Defect Tolerance Estimation and Netlist Optimization for Digital Designs

Shiva S. Thiagarajan
ECE Department
University of Texas at Dallas
Dallas, TX, USA
shivashankar.thiagarajan@utdallas.edu

Suriyaprakash Natarajan
Intel Corporation
Santa Clara, CA, USA
suriyaprakash.natarajan@intel.com

Yiorgos Makris
ECE Department
University of Texas at Dallas
Dallas, TX, USA
yiorgos.makris@utdallas.edu

***Abstract*—Two important aspects of design-for-yield (DFY) are the design for functional yield and parametric yield. While occurrences of gross defects leading to functional yield loss are fixed during the manufacturing stage, small defects (delay defects) due to 'intrinsic' marginalities like parametric variations causing partial opens/shorts are yet to be fully understood and characterized. Contemporary synthesis tools are developed to optimize for power, performance and area constraints, while defect and yield considerations are pushed to process fixes and layout design optimizations. This project aims to introduce defect tolerance as an inherent objective of design synthesis and seamlessly trade it off with power, performance and area, to improve design robustness to delay defects and subsequently improve yield. Towards this objective, we propose a new defect tolerance metric to the library cell characterization along with typical power, timing and area requirements. We model the effects of small defects on cells as delay penalties, and develop defect tolerance metrics to quantify the likelihood of defect tolerance in library cells. Eventually, we plan on utilizing these defect tolerance metrics to drive the identification of vulnerable cells and rewiring/redesigning of netlists to improve delay defect tolerance.**

***Index Terms*—defect tolerance, delay defects, timing yield**

I. Introduction

As the CMOS technology scales towards the 7nm process and beyond, process variations and manufacturing defect levels are sharply increasing impacting design performance and leading to yield loss. These yield losses can generally be classified as functional yield, occurring due to catastrophic failures in chips and parametric yield, where chips fail to meet certain power or performance criteria. Defects like gross-defects (opens/shorts) which leads to functional failures, are traditionally fixed at the manufacturing stage using process fixes and design-for-manufacturability guidelines (DFMGs), small-defects which leads to delay degradation during manufacturing are yet to be characterized. Though empirical methods like DFMGs have been successful in managing gross-defects in prior technology nodes, it is taking longer to achieve the required defect coverage due to increasing complexity in process technologies as well as fail to limit the effects of small-defects (delay defects) in latest designs. These small-defects like partial shorts/opens which arise due to 'intrinsic' marginalities in design-process interactions, typically introduce delays or cause leakages which consequently affect the power and performance of a design leading to parametric yield loss.

The defects observed during manufacturing can be largely modeled as full/partial opens and shorts. The full open/short defects typically manifests itself as functional failures in circuits, while the partial defects introduces delays that can degrade the circuit performance. These defects can materialize with different sizes and location during manufacturing and can affect the circuit parameters depending on the probability of defect size/location occurrence. Based on these size/location probabilities, the defect can lead to varying degrees of opens/shorts in circuits, with egregious defects leading to functional failures. Traditional fault models like stuck-at and transition fault models abstract these defect behaviors as boolean primitives, which are effective to diagnose/test functional faults, but small delay timing variations occurring due to partial opens/shorts are not effectively captured or modeled by this approach.

Many of the defect analysis methodologies, fault modeling techniques and cell simulation applications are adapted to avoid defects during manufacturing or for enhancing test pattern generation to identify these defects and filter out the defective chips. Apart from this, modular redundancy methods are commonly used to mitigate the effects of defective cells in a design thereby adversely affecting the design area/power requirements. Prior research largely concerns cell-level netlist analysis and cell-library simulations for test pattern generation [1], fault/defect coverage for delay faults [2] and reliability analysis for functional faults [3]. Redundancy and re-configurability of defective cells is proposed in [4] and delay optimization of a logic network post technology mapping for FPGAs is proposed in [5].

Most of these approaches pursue a defect-avoidance strategy by identifying and testing for these defects, introducing redundancy for re-configuration purposes and analyzing netlists for

This work was supported by Semiconductor Research Corporation (SRC).

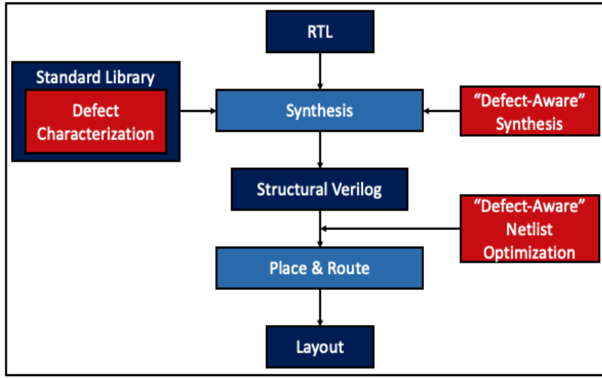

Fig. 1. Defect Tolerant Design Overview

functional reliability. This work proposes a defect-tolerance strategy at the design synthesis stage for mitigating the effects of small delay defects (SDDs) on circuit timing. Defect tolerance estimation and Netlist design is motivated by the fact that existing design automation (EDA) tools are developed to optimize for power, performance and area objectives, but are not adapted to include defect considerations in designing netlists. It is to be noted that the student author of this paper was privy to a recent publication in 2020 VLSI Test Symposium (VTS) [6] during his internship, and this work is an adaptation of this paper in academic setting for further research.

The paper is organized as follows. Section 2 provides the overview of the defect tolerant design flow. Section 3 describes the re-characterization methodology of library cells for delay defects. Section 4 describes the netlist gate ranking flow and the use of library recharacterization in identifying vulnerable gates. Section 5 is the conclusion.

II. Overview

In principle, the defect tolerant approach towards designing netlists has two primary objectives: 1) enhance the cell-library database with defectivity information thereby ensuring design synthesis uses those cells which are better at tolerating defects, and 2) identify vulnerable cells in the netlists mainly targeting those on critical paths and rewire/redesign those cells locally in the netlist to improve the defect tolerance on those paths. In Fig.1. the generic idea of a defect tolerance netlist design methodology is illustrated. The flow in the middle is the generic RTL2GDS flow which is implemented by many proprietary and open-source EDA tools which optimizes for power, performance and area constraints. This project tries to intercept this flow by essentially introducing a delay defect tolerance constraint along with the PPA requirements, by re-characterizing the library for delay defects, modeling the delay responses into a probabilistic metric which drives the synthesis of a netlist and/or rewiring/structurally transforming the vulnerable cells post-structural design.

III. Library Recharacterization

The first building block of the design is the cell library which are characterized for the power, performance and area. This characterized cell information database is utilized by EDA tools to synthesize netlists which satisfy the PPA requirements of the design. This work proposes to introduce a new dimension to library cell characterization by implementing a timing-delay simulation of the defects and modeling this defect delay information as probabilities to quantify the defect tolerance of these cells. This modeled defect information will be further utilized to identify the vulnerable instances in a design thereby driving local rewiring/resynthesis.

To develop these defect tolerance metrics and cell defect probabilities, we leverage a standard cell library characterization technique called Cell-Aware Test Methodology [1] which is applied for scan test generation purposes. The first step in library re-characterization for defect tolerance is to identify the nets which are candidates for open/short defects in the cell netlist. This selection is of nets is based on the analysis of the extracted parasitics from the cell layout. Based on the analogy of high resistance of a net for opens and low resistance between nets for shorts, the potential locations for defect simulations are selected. The second step is to injected these candidate locations with a pre-set resistor values with very high resistances for open-circuit, and very low resistances between nets for short-circuit. This pre-set defect values (sizes) are modeled from $10\,\Omega$ to $100\,000\,\Omega$ varying from short to open, respectively. The defect model essentially consists of the defect location and size pairs, which are injected in the cell netlist one at a time, to perform spice simulations for timing delays. In cell-aware methodology, circuit simulations are performed on these defect injected netlists and those cell input stimuli which results in functional errors, are tabled to create User-Defined-Fault-Models (UDFMs). Subsequently, these UDFMs are then used with Automatic Test Pattern Generation (ATPG) tools to generate scan patterns for these cell instances in the netlists. We leverage this functional test pattern generation work to determine the cell vulnerability by performing timing simulations instead of functional circuit simulations.

For timing simulations, we implement dynamic 2-time frame (2-T.F) simulations with transitioning cell input patterns for a specific load-slope combination. It is true that the simulation must be performed for the exhaustive combinations of cell inputs, output loads and input slews, this becomes computationally intensive as the required number of simulations per cell is given by,
$Number of simulations = 2^{2*n} * |InputSlew| * |OutputLoad| * |DefectSizes| * |DefectLocations|$
where n is the number of inputs to the gate.

The timing simulation of defect injected cells, yield a defect delay distribution of the cell response with manifested defects.

These delay responses are modeled into a probability mass function (PMF) called delay defect distribution (D^3) for every cell in the library which reflects the cell timing behavior under the small delay defects (SDDs). These delay defect distributions can be thresholded based on a reference delay value, thereby generating a probabilistic metric to define the defect vulnerability of the cell. This metric can be used to derate the timing delays of library cells, which in turn can steer the synthesis to towards designing defect tolerant netlist. This application of the metric is extensively studied in paper [6]. Apart from biasing the synthesis tool, this work also tries to explore the the relevance of the delay defect distribution (D^3) to the cell instances in the netlists. This is further explained in section IV.

Starting with the test case of 65nm library containing 60 cells of combinational and sequential cells of various drive strengths, we extracted post-layout parasitics for the purpose of choosing candidate nets for defect injection. For a range of specific defect sizes, open (high resistances) defects on same nets and short (low resistances) on different nets were injected on the spice netlist. For a specific load-slope combination, exhaustive 2 T.F. input patterns were simulated in spice for obtaining the delay response for every cell in the library. These delay responses were modeled into delay defect distribution (D^3) which will be used along with timing information from the design netlist for identifying vulnerable gates.

IV. Netlist Defectivity Analysis

In netlist defectivity analysis, the primary objective is to rank the design netlist gates based on their vulnerability to defects. One underlying assumption in this analysis is that the vulnerability of the cell instances in a netlist will depend on 1) the presence of a cell on a critical path, 2) number of input patterns that activates the cell, 3) local connectivity with other cells in the netlist and finally, 4) the variability observed in cell delay due to defects. For extracting and processing the design netlist information, we obtain the structural netlist and the related timing information by querying the timing database (PrimeTime). The cell delay variability due to defects was modeled into a probability distribution in the library recharacterization step, which will be used with the delay timings of the cell instances in the netlist to identify and rank vulnerable gates.

There are a few approaches to rank the gates in a netlist. A straightforward approach is to generate the average delay-slack of paths passing though a cell instance in the netlist. This average slack (across paths) of every cell is used to threshold the delay defect distribution (D^3) obtained from library recharacterization. The number of defective cases which falls above this threshold in the distribution is used to define a probabilistic metric of defect tolerance for every cell instance in the netlist. This is illustrated in Fig.2. for nand-based adder implementation. These gate susceptibility values of the gates

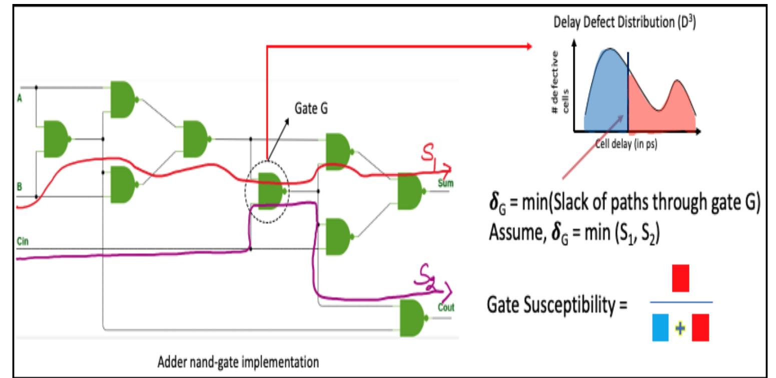

Fig. 2. Netlist Gate Susceptibility

can be used to rank the gates in the netlist and can further drive local structural rewiring based on prior works like [7] which is based on logic flexibility.

Though the above methodology is a crude way of representing the cell vulnerability, we explore a more refined approach to gate ranking. The test case we used for this experiment is a ITC99-B01 benchmark circuit which is a FSM that compares serial flows. The design was functionally validated using a Verilog test bench, and synthesized with the 65nm reference library, which was recharacterized for delay defects in the last step. The resulting place and route Verilog netlist was imported to PrimeTime for timing validation and for extracting the required timing information for all timing arcs. We primarily inquire for 2 types of timing information, the net/cell arc timings and the delay-slack timings of the paths in the netlist. The net/cell arc timing were utilized for generating a graph model of the netlist which will be used for logic simulation to determine the fraction of input patterns activating the defect susceptible cell as well as propagating the defect to primary outputs, and for extracting the local connectivity of the cell instances for rewiring. The delay-slack path timings are processed to obtain the average slack of paths passing though every cell instance in the netlist. Essentially, this average slack is the buffer in terms of delay variability every cell can operate without violating timing constraints. Currently, we are investigating these timing information along with the delay variabilities from library recharacterization to develop a refined metric for quantifying the defect tolerance of netlist cell instances.

V. Conclusion

In addition to the power, performance and area objectives of design synthesis, this work advocates the necessity of defect tolerance approach to design netlists. Designing circuits that are robust to defects are mostly pushed to the manufacturing stage with process fixes and DFMGs, but this project pushes for defect tolerance to be an inherent objective in structural netlist design. We investigated the library cell delay timings

under delay defects and modeled the delays into probabilistic distributions for characterizing the delay variability of the cells. We further established a simple methodology to explore the effects of defect induced delay variability to rank cell instances in the netlist. Other potential applications of defect tolerance estimation, (i) diagnosis of manufactured circuits to identify the failing gates, and (ii) driving efficient test content generation for timing faults, etc. are area for further research.

ACKNOWLEDGMENT

We thank Vibhav Kumarswami Salimath for providing the library database and Steve Martindell for pointers to characterization tools.

REFERENCES

[1] F. Hapke, R. Krenz-Baath, A. Glowatz, J. Schloeffel, H. Hashempour, S. Eichenberger, C. Hora, and D. Adolfsson, "Defect-oriented cell-aware atpg and fault simulation for industrial cell libraries and designs," in *2009 International Test Conference*, 2009, pp. 1–10.

[2] Zhuo Li, Xiang Lu, Wangqi Qiu, Weiping Shi, and D. M. H. Walker, "A circuit level fault model for resistive opens and bridges," in *Proceedings. 21st VLSI Test Symposium, 2003.*, 2003, pp. 379–384.

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

[4] J. Tang, D. Wang, Y. Polyanskiy, and G. W. Wornell, "Defect tolerance: Fundamental limits and examples," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5240–5260, 2018.

[5] A. Mishchenko, R. Brayton, and S. Jang, "Global delay optimization using structural choices," in *Proceedings of the 18th Annual ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 181–184. [Online]. Available: https://doi.org/10.1145/1723112.1723144

[6] S. Natarajan, A. F. Malavasi, and P. A. Meinerzhagen, "Automated design for yield through defect tolerance," in *2020 IEEE 38th VLSI Test Symposium (VTS)*, 2020, pp. 1–6.

[7] S. Almukhaizim, Y. Makris, Y. Yang, and A. Veneris, "On the minimization of potential transient errors and ser in logic circuits using spfd," in *2008 14th IEEE International On-Line Testing Symposium*, 2008, pp. 123–128.