

A Hybrid Framework for Design and Analysis of Fault-Tolerant Architectures

Debayan Bhaduri Sandeep Shukla
Virginia Tech
{dbhaduri, shukla}@vt.edu

Deji Coker Valerie Taylor
Texas A&M University
{coker, taylor}@cs.tamu.edu

Paul Graham Maya Gokhale
Los Alamos Nat'nl Labs
{grahamp, maya}@lanl.gov

Abstract

It is anticipated that self assembled ultra-dense nanomemories will be more susceptible to manufacturing defects and transient faults than conventional CMOS-based memories, thus the need exists for fault-tolerant memory architectures. The development of such architectures will require intense analysis in terms of achievable performance measures— power dissipation, area, delay and reliability. In this paper, we propose and develop a hybrid automation framework, called HMAN, that aids the design and analysis of fault-tolerant architectures for nanomemories. Our framework can analyze memory architectures at two different levels of the design abstraction, namely the system and circuit levels. To the best of our knowledge, this is the first such attempt at analyzing memory systems at different levels of abstraction and then correlating the different performance measures. We also illustrate the application of our framework to self-assembled crossbar architectures by analyzing a hierarchical fault-tolerant crossbar-based memory architecture that we have developed.

1. Our Automation Methodology

Hybrid Memory Analyzer (HMAN) is composed of fault and circuit models that are specific to nanoscale memories. The toolset is designed such that it scales well for the analysis of ultra-dense nanosystems. HMAN is capable of evaluating the reliability and area overheads of fault-tolerant memories at the system level and can also measure the delay penalties at the circuit level. Figure 1 shows our design flow which is outlined below:

1. The system designer designs a fault-tolerant memory. For example, the fault-tolerance technique may be sparing, banking, or our multi-junction hierarchical crossbar-based architecture [2].
2. Configuration parameters are given as inputs to HMAN. These include the specific fault-tolerant technique being used in the memory design, the memory size, the desired reliability thresholds, redundancy levels at the different gran-

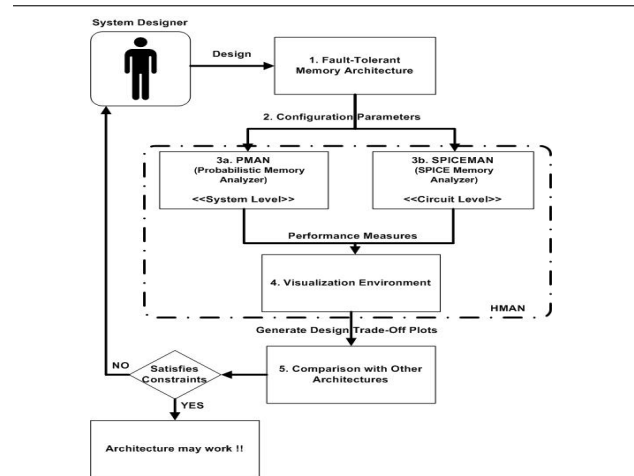


Figure 1. HMAN Framework

ularity levels [3], R_H , R_{LO} , interconnect network capacitance, and failure distributions of the junctions and peripheral interconnects.

3. HMAN consists of two basic analyzers. These two analyzers work at different levels of design abstraction and evaluate different performance measures. These measures are correlated to determine meaningful design Pareto points.

3a. PMAN (Probabilistic Memory Analyzer) is a probabilistic model-checking based tool that works on top of the PRISM [4] engine. Parameterized MDP models are used to represent the probabilistic behavior of fault-tolerant memory architectures, so that different memory configurations can be analyzed with ease. PMAN's fault models can be defined with simple scripts. The fault model used in this work does not support pattern specific fault occurrences. All junctions have an equal likelihood of failing and this probability value is specified by the designer. Also, failure distributions of the peripheral interconnects model the transient faults that may affect the system.

3b. SPICEMAN (SPICE based Memory Analyzer) is a tool based on HSPICE that analyzes memory architectures at the

circuit level. The HSPICE model is defined using PERL. MATLAB is used to compute the appropriate load resistance R_L necessary to achieve the desired ON/OFF ratio (logic thresholds) for a specific architectural configuration and then provided to PERL. The PERL script is parameterized and takes as input the memory size and the amount of redundancy being inserted. In this work, SPICEMAN uses the circuit model in [1] to compute the worst case read-out time for a single bit. This gives a lower bound on the performance of different crossbar-based nanomemories.

4. Since it is easier to observe and compare different trade-off points visually, we have developed a MATLAB script that plots these interacting trade-off points. This step in our design flow is important, since trade-offs such as reliability-redundancy from PMAN and redundancy-delay from SPICEMAN can be visually correlated.

2. Experimental Results

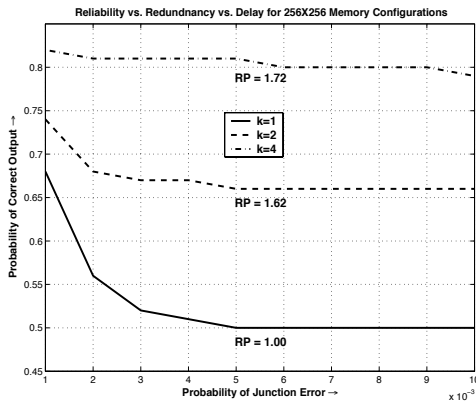


Figure 2. Reliability vs. Redundancy vs. Delay trade-offs for 256×256 Memory

Reliability vs. Redundancy vs. Delay: We use HMAN to compute reliability-redundancy-delay trade-offs for different memory configurations of our fault-tolerant scheme [2]. Figure 2 shows the plots for a 256×256 memory configuration. We plot the trade-off points for small junction failure probabilities ranging from $[0.001, 0.01]$. It can be observed from Figure 2 that as the redundancy (value of k) increases, the reliability indicated by the probability of signal output correctness increases. This observation is independent of the junction failure probability value. Also, the value of RP increases when the redundancy in each cell k increases, implying an intuitive increase in the signal delay leading to performance degradation of the memory.

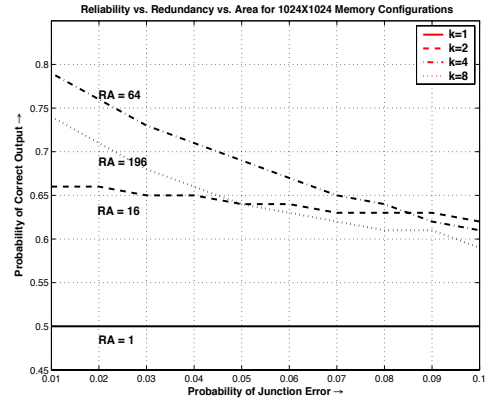


Figure 3. Reliability vs. Redundancy vs. Area trade-offs for 1024×1024 Memory

Reliability vs. Redundancy vs. Area: Figure 3 shows the reliability-redundancy-area trade-offs for the 1024×1024 memory configuration. These design points pertain to the our multi-junction architecture and are computed for large junction failure probabilities ranging from $[0.01, 0.1]$. A few of the major and counter-intuitive observations from Figure 3 are: *for a redundancy factor of $k = 8$, the reliability of the system degrades for junction failure probabilities between $[0.01, 0.45]$ relative to the system with $k = 4$. It is also observed that beyond a junction failure probability of 0.45, a redundancy factor of 2 is better than 8.* This counter-intuitive observation is due to the redundancy factors reaching the point of diminishing returns in terms of system reliability gains. It is also worth mentioning that when this memory configuration with a redundancy factor of 8 was analyzed by SPICEMAN at the circuit level, the signal delay computed was very high. This is due to the high cumulative junction resistance induced by the redundancy factor of each cell. *Such concurrence of the conclusions drawn from analysis of the same system at different levels of the design abstraction is a major highlight of this work.*

References

- [1] C. Amsinck, N. Spigna, S. Sonkusale, D. Nackashi, and P. Franzon. Scaling challenges for molecular electronic array structures. In *Workshop on Non-Silicon Computation (NSC)*, 2003.
- [2] D. Bhaduri, D. Coker, S. Shukla, V. Taylor, P. Graham, and M. Gokhale. A hybrid framework for design and analysis of fault-tolerant architectures and its applications to nanoscale molecular crossbar memories. Technical report, Fermat Lab, Virginia Tech, 2005.
- [3] D. Bhaduri and S. Shukla. Nanoprism: A tool for evaluating granularity vs. reliability trade-offs in nano-architectures. In *GLSVLSI*, Boston, MA, April 2004. ACM.
- [4] Web Page: www.cs.bham.ac.uk/~dxdp/prism/.