)	% TMR saving	Delay (ns)	% of original	TMR comp. (%)	% TMR saving
Scrambler	130771	-	-		3.19	100		
Triple Modular Redundancy	261543	200	100	0	3.21	100	100	6.54
Hamming	101897	77	39	61	2.97	93.10	92.52	7.17
BCH	212022	162	80	19	2.98	93.42	92.83	6.85
LDGM (McKay design)	172492	131	65	34	2.99	93.73	93.15	7.48
LDGM (Random design)	214830	164	80	18	2.88	90.28	89.72	6.85
LDPC (McKay design)	265280	202	98	-1	2.93	91.85	91.28	10.28
LDPC (Richardson design)	265298	202	99	-1	3.02	94.67	94.08	8.72
UEP-LDPC	265659	203	99	-2	3	94.04	93.46	5.92

Table 4: BCH Chien Search Synthesis results comparison.

Code Scheme	Area	Overhead (%)	TMP comp. (%)	% TMR saving	Delay (ns)	% of original	TMR comp. (%)	% TMR Saving
Chien Search core	7310	0			3.04	100		
Triple Modular Redundancy	14620	200	100	0	3.2	105.26	100.00	-39.06
Hamming	18921	258	82	-29	2.48	81.58	77.50	-27.81
BCH	73008	998	300	-399	4.09	134.54	127.81	31.88
LDGM (McKay design)	19662	268	85	-34	2.18	71.71	68.13	22.50
LDGM (Random design)	20512	280	82	-40	2.2	72.37	68.75	31.88

LDPC (McKay design)	94110	1287	377	-544	4.65	152.96	145.31	31.25
LDPC (Richardson design)	104972	1436	418	-618	4.69	154.28	146.56	-45.31
UEP-LDPC	111025	1518	444	-659	4.45	146.38	139.06	-46.56

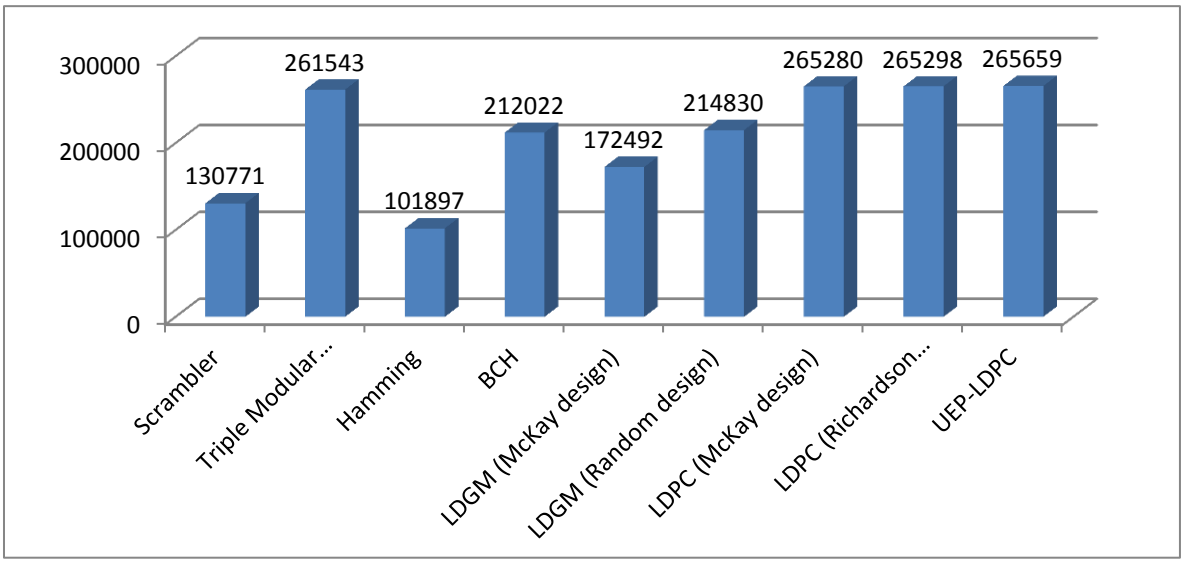


Figure 30: Area consumption for several CPE schemes applied on Scrambler core

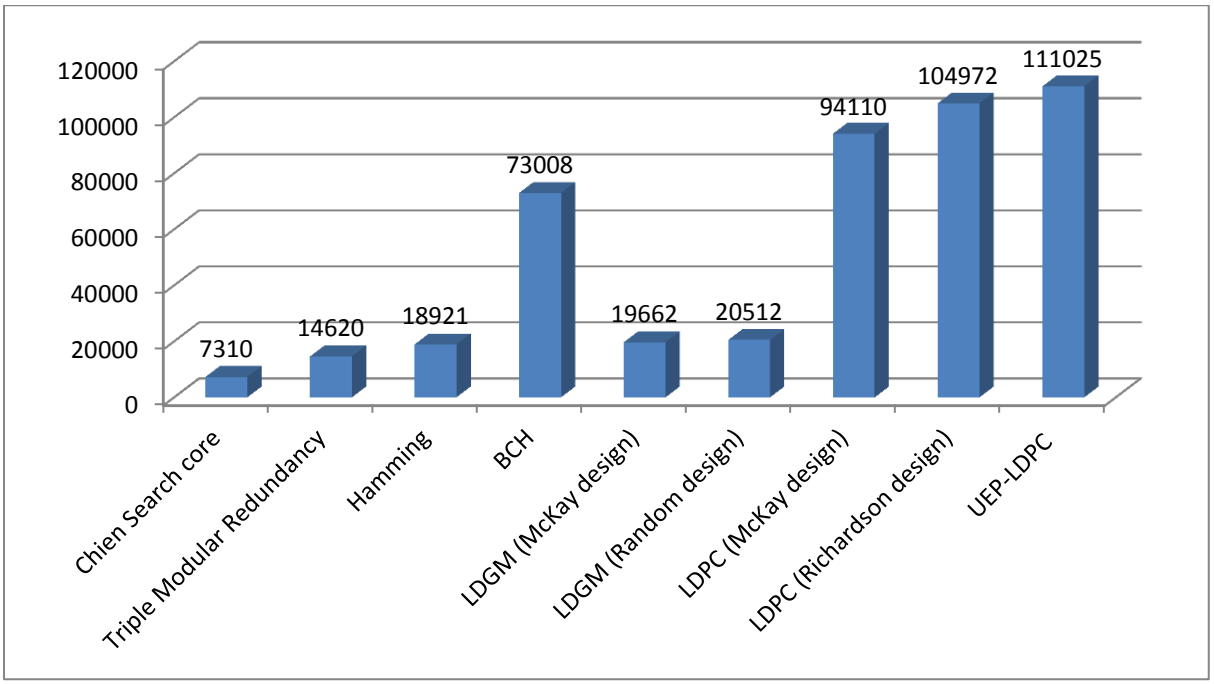


Figure 31: Area consumption for several CPE schemes applied on Chien Search core

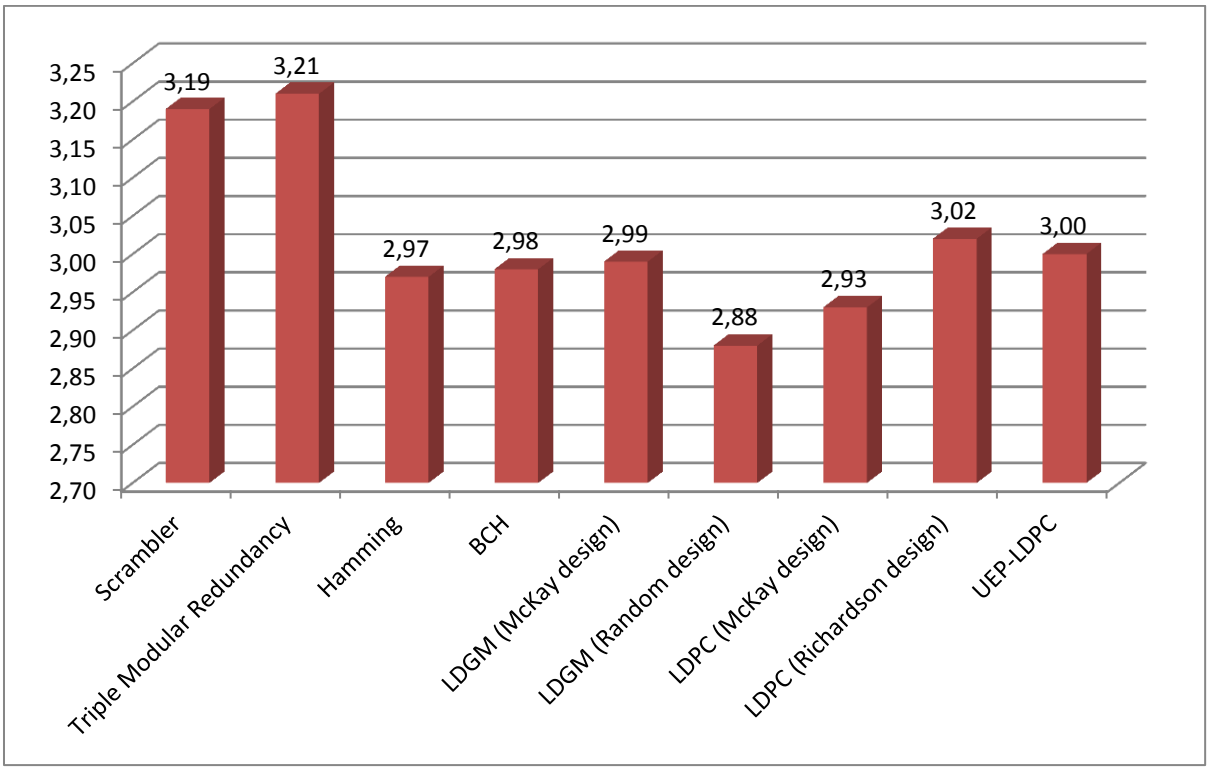


Figure 32: Delay comparison for CPE applied on Scrambler Core

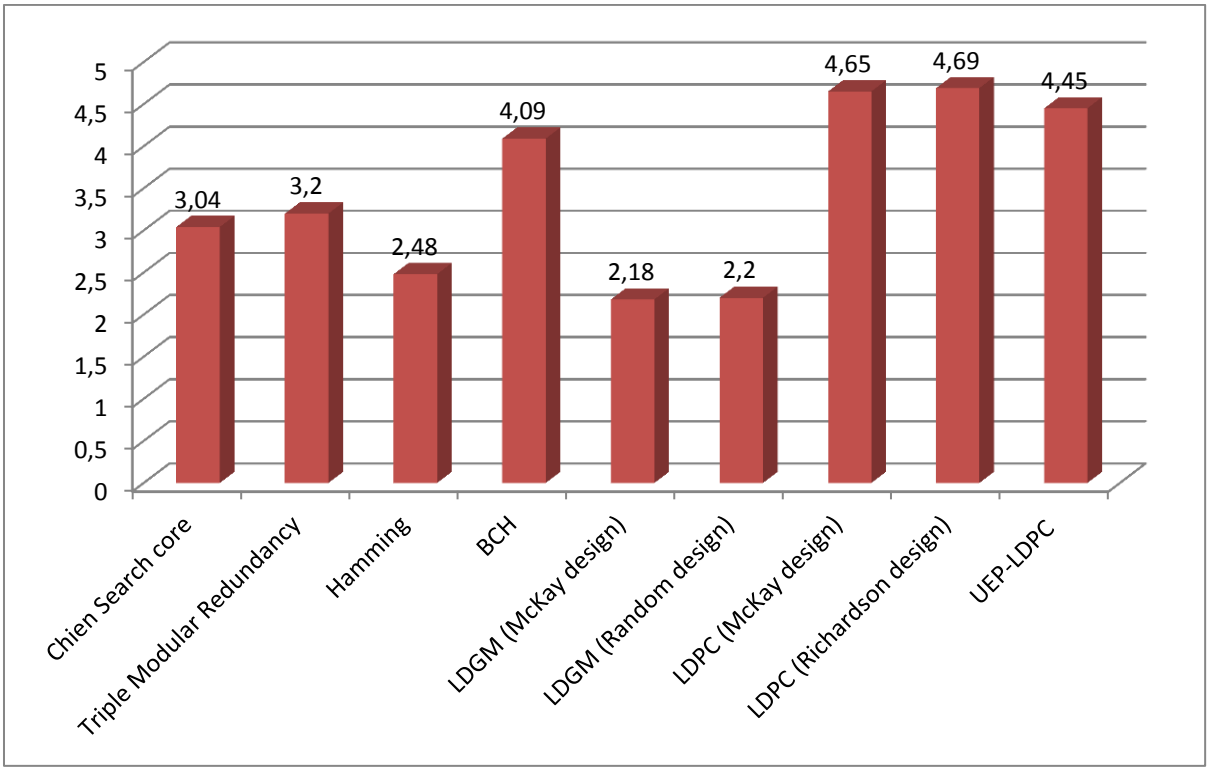


Figure 33: Delay comparison for CPE applied on Chien Search Core

From the data gathered, the application of the proposed scheme is an appealing solution, even for powerful ECC, in the case of the regular combinatorial circuits, while only the use of simple Hamming or TMR schemes seems to be a feasible option for irregular complex logic. Several considerations and possible ways forward are presented in next section. These avenues will be studied more in detail during the next year.

5.2.3. Conclusion and Future Work

Each of the ECC code presented has a different error correction capability that has not been considered yet. Considering for example the Chien search core considered above, Table 4 shows that the use of LDPC codes requires substantially more area that the others, however it does not consider that they guarantee the ability to correct complex multiple-errors events. Ideally, in order to assess the quality of an ECC scheme over the other a figure of merit taking into consideration the extra hardware requirements balanced out against the fault tolerance improvements that need to be developed.

How to define and obtain such figure of merit is an open question. In particular, attempting to define a method to evaluate the performance of the CPE scheme leads to many questions: Should only the maximum number of errors be considered or should a simulation based approach be taken? Should the combinational circuit complexity being considered when selecting a particular code? Does the logic network representing the circuit have an effect on the error type and distribution? Is there a systematic way to synthesise a logic network which is decodable?

Another point of interest is the link between the generator matrix $\boldsymbol{G}$, of the ECC code and the complexity of the resulting CPE architecture. The area of the parity prediction circuitry is proportional to the complexity of the generator matrix of the code, i.e., the numbers of ones in the matrix. It is hence possible to reduce the area by simple choosing an optimal $\boldsymbol{G}$ for a given ECC code. Morover, for each code type the growth of the number of ones in relation to the information size k is different. For some codes it grows linearly for other exponentially. This means that the choice of the code to use is dependent on the number of inputs of the combinatorial logic under test. A better understanding of this link, and its repercussions on the hardware consumption, is needed. This will allow defining a systematic methodology to decide on the best ECC code to be used for a given circuit. Such a methodology will be very important in deriving a framework towards the development of a Boole-Shannon type limit for designing reliable digital circuits.

5.3. Unreliable Decoder/Specific Functions

In this section, we extend our investigation on the problem of protecting the functionality of a combinatorial circuit in the scenario where also the decoder is implemented with faulty hardware. To enable us to deal with such scenario several assumption are necessary. First we assume that the functionality of the combinatorial circuit $\mathbb{CC}$ can be described as a *linear Boolean function* $\mathcal{F}: GF(2)^l \to GF(2)^k$. In other words, there exists a binary matrix $F \in M_{k,l}(GF(2))$ such that $O \stackrel{\text{def}}{=} \mathcal{F}(I) = F \cdot I,\ \forall I = (i_1, \dots, i_l) \in GF(2)^l$. Note that this restriction was not necessary in the previous section.

We further assume that the circuit $\mathbb{E}$ implements the encoding of a *systematic Low-Density Parity-Check* (LDPC) code. The significance of this assumption is twofold.

- First, the encoding function $\mathcal{E}: GF(2)^k \rightarrow GF(2)^n$ is *systematic*, meaning that that for any $O \in GF(2)^k$, the corresponding codeword $C \stackrel{\text{def}}{=} \mathcal{E}(O)$ is of the form $C = (O, P)$, with $P \in GF(2)^m$, $m = n - k$.
- Second, there exists a sparse parity-check matrix H, such that $H \cdot C = 0$, for any codeword C. Without loss of generality, we may assume that $H = [H_1|H_2]$, with $H_1 \in M_{m,k}(GF(2))$ and $H_2 \in M_{m,m}(GF(2))$, such that H_2 is invertible.

Therefore, we may write:

$$H \cdot C = 0 \Leftrightarrow [H_1|H_2] \cdot (O, P) = 0 \Leftrightarrow P = H_2^{-1} \cdot H_1 \cdot O = H_2^{-1} \cdot H_1 \cdot F \cdot I$$

Then, by defining $S = H_2^{-1} \cdot H_1 \cdot F \in M_{m,l}(GF(2))$, we have:

$$P = S \cdot I, \qquad \forall I \in GF(2)^l$$

The matrix S is the equivalent of the function $\mathcal{P}(\cdot)$presented in Eq. (14) once the two assumptions taken are valid.

Finally, we denote by $\mathbb{S}$ a combinatorial circuit implementing (the multiplication by) S.

Using the above notation, a fault-tolerant implementation $\mathcal{F}$ can be achieved by using the CPE approach as shown in Figure 34, where $\mathbb{D}_H$ denotes a circuit implementing a decoding algorithm $\mathcal{D}_H$ of the LDPC code defined by the parity-check matrix H.

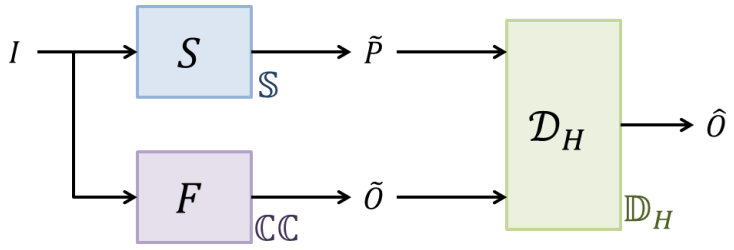


Figure 34: CPE approach for fault-tolerant computing of $O = \mathcal{F}(I) = F \cdot I$

Since we consider noisy (unreliable) circuits, we always distinguish between the functionality of a circuit (e.g., $S, F, \mathcal{D}_H$) and the circuit itself (e.g., $\mathbb{S}, \mathbb{CC}, \mathbb{D}_H$). In particular, the circuits $\mathbb{S}$ and $\mathbb{CC}$ produce noisy versions, denoted by $\tilde{P}$ and $\tilde{O}$, of $P = S \cdot I$ and $O = F \cdot I$, respectively. The decoding circuit $\mathbb{D}_H$ produces an estimate of O, denoted by $\hat{O}$.

The success of the decoding operation (i.e., $\hat{O} = O$) depends on the reliability of the three circuits from Figure 34: the reliability of $\mathbb{CC}$ and $\mathbb{S}$ circuits determines the noisiness of the decoder's input $(\tilde{O}, \tilde{P})$, while the reliability of $\mathbb{D}_H$ determines the ability of the decoder to provide effective error correction.

In Deliverable D3.1 [i-RISC/D3.1], we investigated the performance of several LDPC decoders running on noisy hardware. Several Min-Sum-based and FAIDs decoders have been analyzed, and we proved that they are able to provide reliable error protection, even if they run on noisy hardware. Moreover, we have shown that some decoders are intrinsically robust to circuit noise. For instance, the noisy Self-Corrected Min-Sum (SCMS) decoder has been shown to exhibit almost the same performance as the noiseless one **[Kameni13-a, Kameni13-b]**. These results make us confident on the potential of the CPE approach in providing fault-tolerant implementations of combinatorial circuits.

5.3.1. Overall Circuit Complexity Analysis

Throughout this section, by *size* of a circuit we mean the number of the elementary logic gates that have to be interconnected to form the circuit function.

As it can be seen from Figure 34, the size of the fault-tolerant CPE implementation is given by:

Size(**CPE**) = Size($\mathbb{CC}$) + Size($\mathbb{S}$) + Size($\mathbb{D}_H$)

Although LDPC decoding algorithms consist of iterative message passing procedures, in practical implementations only the circuitry corresponding to a decoding iteration has to be instantiated in hardware. In this case, the size of the decoding circuit is known to increase linearly with the size of the decoding input. Thus, assuming that the rate k/n of the code is constant, we can write Size($\mathbb{D}_H$) = $\mathcal{O}(n)$ = $\mathcal{O}(k)$.

If F is a random matrix, the size of the combinatorial circuit implementing F increases linearly with $l \times k$. Therefore, assuming that the ratio l/k of matrix dimensions remains constant, we can write Size($\mathbb{CC}$) = $\mathcal{O}(l \times k) = \mathcal{O}(k^2)$. Similarly, we have Size($\mathbb{S}$) = $\mathcal{O}(k^2)$.

Moreover, since $\mathbb{CC}$ and $\mathbb{S}$ are of comparable sizes, which asymptotically dominate the size of the decoding circuit $\mathbb{D}_H$, we can (asymptotically) approximate the size of the fault-tolerant CPE implementation by:

Size(**CPE**) $\approx 2 \times$ Size($\mathbb{CC}$)

This is to be compared with the size of classical approaches for fault-tolerant computing, as for instance *N*-Modular Redundancy (*N*-MR) systems, illustrated in Figure 35. In such systems, the combinatorial circuit $\mathbb{CC}$ is replicated *N* times, and a majority voting (MV) systems is used to estimate the correct output the circuit. Thus, clearly:

Size(***N*-MR**) = $N \times$ Size($\mathbb{CC}$)

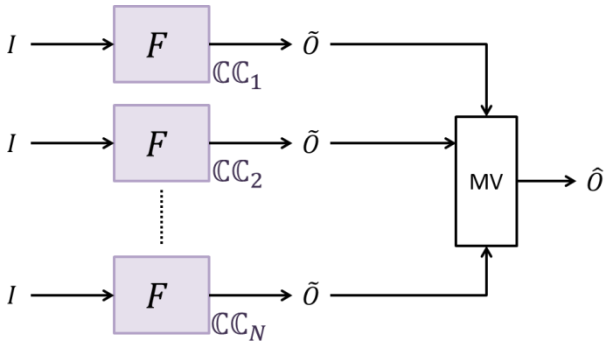


Figure 35: N-MR approach for fault-tolerant computing of $O = \mathcal{F}(I) = F \cdot I$

Finally, it is also worth noting that in contrast with the *N*-RM approach, which relies on repetition codes, the proposed CPE approach relies on LDPC codes, known as *capacity approaching codes*, which greatly improves the reliability of the whole system.

5.3.2. The Case of Sparse S Matrix

The case of a sparse S matrix is probably one of the most advantageous cases. First, if S is sparse, the size of the circuit computing S increases only linearly with k, that is, Size($\mathbb{S}$) = $\mathcal{O}(k)$. Second, the

sparsity of S also results in a lower error probability on $\tilde{P}$, hence improving the reliability of the decoder input.

Recall that matrices F, S, and $H = [H_1|H_2]$ are related by $S = H_2^{-1} \cdot H_1 \cdot F$. Therefore, for a given matrix F, we would like to find sparse matrices $H = [H_1|H_2]$ and S, such that:

- $S = H_2^{-1} \cdot H_1 \cdot F$, and
- The LDPC code defined by H has good decoding performance.

Such a decomposition method is currently under investigations.

Finally, it is worth noting that exist matrices F such that S is sparse: it suffices to choose first $H = [H_1|H_2]$ and S, and then to take F the matrix defined by $F = H_1' \cdot H_2 \cdot S$, where H_1' is such that $H_1' \cdot H_1 = I$. Although this does not necessarily represent a "practical" case, it could be seen as a first step toward a proof of concept of the proposed approach.

These approaches (which are part of the Task 5.3) will be further investigated during the second year of the project.

6. Conclusions and Next Steps

In this report we summarized the activities within WP5 for the first 12 months of the i-RISC project. We investigated existing data structures used in the synthesis of digital circuits that are also useful in the context of efficient, fault tolerant circuit synthesis. A special emphasis was put on the versatility of the data structure in terms of supported types of logic networks/circuits(combinational and sequential), scalability with increased circuit complexity, pre and post- technology mapping optimization results, availability of open source tool support, etc. We selected AND-Inverter Graphs (AIG) as basic data structures for circuit representation due to their scalability and ability to capture the error models developed in WP2, as well as power, area, and delay information. This data structure is suitable to both sequential and combinational circuits and has been proven on both ASIC and FPGA technologies. We further utilized the selected data structure in the context of the ABC open source tool to synthesise and analyse the reliability of some simple combinational circuits. An initial tool incorporating the AIG and some local transformation rules based on Boolean algebra has been proposed for computing the reliability function and it was demonstrated that through the selective application of the proposed rules, the reliability could be significantly improved. The proposed data structure will also be used to explore systematic multi-objective optimization methodology of fault tolerant circuits in Task 5.4.

Following the data structure selection, a novel design flow was proposed, which combines state of the art academic (including custom i-RISC) tools with more established industry tools. The proposed design flow is used to synthesise, optimize, analyze and validate our hypothesis and results. Some industry tools which combine HSPICE and Verilog/VHDL were also evaluated and will be used throughout the WP6 tasks in conjunction with i-RISC custom tools. Tasks 5.1 and 5.2 were successfully completed as we proved that the proposed data structures and design flow allow custom algorithms and methodologies to be integrated into widely used circuit design frameworks.

In the context of Task 5.2 we also introduced a Probability Density Functions (PDFs) based Integrated Circuit (IC) reliability assessment framework. This fast and highly accurate approach, which is fundamentally different than state of the art methods, is motivated by the fact that the effectiveness of reliability driven design-time optimizations and reliability run-time management frameworks directly depends on the reliability evaluation accuracy. To increase the evaluation accuracy, instead of relying on a single probabilistic value to reflect the reliability status of a circuit, we proposed to employ a distribution of probabilities for a closer adherence to a faulty circuit stochastic behavior. To the best of our knowledge, we are the first to propose a method to assess the reliability of a circuit based on PDFs. The proposed framework, which is based on a variational inference method, and exploits the geometry of the statistical manifold to yield a fast and scalable reliability assessment approach, can be utilized in reliability aware synthesis tools and constitutes a first step towards achieving Task 5.4 goals.

We also started work on the Task 5.3 with a number of encouraging developments on Error Correction Coding driven graph augmentation. We identified a number of combinational functions, which were then encoded and the hardware implementation results for a number of coding schemes were compared and contrasted. Further efforts will be in developing also efficient decoding schemes for the fault tolerant circuits in the presence of a given error model. An initial set of functions was identified which will allow us an initial analysis of complexity of implementation versus error correction capability (a first step in our quest towards a Boole-Shannon limit for digital circuits). The

emphasis for the remaining period will be in developing tools which will systematically synthesize reliable circuits, using a reliability driven multi-objective optimization approach. Also, we will continue our quest towards expanding the number of functions for which a Boole-Shannon limit can be established.

References

[ABC12] ABC: A system for Sequential Synthesis and Verification, Berkeley Verification and Synthesis Research Center, 2012 (www.eecs.berkeley.edu/~alanmi/abc/abc.htm).

[Absil08] P. A. Absil, R. Mahony, R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2008.

[Amari00] S. Amari, and H. Nagaoka, *Methods of information geometry*, Volume 191 of Translations of Mathematical Monographs, American Mathematical Society, 2000.

[Amari85] *Differential-geometrical methods in statistics*. Volume 28 ofLecture Notes in Statistics, Springer-Verlag, 1985.

[Amari98] S. Amari, "Natural gradient works efficiently in learning," in *Neural Computation 10(2)*, 1998, pp. 251–276.

[Bahar03] I. Bahar, J. L. Mundy, and J. Chen, "A probabilistic-based design methodology for nanoscale computation," in *International Conference on Computer Aided Design*, 2003, pp. 480–486.

[Bala07] P. Balasubramanian and K. Anantha, "Power and delay optimised graph representation for combinational logic circuits", *International Journal of Computer Science*, Vol. 2, No. 1, 2007, pp. 47–53.

[Balasubramanian07] Balasubramanian, P.; Edwards, D.A., "Synthesis of Power and Delay Optimized NIG structures," Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on , vol., no., pp.239,242, 22-26 April 2007.

[Balasubramanian06] P. Balasubramanian, C. H. Narayanan and K. Anantha, "Low Power Design of Digital Combinatorial Circuits with Complementary CMOS Logic", *International Journal of Electronics, Circuits and Systems*, Vol. 1, No. 1, 2006, pp. 10–18.

[Beal03] M. J. Beal, "Variational algorithms for approximate bayesian inference", 2003.

[Bhaduri05] D. Bhaduri and S. Shukla, "Nanolab: A tool for evaluating reliability of defect-tolerant nano architectures," in *IEEE Transactions on Nanotechnology, 4(4)*, 2005, pp. 381–394.

[Bollig96] Beate Bollig, Ingo Wegener. Improving the Variable Ordering of OBDDs Is NP-Complete, IEEE Transactions on Computers, 45(9):993–1002, September 1996.

[Borkar05] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation,"Micro, IEEE, vol. 25, no. 6, pp. 10–16, 2005.

[Brayton 10] R. Brayton and A. Mishchenko, "Abc: An academic industrial-strength verification tool," in Proceedings of the 22Nd International Conference on Computer Aided Verification, pp. 24–40, 2010.

[Brayton87] R. Brayton, R. Rudell, A. Sangiovanni-Vincentelli, and A. Wang, "Mis: A multiple-level logic optimization system," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 6, no. 6, pp. 1062–1081, 1987.

[Brayton10] R. Brayton and A. Mishchenko, "ABC: An academic industrial-strength verification tool", *Proceedings of CAV'10, Springer, LNCS 6174*, 2010, pp. 24–40.

[Bryant86] R. Bryant, "Graph-based algorithms for Boolean function manipulation", *IEEE Transactions on Computers*, Vol. 35, No. 8, 1986, pp. 677–691.

[Choudhury09] M.R. Choudhury, and K. Mohanram, "Reliability analysis of logic circuits." in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(3)*, 2009, pp. 392–405.

[Constantinescu03] C. Constantinescu, "Trends and challenges in vlsi circuit reliability,"Micro, IEEE, vol. 23, no. 4,pp. 14–19, 2003.

[Cranwell07] R. Cranwell, "Ground Vehicle Reliability Design-for-Reliability," in *DoD Maintenance Symposium, Orlando, FL*, 2007.

[Crowe98] D. Crowe, and A. Feinberg, "Stage-Gating Accelerated Reliability Growth in an Industrial Environment," in *Proceedings Institute of Environmental Sciences*, 1998, pp. 1–9.

[Darringer81] J. Darringer, W. H. Joyner, C. Berman, and L. Trevillyan, "Logic synthesis through local transformations," IBM Journal of Research and Development, vol. 25, no. 4, pp. 272–280, 1981.

[Mehrotra11] R. Mehrotra, T. English, M. Schellekens, S. Hollands, and E. Popovici, "Timing-driven power optimisation and power-driven timing optimisation of combinational circuits," Journal of Low Power Electronics, vol. 7, no. 3, pp. 364–380, 2011.

[Ercolani89] S. Ercolani, M. Favalli, et. al., "Estimate of signal probability in combinational logic networks," in *Proceedings of the 1st European Test Conference*, 1989, pp. 132–138.

[Felipe08] M. Felipe, R. Teresa and T. Yago, "Disjoint Region Partitioning for Probabilistic Switching Activity Estimation at Register Transfer Level", PATMOS, 2008, pp. 399–408.

[FelipeY08] M. Felipe, T. Yago and R. Teresa, "A BDD Proposal for Probabilistic Switching Activity Estimation", International Conference on Design of Circuits and Integrated Systems (DCIS), Grenoble, France, Nov. 2008, pp. 54–62.

[Felipe05] M. Felipe, T. Yago and R. Teresa, "Exploiting VHDL-RTL features to reduce the complexity of power estimation in combinational circuits", Research in Microelectronics and Electronics, Jul. 2005, pp. 111–114.

[Figueiro11] Figueiro, T.; Ribas, R.P.; Reis, A.I., "Constructive AIG optimization considering input weights," Quality Electronic Design (ISQED), 2011 12th International Symposium on , vol., no., pp.1,8, 14-16 March 2011.

[Franco08-a] D.T. Franco, M.C. Vasconcelosa, L. Navinera, and J.-F. Navinera, "Reliability analysis of logic circuits based on signal probability," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2008, pp. 670–673.

[Franco08-b] ——, "Signal probability for reliability evaluation of logic circuits," in *Microelectronics Reliability, 48(8)*, 2008, pp. 1586–1591.

[Geiger90] D. Geiger, T. Verma, and J. Pearl, "Identifying independence in Bayesian networks," in *Networks*, 1990, pp. 507–534.

[Gibbs12] A. L. Gibbs, and F. E. Su, "On choosing and bounding probability metrics", in *International Statistical Review* 70(3), 2012, pp. 419–435.

[Gilks96] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in practice*. Chapman and Hall, 1996.

[Gredilla11] M. Lazaro-Gredilla, and M. K. Titsias, "Variational heteroscedatic Gaussian process regression," in *Proceedings of the International Conference on Machine Learning*, 2011.

[Han02] J. Han, and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," in *IEEE Transactions on Nanotechnology, 1(4)*, 2002, pp. 201–208.

[Han05] J. Han, E. R. Taylor, J. B. Gao, and J. A. B. Fortes, "Faults, error bounds and reliability of nanoelectronic circuits," in *IEEE International Conference on Application-Specific Systems, Architecture Processors, 51*, 2005, pp. 247–253.

[Han11] J. Han, H. Chen, E. Boykin, and J. A. B. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," in *Microelectronics Reliability, 51*, 2011, pp. 468–476.

[Hoffman13] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," in *Journal of Machine Learning Research 14*, 2013, pp. 3235–3268.

[Honkela10] A. Honkela, T. Raiko, M. Kuusela, M. Tornio and J. Karhunen, "Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes," in *Journal of Machine Learning Research 11*, 2010, pp. 3235–3268.

[Horton02] G. Horton, "A new paradigm for the numerical simulation of stochastic Petri nets with general firing times," in *Proceedings of the European Simulation Symposium (ESS)*, 2002.

[Ibrahim11] W. Ibrahim and V. Beiu, "Using Bayesian networks to accurately calculate the reliability of Complementary Metal Oxide Semiconductor gates," in *IEEE Transactions on Reliability, 60(3)*, 2011, pp. 47–50.

[i-RISC/D3.1] FP7-ICT/FET-OPEN/ i-RISC project, Deliverable 3.1, "Fault tolerant LDPC encoding and decoding", January 2014.

[Kameni13-a] C.L. Kameni Ngassa, V. Savin, D. Declercq, "Analysis of Min-Sum based Decoders Implemented on Noisy Hardware", *Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, USA, November 2013.

[Kameni13-b] C.L. Kameni Ngassa, V. Savin, D. Declercq, "Min-Sum-based decoders running on noisy hardware," *IEEE Global Communications Conference (GLOBECOM)*, Atlanta, GA, USA, December 2013.

[King06] N. King, and N. D. Lawrence, "Fast variational inference for Gaussian process models through KL correction," in *17th European Conference on Machine Learning*, 2006, pp. 270–281.

[Ko01] S.B. Ko, T. Xia, and J.C. Lo, "Efficient Error Prediction in FPGA," in *IEEE Int'l Symposium on Defect and Fault Tolerance in VLSI Systems*, Oct. 2001, pp. 176–181.

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

[Krishnaswamy05] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Ac- curate reliability evaluation and enhancement via probabilistic transfer matrices." in *Proceedings of the Design, Automation and Test in Europe*, 2005, pp. 282–287.

[Kuusela09] M. Kuusela, T. Raiko, A. Honkela, and J. Karhunen, "A gradient-based algorithm competitive with variational Bayes EM for mixture of Gaussians," in *Proceedings of International Joint Conference on Neural Networks*, 2009, pp. 1688–1695.

[Lee59] C. Y. Lee. "Representation of Switching Circuits by Binary-Decision Programs". Bell Systems Technical Journal, 38:985–999, 1959.

[Levin64] V. Levin, "Probability analysis of combination systems and their reliability." in *Engeering Cybernetics, 6*, 1964, pp. 78–84.

[Lindgren01] M. T. P. Lindgren, M. Kerttu and R. Drechsler, "Low power optimisation technique for BDD mapped circuits", ASP-DAC, 2001, pp. 615–621

[Mahdavi09] S.J. Seyyed Mahdavi, and K. Mohammadi, "SCRAP: sequential circuits reliability analysis program." in *Microelectronics Reliability, 49(8)*, 2009, pp. 924–933.

[Manich96] S. Manich , M. Nicolaidis and J. Figueras "Enhancing Realistic Fault Secureness in Parity Prediction Array Arithematic Operators by IDDQ Monitoring", *IEEE VLSI Test, Symposium*, 1996.

[Machado12] Machado, L.; Martins, M.; Callegaro, V.; Ribas, R.P.; Reis, A.I., "KL-cut based digital circuit remapping," NORCHIP, 2012 , vol., no., pp.1,4, 12-13 Nov. 2012

[Meinel98]Ch. Meinel, T. Theobald, "Algorithms and Data Structures in VLSI-Design: OBDD – Foundations and Applications", Springer-Verlag, Berlin, Heidelberg, New York, 1998.

[**Mehrotra13**] Rashmi Mehrotra, "Systematic Delay-driven Power Optimisation and Power-driven Delay Optimisation of Combinational Circuits", PhD Thesis 2013, University College Cork, cora.ucc.ie.

[Minami02] M. Minami, and S. Eguchi, "Robust blind source separation by beta-divergence", in *Neural Computation* 14(8), 2002, pp. 1859–1886.

[Mishchenko06-a] A. Mishchenko, S. Chatterjee and R. Brayton, "Dag-aware AIG rewriting a fresh look at combinational logic synthesis", *Proceedings of the 43rd annual conference on Design automation*, 2006, pp. 532–535.

[Mishchenko06-b] A. Mishchenko and R. K. Brayton, "Scalable logic synthesis using a simple circuit structure," in Proc. IWLS, pp. 15–22, 2006.

[Mischchenko13] A. Mishchenko, N. Een, R. Brayton, M. Case, P. Chauhan, and N. Sharma, "A semi-canonical form for sequential AIGs", Proc. DATE'13, pp. 797-802.

[Mohanram 09] M. Choudhury and K. Mohanram, "Reliability analysis of logic circuits," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 28, no. 3, pp. 392–405, 2009.

[Mohanram03] K. Mohanram , E. S. Sogomonyan , M. Gossel and N. A. Touba Synthesis of low-cost parity-based partially self-checking circuits, Proc. IEEE On-Line Testing Symp., pp.35 -40 2003

[Molnar05] S. Lazarova-Molnar, "The proxel-based method: Formalisation, analysis and applications," in *Ph.D. dissertation, Otto-von-Guericke University of Magdeburg, Germany*, 2005.

[Nocedal06] J. Nocedal, and S. J. Wright, *Numerical Optimization*. 2nd ed, Springer, 2006.

[Palem12] K. Palem and A. Lingamneni, "What to do about the end of moore's law, probably!," in Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE, pp. 924–929, 2012.

[Patel03] K. Patel, I. Markov, and J. Hayes, "Evaluating circuit reliability under probabilistic gate-level fault models." in *Proceedings of the International Workshop on Logic Synthesis*, 2003, pp. 59–64.

[Pearl88] J. Pearl, *Probabilistic reasoning in intelligent systems: network of plausible inference*. Morgan Kaufmann Publishers, Inc., 1988.

[Pecht09] M. Pecht, *Product Reliability, Maintainability, and Supportability Handbook*, 2nd edition, CRC Press, 2009.

[Pedram96] S. Iman and M. Pedram, "Pose: power optimization and synthesis environment," in Design Automation Conference Proceedings 1996, 33rd, pp. 21–26, 1996.

[Pippenger89] N. Pippenger, "Invariance of Complexity Measures for Networks with Unreliable Gates," *J. Assoc. Comput. Mach*. 36, p. 531, 1989.

[Pippenger90] N. Pippenger, "Developments in 'The Synthesis of Reliable Organisms from Unreliable Gates," *Proceedings of Symposia in Pure Mathematics*, pp. 311-324, 1990.

[Rejimon05] T. Rejimon and S. Bhanja, "Time and space efficient method for accurate computation of error detection probabilities in VLSI circuits," in *IEE Proceedings on Computers and Digital Techniques, 152(5)*, 2005, pp. 679–685.

[Rejimon06] ——, "Probabilistic error model for unreliable nano-logic gates," in *6th IEEE Conference on Nanotechnology, 1*, 2006, pp. 47–50.

[Ring12] W. Ring, and B. Wirth, "Optimization methods on Riemannian manifolds and their applications to shape spaces," in *SIAM J. Optim. 22(2)*, 2012, pp. 281–285.

[Rivers04] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The case for lifetime reliability-aware microprocessors," in Annual International Symposium on Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on, pp. 276–287, 2004.

[Roelke07] G. R. Roelke, R. O. Baldwin, and D. Bulutoglu, "Analytical models for the performance of von Neumann multiplexing," in *IEEE Transactions on Nanotechnology, 6(1)*, 2007, pp. 75–89.

[Sato01] M. Sato, "Online model selection based on the variational Bayes," in *Neural Computation 13(7)*, 2001, pp. 1649–1681.

[Sato13] H. Sato, and T. Iwai, "A new, globally convergent Riemannian conjugate gradient method," 2013.

[Sentovich92] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P. Stephan, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "SIS: A system for sequential circuit synthesis", *Technical Report UCB/ERL M92/41, EECS Department, University of California, Berkeley*, 1992.

[Sogomonjan93] E. S. Sogomonjan and Michael Gössel. Design of self-parity combinational circuits for self-testing andon-line detection. In Proceedings of the IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems, pages 239-246, Washington, DC, USA, 1993. IEEE Computer Society.

[Sung08] J. Sung, Z. Ghahramani, and S. Bang, "Latent-space variational Bayes," in *Pattern Analysis and Machine Intelligence, 30(12)*, 2008, pp. 2236–2242.

[Tani93] S. Tani, K. Hamaguchi and S. Yajima, "The Complexity of the Optimal Variable Ordering Problems of A Shared Binary Decision Diagram ", Proceedings of the 4th International Symposium on Algorithms and Computation, 1993, pp. 389–398.

[Tinmaung07] K. O. Tinmaung, D. Howland and R. Tessier, "Power-Aware FPGA Logic Synthesis Using Binary Decision Diagrams", Proceedings of the 2007 ACM/SIGDA 15th international symposium on Field programmable gate arrays, NY USA, 2007, pp. 148–155.

[Taylor06] E. Taylor, J. Han, and J. Fortes, "Towards accurate and efficient reliability modeling of nanoelectronic circuits," in *6th IEEE Conference on Nanotechnology, 1*, 2006, pp. 395–398.

[Taylor68-a] M. G. Taylor, "Reliable information storage in memories designed from unreliable components", *Bell System Technical Journal*, vol. 47, pp. 2299-2337, 1968.

[Taylor68-b] M. G. Taylor, "Reliable computation in computing systems designed from unreliable components", *Bell System Technical Journal*, vol. 47, pp. 2339-2366, 1968.

[Teh07-a] Y. W. Teh, D. Newman, and M. Welling, "A collapsed variational Bayes inference algorithm for latent Dirichelet allocation," in *Advances in Neural Information Processing Systems 19*, 2007.

[Teh07-b] Y. W. Teh, K. Kurihara, and M. Welling, "Collapsed variational inference for HDP," in *Advances in Neural Information Processing Systems 19*, 2007.

[Touba97] N. A. Touba and E. J. McCluskey Logic Synthesis of Multilevel Circuits with Concurrent Error Detection, IEEE Trans. CAD, vol. 16, pp.783 -789 1997

[Ueda95] H. Ueda and K. Kinoshita, "Low power design and its testability", Proceedings of the Fourth Asian Test Symposium, India, Nov. 1995, pp. 361–366.

[Ueda99] H. Ueda and K. Kinoshita, "Power Estimation And Reduction Of CMOS Circuits Considering Gate Delay", IEICI Transactions on Information and System, Vol. E82-D, No. 1, Jan. 1999, pp. 301–308.

[Verma98] T. Verma, and J. Pearl, "Causal networks: semantics and expressiveness," in *Proceedings of the 4th Workshop on Uncertainty in AI*, 1988, pp. 352–359.

[Vrudhula06] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the nbti effect for reliable design," in Custom Integrated Circuits Conference, 2006. CICC '06. IEEE, pp. 189–192, 2006.

[Wainwright08] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," in *Foundations and Trends in Machine Learning 1, 1-2*, 2008, pp. 1–305.

[Wang13] Y. Wang, "Aging Assessment and Reliability Aware Computing Platforms", Ph.D. thesis, TU Delft, Delft, The Netherlands.

[Wight00] R. L. Wright, M. A. Shanblatt, DCS Corp and V. A. Alexandria, "Improved switching activity estimation for behavioral and gate level designs", Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000, pp. 172–175.

[Wu05] D. Wu and J. Zhu, "FBDD: A folded logic synthesis system", *In Proceedings of the International Conference on ASIC (ASICON)*, Oct. 2005, pp. 746–751.

[Yanushkevich05] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko and R. S. Stankovic, "Decision Diagram Techniques for Micro- and Nanoelectronic Design Handbook", *CRS Press*, 2006, pp. 429–445.