ReSpace/MAPLD Conference
Albuquerque, NM, August 2011.

A Fault-Handling Methodology by Promoting Hardware Configurations via PageRank

Naveed Imran and Ronald F. DeMara
Department of Electrical Engineering and Computer Science,
University of Central Florida
E-mail: naveed@knights.ucf.edu, demara@mail.ucf.edu

Agenda

- **Motivation**
- **Fault-Tolerance of FPGA-based Designs** ***(previous approaches)***
- **Fault-handling by promoting hardware configurations** ***(current work)***
- **Simulation Results**
- **Conclusions**

Motivation

Progress in Semiconductor Technology

- Reduced feature size -> increasing vulnerability to errors, aging-induced permanent faults *
- Runtime faults in addition to manufacture-time defects of future nanoscale devices **

Reliability

- Importance of fault tolerance in mission critical applications

FPGAs

- FPGA as a reconfigurable platform
- Faults in logic resources and configuration memory

* Suresh Srinivasan, Krishnan Ramakrishnan, Prasanth Mangalagiri, Yuan Xie, Vijay krishnan Narayanan, Mary Jane Irwin, Karthik Sarpatwari, "Towards Increasing FPGA Lifetime", IEEE Trans. Dependable and Secure Computing. vol. 5, Issue 2, pp. 115-127, 2008.
** W. Rao, C. Yang, R. Karri, and A. Orailoglu, “Toward future systems with nanoscale devices: Overcoming the reliability challenge,” *Computer,* vol. 44, no. 2, pp. 46 –53, Feb. 2011.

Fault-Tolerance of FPGA-based Designs

- **Redundancy-based techniques**

  Example: Duplex arrangement, Triple Modular Redundancy (TMR)

- **Online Testing methods [3],[4],[5]**

  Example: Online Built-In Self Test (BIST), Roving STAR

- **Evolvable Hardware**

  Example: Design-time evolution [6], Competitive Runtime Reconfiguration (CRR) [10]

Fault-Tolerance of FPGA-based Designs (Contd.)

- **Configuration-level method**

  Avoid the faulty resources by manipulating the configuration bitstream [8].
- **Multiple Configurations method**

  Generating many alternate configurations of the circuit at design time and reloading a particular configuration at run time when confronted with a fault [9].
- **Multiple Configurations with Refurbishment Scheme**

  Generate a diverse population of circuit configurations utilizing alternative device resources and refurbish via *Consensus Based Evolution* [7].

Fault-handling by promoting hardware configurations

At Design-Time:

- Generate a diverse set of functionally identical configurations utilizing alternate hardware resources in an FPGA, at *design-time*.

At Runtime:

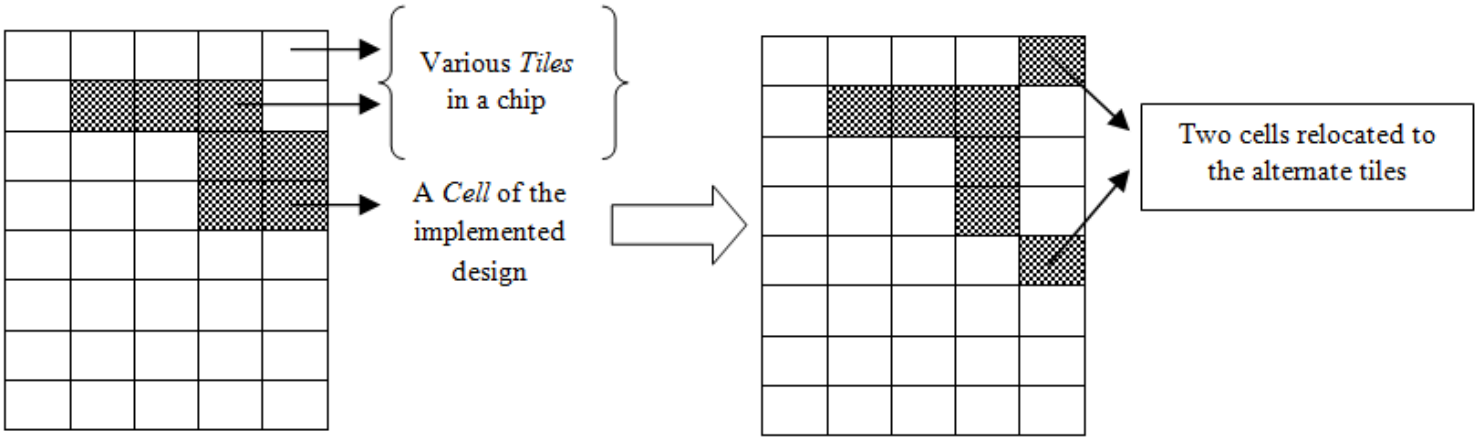
- These hardware-realizations are evaluated online by *actual inputs* of the circuit.
- The *Circuits Similarity Graph* (CSG)
- The pool is sorted using the *PageRank* algorithm, currently utilized by the Internet indexing procedures [1]

Fault Recovery:

- The configurations having high rank are instantiated.

Generating Diverse Configurations at Design-time

- Partition the circuit into various *cells*, the chip into *tiles*.
- Each cell contains multiple LUTs
- An initial location of the cell can be obtained from the post place-and-route circuit model -> The seed design configuration
- The cells are relocated to different tiles in the chip to generate an alternate configuration.
- A diverse set of configurations is generated at design time utilizing alternate resources.



Circuit-Similarity Graph

- The *CSG* is a graph G = (V,E,W), where V is the vertex set, E is the set of edges and W is the associated weight adjacency matrix.
- Each *node* (vertex) represents a particular circuit realization and the *edge* between nodes represents the similarity between circuits in terms of their output.
- For constructing the weight adjacency matrix W, for each entry the corresponding pair of the configurations are evaluated in a *duplex* manner.
- If a realized circuit's output is consistently matched with the other circuits, we consider it more important than others and want to assign it a higher *rank*.
- Apply the *PageRank* algorithm to compute the rank score of each node in the graph. [1],[2],[11],[12]

PageRank Algorithm

- S. Brin and L. Page developed the PageRank algorithm to rank the web pages on world wide web according to their importance.
- Successfully being employed by the Google search engine

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{c(T_1)} + ... + \frac{PR(T_n)}{c(T_n)})$$

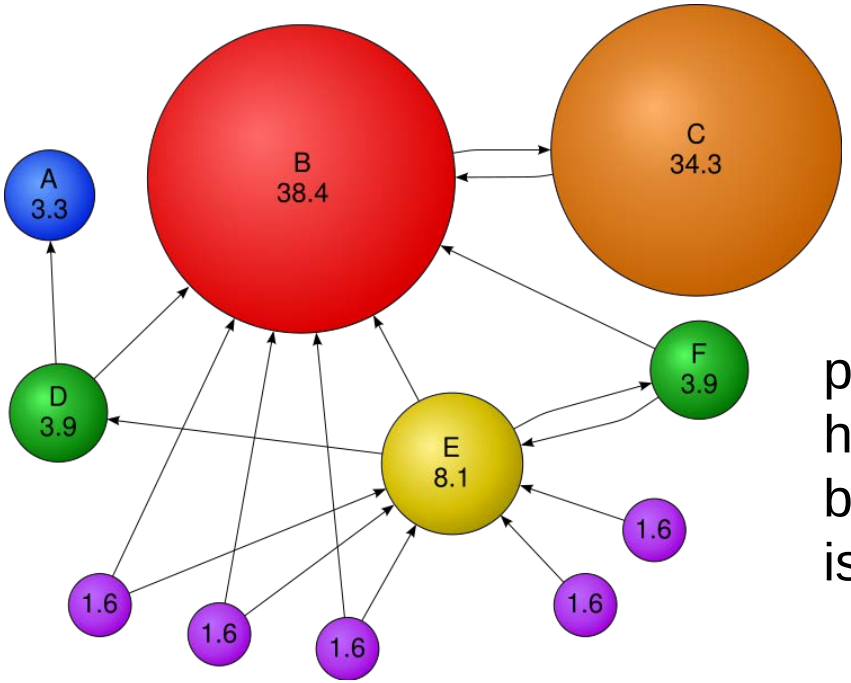
where $PR(A)$ = PageRank of a page A
$T_1 ... T_n$ = The pages which refer to page A
$c(A)$ = number of links going out of page A

S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107–117, 1998.

PageRank Algorithm (contd..)

- The importance of a webpage is based on a factor determined by the number of references made to it by other pages.
- Example:

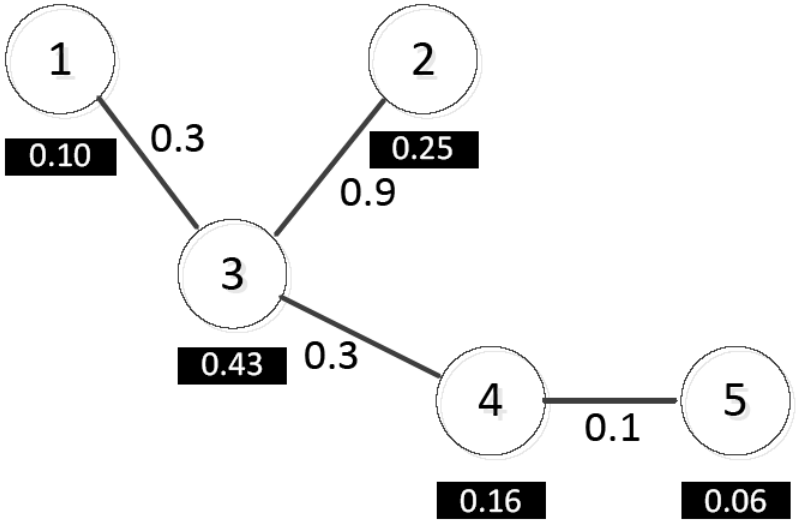
Source - Wikipedia http://en.wikipedia.org/wiki/File:PageRanks-Example.svg



probability of hitting the page C by random surfing is 34.3%

PageRank Algorithm (contd..)

1.0	0.0	0.3	0.0	0.0
0.0	1.0	0.9	0.0	0.0
0.3	0.9	1.0	0.3	0.0
0.0	0.0	0.3	1.0	0.1
0.0	0.0	0.0	0.1	1.0



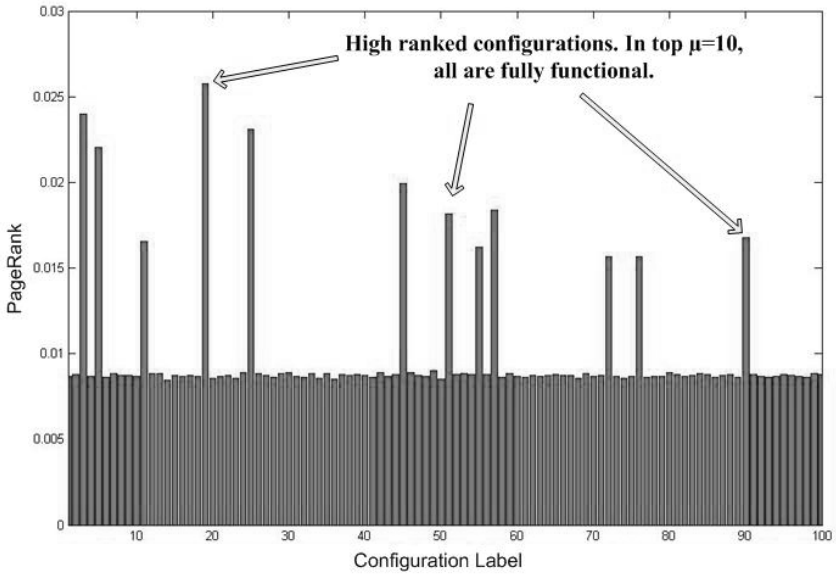
Each element of the matrix corresponds to the discrepancy between a configurations pair

Simulation Setup

- Verilog HDL description of the MCNC benchmark circuits in Xilinx ISE
- Generated 100 alternate configurations for each benchmark circuit at design time.
- Then faults were randomly injected into the chip model and, for the `z4ml` circuit -> 86 circuit configurations affected thereby only leaving 14 designs fully functional.
- The CSG is built by evaluating a pair of circuits to a subset of random inputs of size 30.
- The normalized inverse of the difference in outputs is taken as a similarity measure.
- A sparse CSG instead of evaluating all exhaustive pairs

Simulation Results

- The PageRank value of each circuit implementation identified by its configuration label
- Higher score configurations -> utilize fault free resources
- Preferred configurations can be readily differentiated using this technique.



PageRank of different circuit configurations

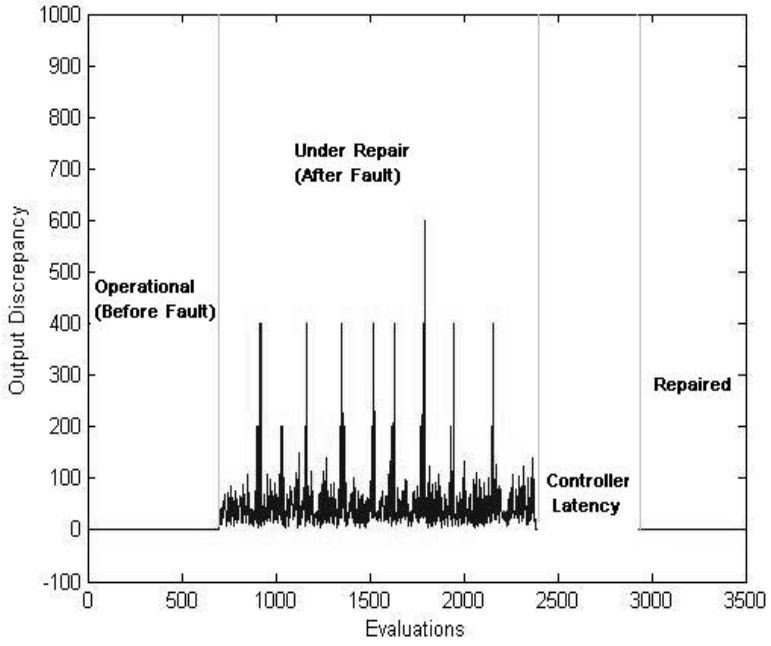
Performance Comparison

TABLE I
COMPARISON BETWEEN EXHAUSTIVE TESTING AND THE PROPOSED APPROACH FOR THE BENCHMARK CIRCUITS

Description	z4ml	cm85a
No. of inputs	7	11
No. of outputs	4	3
Total input set size	128	2048
Number of configurations generated at design time	100	100
All possible pairs set size	$\binom{100}{2}/2$= 4950	4950
Total exhaustive inputs required to test all circuit pairs	633600	1.01376e+7
Number of pairs evaluated in the method	180	180
Number of inputs applied to evaluate each pair	30	200
The number of evaluations used in the method	5400	36000
Total number of configurations promoted correctly	85%	86%
In top $\mu = 10$ identified configurations, the number of correct ones	10 (i.e. **100%**)	10 (i.e. **100%**)
Improvement in terms of input evaluations	**117 fold**	**282 fold**

A Simulation of the circuit operation

- The cumulative value of the absolute difference between outputs of the duplex circuit in each evaluation window.
- Controller latency will depend on circuit operational frequency
- Adaptive identification of preferred configurations without additional test under actual runtime conditions.



References

1. S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107–117, 1998.
2. N. Imran, J. Liu, J. Luo, and M. Shah, “Event Recognition from Photo Collections via PageRank”, presented at the Proceedings of the 17th ACM international conference on Multimedia, Beijing, China, 2009.
3. J. Emmert, C. Stroud, and M. Abramovici, “Online fault tolerance for FPGA logic blocks,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 15, no. 2, pp. 216 –226, Feb. 2007.*
4. S. Dutt, V. Verma, and V. Suthar, “Built-in-self-test of FPGAs with provable diagnosabilities and high diagnostic coverage with application to online testing,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 27, no. 2, pp. 309 –326, Feb. 2008.*
5. M. Gericota, G. Alves, M. Silva, and J. Ferreira, “Reliability and availability in reconfigurable computing: A basis for a common solution,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on,* vol. 16, no. 11, pp. 1545 – 1558, Nov. 2008.
6. D. Keymeulen, R. S. Zebulum, Y. Jin, and A. Stoica, “Fault-tolerant evolvable hardware using field-programmable transistor arrays,” *Reliability, IEEE Transactions on, vol. 49, no. 3, pp. 305–316, 2000.*
7. R. F. DeMara, K. Zhang, and C. A. Sharma, “Autonomic fault-handling and refurbishment using throughput-driven assessment,” *Appl. Soft Comput., vol. 11, pp. 1588–1599, March 2011.*
8. M. Garvie and A. Thompson, “Scrubbing away transients and jiggling around the permanent: long survival of FPGA systems through evolutionary self-repair,” in On-Line Testing Symposium, 2004. IOLTS 2004. Proceedings. 10th IEEE International, 2004, pp. 155–160.
9. J. Lach, W. Smith, M. Potkonjak, “Low overhead fault-tolerant FPGA systems,” IEEE Trans. Very Large Scale Integr. Syst., vol. 6, no. 2, pp. 212–221.
10. R. F. DeMara and Z. Kening, "Autonomous FPGA fault handling through competitive runtime reconfiguration," in *Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on,* 2005, pp. 109-116*.*
11. J. Yushi and S. Baluja, "VisualRank: Applying PageRank to Large-Scale Image Search," *Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, pp. 1877-1890, 2008.*
12. D. Gleich, “ pagerank”, MathWorks.com, 2006*. [Available] http://www.stanford.edu/~dgleich/programs-old.html*