

Defect Tolerance After the Roadmap

Mahim Mishra* and Seth C. Goldstein
Computer Science Department
School of Computer Science
Carnegie Mellon University

1 Introduction

Optical photolithography techniques are approaching physical and economic limits that will drastically reduce the current scaling rate of device miniaturization. New technologies being investigated include Next Generation Lithography (NGL) and Chemically Assembled Electronic Nanotechnology (CAEN), which hold the promise of extremely high densities and sub-10nm feature sizes. However, these technologies are likely to have significantly higher defect densities than current ones. This is especially true for CAEN-based devices: we expect the very nature of chemical fabrication to result in defect densities of as much as 10%. Such high defect densities require a completely new approach to manufacturing computational devices: since every chip is expected to have multiple defects, it will no longer be possible to test them and throw defective ones away. Instead, we will have to devise a way to use defective chips.

A natural solution is suggested by reconfigurable fabrics, i.e., Field-Programmable Gate Arrays (FPGAs). The key idea is that reconfigurability allows one to find the defects and then to avoid them. Reconfigurable fabrics made from next-generation technologies will be tested by configuring them for self-diagnosis. This generates a map of all the chip's defects. Compilers generating circuits for a fabric will use the defect map to route around the defects. The manufacturing process will be simplified because complex computational structures or highly accurate, defect-free features will not be needed. In some sense, this introduces a new manufacturing paradigm, one which trades-off complexity at manufacturing time with post-fabrication programming. The reduction in manufacturing time complexity makes reconfigurable fabrics a particularly attractive architecture for CAEN-based fabrics, since directed self-assembly will most easily result in highly regular, homogeneous structures.

Our contribution is a testing method to find the defects in a reconfigurable fabric with a high defect density, which scales with defect rate and fabric size. We present the theory underlying our method, and candidate circuit implementations for carrying out the tests.

2 Related Work

Similar defect tolerance issues have been dealt with in custom computing systems (e.g. [1, 2, 3]). The Teramac custom computer ([4, 1]) is the most notable example: upto 75% of the FPGAs used in the Teramac were defective. Assembly was followed by a testing phase where the defects in the FPGAs were identified and mapped. Compilers for generating FPGA configurations used this defect map to avoid the defects. Our proposed testing strategy is similar to the one used for the Teramac. However, the problem we address is significantly harder because the Teramac (and other such systems) used CMOS devices whose defect rates are significantly lower.

Our testing and analysis techniques have resonances with a large body of work in Statistics and Information Theory on *Group Testing* ([5]), which is a collection of techniques for finding members of a population which satisfy a particular property (in other words, which are “defective”). Our

*Author to whom correspondence should be directed: Ph: (412) 268-3562, Fax: (412) 268-4801, E-mail: mahim@cs.cmu.edu

work is based on certain aspects of *non-adaptive, probabilistic group testing*. However, none of the problems discussed in the group testing literature have constraints as hard as ours: they have lower defect rates and allow a smaller granularity of access to population members than is possible here.

Modern DRAM and SRAM chips and FPGAs can tolerate some defects by having redundancy built into them, such as an extra row of memory cells. This is generally not possible for logic; besides, our defect rates are so high that finding even one defect-free row may not be possible.

3 Proposed Testing Method

Our approach consists of configuring the fabric components¹ into circuits which test themselves. Since test circuits are configured using resources which are later used during normal fabric operation, testing incurs no area and delay penalty. Each component is made a part of many different test circuits, and information about the error status of each of those circuits is collected. This information is used to deduce and confirm the exact location of the defects. Since we are unlikely to have fine-grained access to fabric components, our test circuits will be large, consisting of tens and perhaps even hundreds of components. With high defect rates, test circuits which only tell us if they are defective or not will be useless: almost each and every test circuit will have at least one defective component. The key idea to our approach is to use more powerful test circuits; i.e., circuits that return more than binary information about the presence of defects in their components. One example would be circuits that report the presence of none, some or many defects.

We propose splitting the process of defect-mapping into two phases: a *probability-assignment phase* and a *defect location phase*. The probability-assignment phase attempts to separate the components in the fabric into two groups: those that are “probably good” and those that are “probably bad”. This is done by configuring multiple test circuits on the fabric, and using Bayesian analysis to calculate posterior probabilities for each component being defective based on the results of test circuits that component was a part of. The components identified as “probably bad” in the probability-assignment phase are discarded; those identified as “probably good” will have an expected defect density that is low enough so that in the defect-location phase, we can use circuits that return 0-1 information about the presence of defects to pin-point them.

This method of defect-mapping should produce no false negatives (bad components identified as good), given test circuits which can detect all the modelled defects. However, there may be a significant number of false positives. This number will depend on the type of test-circuits used, number of tests run and rigorousness of the post-testing analysis. Another important quality of the method is that test-circuit generation is largely oblivious: the results of previous tests are not used to generate new circuits except between the probability-assignment and defect-location phases.

4 Evaluation

We have performed simulations using different test-circuits to gauge the effectiveness of this method. The quality criterion we use is the *recovery* of the process: the proportion of good components on the fabric that our testing procedure identified as such (recall that no bad components are identified as good). Figure 1 shows recovery using two different kinds of circuits:

- “Counter” circuits, which can count the number of defective components upto a certain threshold. These circuits, although powerful, are probably impossible to realize practically. As expected, our results improve as the number of defects that can be counted increases.

¹We are deliberately leaving the meaning of “component” unspecified. It will depend on the final design of the fabric: a component may be one or more simple logic gates, or a look-up table implementing an arbitrary logic function; also, the on-fabric interconnects will also be “components” in the sense that they may also be defective.

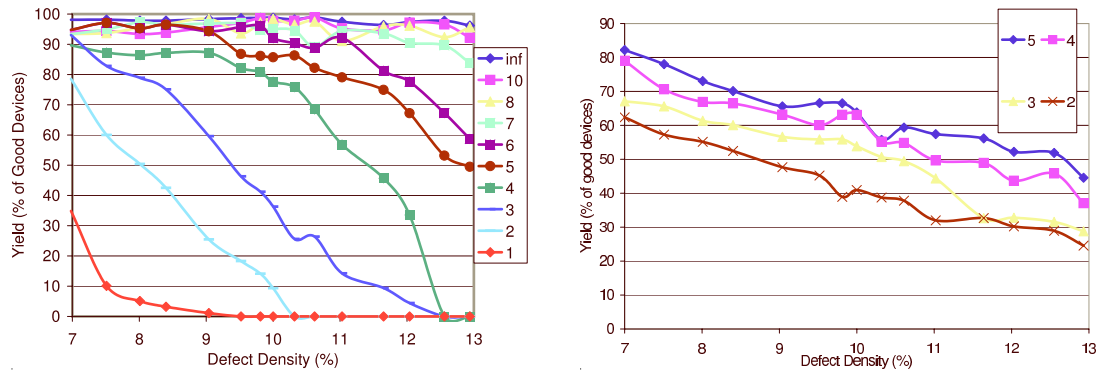


Figure 1: Yields for 2 different types of test circuits. **Left:** counter circuits; each line is a circuit that has a different upper limit for the number of defects it can count. **Right:** LFSR-based circuits that can say if there were none, some or many defects in the circuit components; circuits represented by smaller-numbered lines are easier to implement but give less accurate information.

- Circuits which can return information on whether there were none, some or many defects. Such circuits can be simulated by reconfiguring the same components with circuits of varying robustness. For example, the fabric components can be configured into an LFSR which can then be split into two or more smaller LFSRs. The number of LFSRs which return an erroneous result can then tell us if there were none, some or many defects. We have also looked at circuits based on cellular automata; however, they require too many resources to implement and so do not give a high level of defect resolution.

These results show that significant recovery is achievable in the presence of high defect rates by using relatively simple and quick testing techniques. We are currently evaluating other types of test circuits, as well as methods of analyzing circuit results other than Bayesian analysis.

5 Conclusion

We believe this is a promising approach to defect tolerance in reconfigurable fabrics made of nanometer scale devices. Our initial work has shown encouraging results; we are now working on enhancements which would use information gained from the techniques discussed here to obtain close to 100% recovery.

References

- [1] B. Culbertson, R. Amerson, R. Carter, P. Kuekes, and G. Snider, "Defect tolerance on the teramac custom computer," in *Proceedings of the 1997 IEEE Symposium on FPGA's for Custom Computing Machines (FCCM '97)*, (Napa Valley, CA), April 16-18 1997.
- [2] J. Emmert, C. Stroud, B. Skaggs, and M. Abramovici, "Dynamic fault tolerance in fpgas via partial reconfiguration," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2000)*, (Napa Valley, CA), pp. 165–174, Apr. 2000.
- [3] S. K. Sinha, P. M. Karmachik, and S. C. Goldstein, "Tunable fault tolerance for runtime reconfigurable architectures," in *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM 2000)*, (Napa Valley, CA), pp. 185–192, Apr. 2000.
- [4] J. R. Heath, P. J. Kuekes, G. S. Snider, and R. S. Williams, "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology," *Science*, vol. 280, pp. 1716–1721, June 12 1998.
- [5] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and its Applications*, vol. 12 of *Series on Applied Mathematics*. New York: World Scientific, second ed., 2000.
- [6] M. Mishra and S. C. Goldstein, "Scalable defect tolerance for molecular electronics," in *Proceedings of the 1st Workshop on Non-Silicon Computing (NSC-1)*, 8th International Symposium on High-Performance Computer Architecture, (Cambridge, MA), 2002.