A Framework for System Reliability Analysis Considering Both System Error Tolerance and Component Test Quality

Sung-Jui Pan and Kwang-Ting Cheng
Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106
{srpan, timcheng}@ece.ucsb.edu

Abstract

The failure rate, the sources of failures and the test costs for nanometer devices are all increasing. Therefore, to create a reliable system-on-a-chip device requires designers to implement fault tolerance. However, while system-level fault tolerance could significantly relax the quality requirements of the system's building blocks, every fault-tolerant scheme only works under certain failure mechanisms and within a certain range of error probabilities. Also, designing a system with a high failure-rate component could be very expensive because the growth rate of the design complexity and the system overhead for fault tolerance could be significantly greater than the component failure rate. Therefore, it is desirable to understand the trade-offs between component test quality and system fault-tolerant capability for achieving the desired reliability under cost constraints. In this paper, we propose an analysis framework for system reliability considering (a) the test quality achieved by manufacturing testing, on-line self-checking, and off-line built-in self-test; (b) the fault-tolerant and spare schemes; and (c) the component defect and error probabilities. We demonstrate that, through proper redundancy configurations and low-cost testing to insure a certain degree of component test quality, a low-redundant system might achieve equal or higher reliability than a high-redundant system.

1 Introduction

Achieving the highest test quality under test cost constraints has always been the goal of manufacturing testing. However, because of the low yield of nanometer devices, there is an increasing demand to relax testing requirements in order to improve the yield. For example, a defective memory chip may be acceptable in a downgraded system if a defective row or a row containing one or more defective memory cells can be disconnected. On the other hand, because of the increasing test cost and the higher defect rate for nanometer devices, manufacturing testing encounters a higher rate of test escape. In addition, there are increased sources of failures not caused by manufacturing defects such as marginality, transient and soft errors, which are hard or even impossible to detect in the manufacturing line. Therefore, systems based on less reliable components must be designed with some fault-tolerant capabilities. The fundamental scheme of all fault-tolerant techniques is redundancy – the more redundancy in a system, the higher the reliability. However, a high-redundant system implies that significant area overhead, complicated control, and high design complexity will be required for such a system. Therefore, determining an optimal scheme that strikes the best balance between fault tolerance and component test quality for a given reliability requirement becomes an essential task.

For some applications such as multimedia computing and communications, insignificant data corruption at the system level might be acceptable. Jiang and Gupta in [8] proposed a new testing strategy for chips used in such applications: chips are rejected only if (a) they incur failures at control outputs or (b) the value at data outputs differ from the expected value more than a given threshold. Furthermore, Breuer in [4, 5] suggested that a functionally imperfect chip can still be accepted if the error caused by the corrupted data of the chip is smaller than a given threshold. Also, because some faults can only cause insignificant data corruption, Lee *et. al.* in [2, 3] proposed a heuristic to remove these faults from the list of the fault candidates for ATPG, thereby reducing the test cost and also improving the yield. However, the threshold for acceptable data corruption could be difficult to determine for control-intensive systems. Also, different applications using the same chip could have different thresholds for error tolerance, which makes the threshold testing infeasible.

A variety of fault-tolerant techniques have been used to facilitate a system to detect and to mask or correct failures, including error correction codes (ECC) [9], spatial and time redundancy [7], on-line checkers [10], and spares [1]. These techniques use different redundancy configurations for designing a reliable system under certain assumptions of error occurrence. For example, the underlying assumption of a working triple-modular-redundancy system is that in any clock cycle one component in the system at most has a faulty behavior. In order to tolerate some failures in a microprocessor, a small and robust checker is proposed in [10] to detect and to correct failures by re-executing the non-speculative instructions in the checker. Once an error is detected, the checker will correct the error, reset the processor pipeline, and restart the processor at the next instruction. However, a higher failure rate could cause a higher number of restarts of the processor, thus possibly dramatically decreasing the overall processor performance.

Because nanometer devices have a low-yield rate and a

978-3-9810801-2-4/DATE07 © 2007 EDAA

high test cost, designing a system with low or no redundancy using only highly reliable components could cost more than that of a system with a slightly higher-redundancy using components tested within the test budget might. Therefore, considering the fault tolerance schemes and parameters for component test quality are essential for designing a cost-effective and reliable system.

In this paper, our contribution is an analysis framework for system reliability considering (1) the test quality achieved by manufacturing testing, on-line self-checking, and off-line built-in self-test (BIST); (2) the fault-tolerant and spare schemes; and (3) the component defect and error probabilities. Such analysis could help explore the trade-offs between component test quality and system fault-tolerant capability for achieving the desired reliability. We will present the framework and experimental results to illustrate (a) the design of a reliable system must consider both fault tolerance and the test quality of components; (b) on-line self-checking/BIST can help achieve a lower system error rate with fewer redundant components; (c) a finer granularity of spares can help lower the system error rate without increasing the amount of total redundant hardware; and (d) the fault-tolerant schemes could be applied only to a partially selected set of outputs to reduce the overhead without compromising the reliability.

The rest of the paper is organized as follows. Section 2 describes the system error-rate analysis considering the fault-tolerant and spare schemes. Section 3 describes the component-yield analysis considering the test quality of a component. In Section 4, we discuss how to estimate the system error rate for individual system output based on a set of relevant parameters. Section 5 shows the experimental results, and the concluding remarks are given in Section 6.

2 System Error Rate

The system error rate is defined as the probability of errors being produced at system outputs. In this paper, we denote the system reliability as the complement of the system error rate. While we use the N-modular redundancy [9] as the example of the fault-tolerant system to illustrate the analysis framework, this analysis can be used for other fault-tolerant architectures. In Section 2.1, we will briefly review the N-modular redundancy that uses a voter or comparators to detect and to mask errors. We then will discuss the spare scheme in Section 2.2 as well as the method for computing the corresponding system error rate.

2.1 N-Modular Redundancy

The most well-studied N-modular redundancy method is triple modular redundancy (TMR). This consists of three identical components and a voter to detect and to mask failures. We may assume that every error in an erroneous component can be propagated to component outputs. Because the output of the TMR will be consistent with at least two corresponding component outputs, the TMR can function correctly if, in any cycle, there is one faulty component at most in the system. We can generalize the TMR as being a system with $2n+1$ identical components that can function correctly if at least $n+1$ components generate the same output data in any cycle. Because a system with an even number of components may produce an inconclusive result, the number of components used must be odd. However, if the system is designed with the lockout scheme that can disable a faulty component, then the system can function with an even number of components. For example, a system with four-modular redundancy (4MR) will become a TMR system if a faulty component is detected and disabled by the voter.

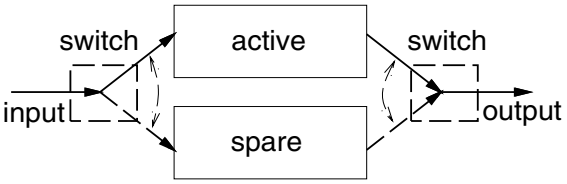

Figure 1: System with one active and one spare components.

Instead of using the voting scheme, we can pair the outputs of every two components for comparison. Thus, an error in a component can be detected if the pair of outputs does not match. For example, we can obtain two pairs of components, P_1 and P_2, for a four-modular-redundancy system. The system output will be the output of one of the component pairs. If the outputs of the active component pair P_1 do not match, the system will disable P_1 and switch to the other pair P_2. In other words, the system (pairwise:4) can function correctly if (a) no error occurs in any of the four components; (b) if only one component is faulty; (c) if two components are faulty, but are within the same pair (i.e., four out of six cases of two faulty components can cause a system failure).

2.2 Spare Scheme, BIST, On-Line Self-Checking

As shown in Figure 1, unlike the voting scheme, which constantly compares outputs among identical components, the system with spares only has one active component during the normal operation. The spare scheme would either rely on on-line self-checking or off-line built-in self-test (BIST) to identify faulty components in the field. Once a faulty active component is identified, the system is automatically reconfigured to replace it with a fault-free spare. The system can continue to operate if at least one component is fault-free. In addition, the self-checking/BIST capability can be used to trigger diagnosis. That is, if the resolution of self-checking/BIST is coarser than that of spares, the source component likely to fail can be identified before triggering the reconfiguration. In this paper, we use logicBIST as the component test method for the experiment.

As illustrated conceptually in Figure 2, the spare scheme has another parameter: granularity of spares. For one extreme case, the spare of the entire system is in one unit, so the system will switch the entire system to the spare system if an error is detected. If the granularity of spares is finer, only a small faulty component will be replaced. It is intuitive that the finer granularity of spares, the lower the overall system error

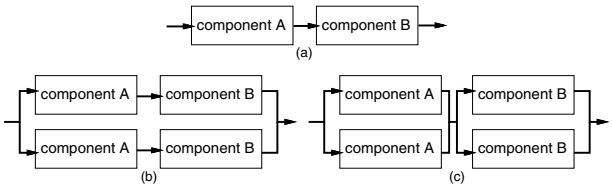

Figure 2: Redundancy techniques: (a) single system (no spare) (b) system redundancy (c) component redundancy.

rate will be. However, the finer the granularity, the more complex the control for reconfiguration must be. In addition, the amount of hardware required to support reconfiguration will be higher. For example, the finest level of granularity for reconfiguration is that each gate can independently be replaced by the corresponding spare gate. For such a scheme, we need to add one more multiplexor for each gate in order to select the proper gate output. As a result, the total number of gates will be at least three times greater than is typically found in a single system.

2.3 System Error Rate of Spare Scheme

We assume that the system consists of two components, one active and one spare; and that both components are able to conduct a self-test. For the simplicity of the analysis, we assume that the added control logic is error free. This system will stop operating when both components fail the self-test. Therefore, the probability that the system is functional (i.e. with at least one passing component) would be:

$$1 - ((1 - R) \times P_D)^2, \tag{1}$$

where R and P_D denote the component yield and the probability of the test to detect an error in the component respectively. The component yield is defined as the probability of no errors/defects in a component. The details of the calculation of component yield will be discussed in Section 3. In this paper, we use the bridging coverage estimation (BCE), proposed in [6], to approximate P_D in order to cover both modeled and unmodeled faults. The fault coverage of stuck-at and transition faults, for example, tend to be modeled faults. Due to test escape, having two passing components does not imply both components are fault-free. It could be possible that one or both components are faulty. Thus, the probability of having a system with one or two passing components can be computed by Equations 2 and 3 respectively.

$$P(\text{1 passing}) = 2R(1-R)P_D + 2(1-R)^2 P_D(1-P_D) \tag{2}$$

$$P(\text{2 passing}) = R^2 + 2R(1-R)(1-P_D) + (1-R)^2(1-P_D)^2 \tag{3}$$

If every error in an erroneous component can be propagated to component outputs, the system error rate is equal to the passing-component yield for the system with one passing component. On the other hand, the system with two passing components fails when both components fail. Thus, the system reliability (i.e. the complement of system error rate) can be computed as follows:

$$\frac{\text{Equation 3}}{\text{Equation 1}}(1 - (1 - R(T))^2) + \frac{\text{Equation 2}}{\text{Equation 1}}R(T), \tag{4}$$

where the $R(T)$ denotes the component yield if the component passes the test set T. This analysis can be easily extended to a system with more than one redundant component.

3 Component Yield

The component yield is defined as the probability of no errors/defects in a component. Thus, the yield of a component with N nets can be modeled as:

$$R = \prod_{i=1}^{N}(1 - ED_i),$$

where ED_i denotes the error density of net i. The error density of a net is the probability of the net being erroneous. If the component-redundancy scheme is implemented for a component with M spares, the component yield could be expressed as:

$$R = \prod_{i=1}^{N}(1 - \prod_{j=1}^{M+1} ED_{ij}),$$

where ED_{ij} denotes the error density of the spare net j for net i, $i = 1 \ldots N$.

Each test pattern detects some faults and has a finite probability of screening out faulty chips. Therefore, the error density for each net should be adjusted accordingly after each test pattern. If net i passes a test set which detect the fault on net i for t_i times (in the following, we simply say "net i passes t_i patterns"), the error density of net i can be updated by the following conditional probability:

$$ED_i(t_i) = \frac{P_i(t_i \cap E)ED_i}{P_i(t_i \cap E)ED_i + P_i(t_i \cap \overline{E})(1 - ED_i)}, \tag{5}$$

where $ED_i(t_i)$, $P_i(t_i \cap E)$, and $P_i(t_i \cap \overline{E})$ denote the error density of net i if the net passes t_i patterns, the probability of an erroneous net i passing t_i patterns, and the probability of an error-free net i passing t_i patterns, respectively. Obviously, $P_i(t_i \cap \overline{E})$ equals to 1. Equation 5 implies that if t_i patterns could detect every possible error associated with net i (i.e. $P_i(t_i \cap E)$ equals to zero), net i will be error free if it passes these patterns. However, not every defect/error can be detected even if the coverage of the test set for the modeled faults (such as stuck-at and transition faults) is 100% and even if every modeled fault is detected multiple times. Therefore, in order to approximate the actual fault/defect coverage, we use the bridging coverage estimation (BCE), proposed in [6] to estimate the coverage for unmodeled faults, and, in turn, use it for estimating $P_i(t_i \cap E)$. Based on the BCE, the probability $P_i(t_i \cap E)$ can be modeled as:

$$P_i(t_i \cap E) = \frac{1}{2^{t_i}}.$$

The probability of an erroneous net i passing t_i patterns decreases exponentially with respect to the count of the net's modeled fault being detected. Note that the error density can be classified into two sub-categories: the manufacturing error density (caused by manufacturing defects) and the post-manufacturing error density (caused by transient, soft, and other noise-induced errors). This analysis framework could be used to estimate the sensitivity of system reliability to such error densities.

4 System Error Rate for Individual Output

We have discussed how to estimate the system error rate and the component yield in Sections 2 and 3 respectively. Note that the definition of system error rate is the probability that no errors are produced at system outputs. However, not every output has the same size of a fan-in circuit. In addition, because of the circuit structure, the detection probability for net i may be different at different outputs. The detection probability of net i at output j is the probability of a pattern that can activate an error at net i and cause an erroneous response at output j. In this section, we will discuss how to estimate the system error rate for the output of a single system as shown in Figure 2(a). The system error rate for each output of different fault-tolerat schemes can be computed using the analysis described in Section 2.

The error rate for a system's output j resulting from an error at net i can be modeled as the product of the error density of net i and the detection probability of net i at output j. To estimate the detection probability of net i, we could construct a miter circuit, as shown in Figure 3. This consists of a good circuit and a faulty circuit with an injected fault at net i. The detection probability of net i is equal to the signal probability of the output j. Computing the signal probability is a well-studied problem for various applications. It has been known that computing the exact signal probability for a large circuit is not feasible due to its high complexity. We therefore use the cutting algorithm proposed in [14] to compute the lower and upper bounds of the signal probability. In this paper, we use the upper bound of the signal probability as the detection probability of net i at output j. If a component passes a test set, the system error rate for output j with N nets in its fan-in circuit can be modeled as:

$$ER_j = 1 - \prod_{i=1}^{N}(1 - ED_i(t_i) \times DP_i),$$

where $ED_i(t_i)$ and DP_i denote the error density of net i that passes t_i patterns and the detection probability of net i at output j, respectively.

5 Experimental Results

To illustrate the analysis framework, we use five different fault-tolerant schemes – pairwise:4, TMR, 4MR with the lockout scheme (4MRL), 5MR, and 5MR with the lockout scheme (5MRL) – to implement five different modules in the Sun picoJavaII microprocessor [11]. We also incorporated logicBIST

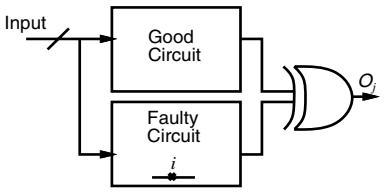

Figure 3: A miter circuit consists of a good circuit and a faulty circuit with an injected fault at net i.

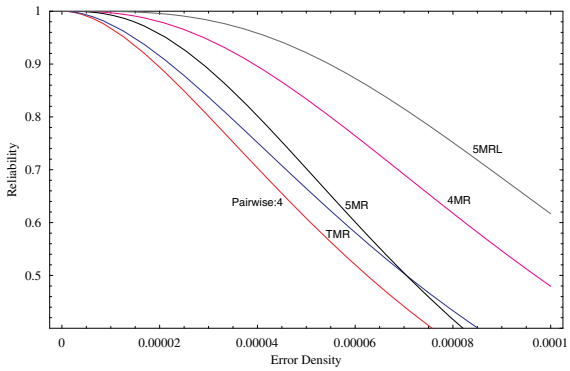

Figure 4: Reliability of different fault-tolerant systems.

for each of the modules that can be triggered to test the module either in the manufacturing line or in the field. In the logicBIST implementation, random test patterns are generated from a 32-bit linear feedback shift register (LFSR). Due to space limitations, most of the figures in this section only show the data for one module in the picoJavaII – the stack unit management (SMU). Similar results have been obtained for other modules as well.

5.1 Fault Tolerance vs. Error Density

The reliability for different fault-tolerant schemes using components without any testing is shown in Figure 4. In contrast to the 5MR, which requires at least three fault-free components for any cycle, the 4MRL requires only two fault-free components because of the lockout scheme. Therefore, the reliability of the 4MRL is greater than that of the 5MR. Although the pairwise:4 scheme has one more redundant component than the TMR, the reliability of the TMR system is greater because in only two out of six cases having two faulty components does not cause the pairwise:4 to fail.

Figure 5 includes the results of using components that have been tested using relatively low-quality tests, such as logicBIST using a low pattern count without any test point insertion, for the TMR system. The two curves TMR_1 and TMR_2 show the results of the TMR systems that use components tested with 3×10^6 and 10^6 random patterns respectively. The results are intuitive: given the same error density, the more test patterns, the greater the component test quality, and thus the greater the system reliability. Note that although the 4MRL has one more redundant component than the TMR, the reliability of both TMR systems is greater than that of the 4MRL when the error density becomes non-trivial. Also, the reliabil-

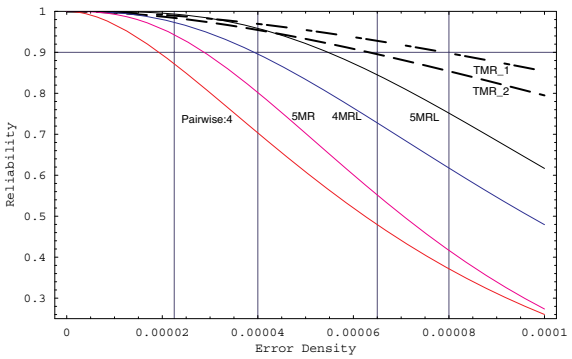

Figure 5: Reliability of fault-tolerant schemes and the TMRs using components tested with 3×10^6 (TMR_1) and 10^3 (TMR_2) random patterns by logicBIST.

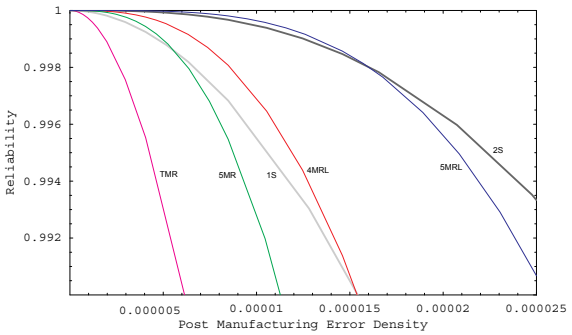

Figure 6: Reliability of systems using spare schemes and fault-tolerant systems.

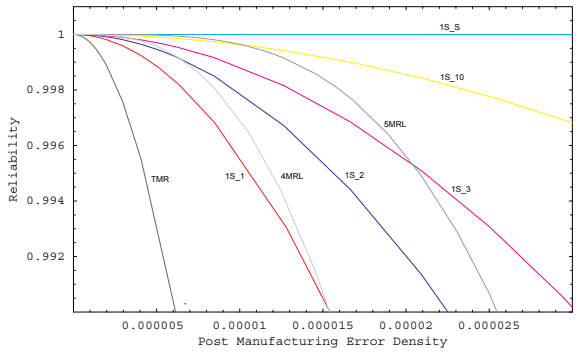

Figure 7: Reliability for systems with different granularity of the spares and fault-tolerant systems.

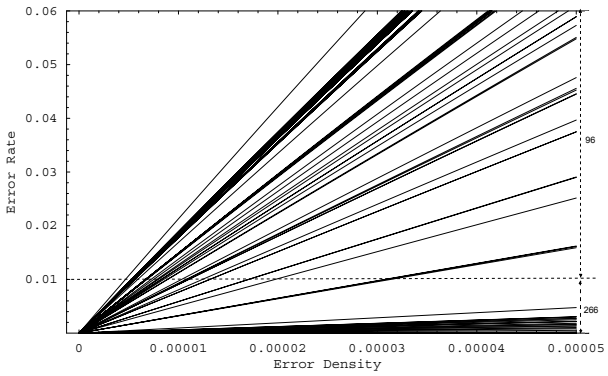

Figure 8: Two groups of system error-rates for outputs.

ity of both TMR_1 and TMR_2 systems decreases much more slowly than that of the 5MRL system as the error density increases despite of having more redundant components and the lockout scheme. These results, while preliminary and while having a limited scope, do strongly indicate the need for considering both fault tolerance and component test quality jointly to design a cost-effective and reliable system.

5.2 Fault Tolerance vs. Spares

Figure 6 shows the reliability of the fault-tolerant systems and systems using the spare scheme. The curves labled as 1S and 2S show the results of systems with one and two spare components respectively. All components of both 1S and 2S systems are tested by logicBIST with 10^6 patterns. As shown in Figure 6, in contrast to the TMR system that requires three components, the 1S system (using two components in total) can achieve a greater reliability. These results clearly show that a high-redundant system is not necessarily more reliable than a system using the spare scheme with a limited number of spares.

Instead of using the entire system as one spare as shown in Figure 2(b), we partitioned the original system into k equally-sized components. We assume that each partitioned component has one spare and both can be tested by logicBIST's random patterns. In Figure 7, the curve labeled as 1S_k denotes the result of the system partitioned into k components The curve labeled 1S_S denotes the system implemented with the finest granularity, i.e. each individual gate can be replaced independently by its spare. The results confirm that the finer granularity of spares, the greater system reliability becomes, although utilizing this scheme would require greater hardware and performance overhead due to the reconfiguration logic. Therefore, in addition to adding redundancy, we should also analyze the reliability impact of the spare's granularity in designing a system.

5.3 System Error-Rate vs. Error Density

Figure 8 shows the system error rate for each output of a single system – each line represents the error rate of an output. Note that different outputs have different error rates. For some applications in multimedia and communications, the significance of errors at different outputs could be different, too. For example, a low error rate at the least significant bits of a data might be acceptable for image analysis applications. Therefore, combining these two factors at each output (i.e. error rate and error significance), we could reduce the overhead by implementing fault tolerance only at a partially selected set of outputs that have a non-trivial error rate with a non-trivial significance. As shown in Figure 8, when the error density is 5×10^{-5}, only 96 out of a total 362 outputs have an error rate greater than 0.01. Therefore, instead of duplicating the entire component, we could duplicate only a subset of circuits

Ckt.	Stuck-at Coverage	BCE Coverage
SMU	71.6%	71.3%
EX	87.8%	87.7%
ICU	98.6%	96.9%
IFU	59.2%	53.7%
PIPE	96.8%	95.5%

Table 1: Fault coverage of five different modules.

ED	10^{-6}		4×10^{-6}		8×10^{-6}		10^{-5}		2×10^{-5}	
Ckt.	ER	ERO	ER	ERO	ER	ERO	ER	ERO	ER	ERO
SMU	3.0×10^{-5}	1.4×10^{-5}	5.2×10^{-4}	2.2×10^{-4}	2.1×10^{-3}	8.6×10^{-4}	3.2×10^{-3}	1.3×10^{-3}	1.2×10^{-2}	5.2×10^{-3}
EX	1.3×10^{-3}	1.5×10^{-5}	1.9×10^{-2}	2.4×10^{-4}	6.5×10^{-2}	9.6×10^{-4}	9.5×10^{-2}	1.5×10^{-3}	2.7×10^{-1}	5.7×10^{-3}
ICU	1.3×10^{-4}	1.1×10^{-7}	2.0×10^{-3}	1.7×10^{-6}	7.8×10^{-3}	6.9×10^{-6}	1.2×10^{-2}	1.1×10^{-5}	4.3×10^{-2}	4.3×10^{-5}
IFU	4.4×10^{-4}	4.2×10^{-5}	6.7×10^{-3}	6.6×10^{-4}	2.5×10^{-2}	2.6×10^{-3}	3.7×10^{-2}	4.0×10^{-3}	1.2×10^{-1}	1.5×10^{-2}
PIPE	4.0×10^{-6}	9.6×10^{-9}	6.0×10^{-5}	1.5×10^{-7}	2.5×10^{-4}	6.1×10^{-7}	3.9×10^{-4}	9.6×10^{-7}	1.5×10^{-3}	3.8×10^{-6}

ED: Error Density, ER: System Error Rate, and ERO: System Error Rate for a System Output

Table 2: System error-rates for different modules implemented with the TMR.

that support these 96 outputs, if an error rate of this level is acceptable for the application.

We implemented the TMR scheme for five modules of the picoJavaII and each component was tested with 10^6 random patterns. Table 1 shows the fault coverage (stuck-at and BCE) for these five modules. Table 2 shows the TMR system error rate (ER) and the maximum TMR system error rate for an system output(ERO) for different error densities. As shown in Table 2, at the error density of 10^{-6}, the system error rate for these five modules could be 2X, 86X, 1181X, 10X, and 417X greater than the system error rate for an system output, respectively. These results indicate that each output can only observe a small fraction of errors from the corresponding fan-in circuit because of the circuit structure.

6 Conclusion

In this paper, we propose an analysis framework for the system reliability to help explore the trade-offs between component test quality and system fault-tolerant capability. Such an analysis helps derive essential information for designing and optimizing a system that achieves a desired level of system reliability under cost constraints. The input parameters for the analysis include (1) the error probability of a signal in the system; (2) the fault-tolerant and spare schemes; (3) the testing scheme and the corresponding test quality it achieves; and (4) the granularity of spares. In addition, we also estimate the system error rate for individual system output that is also an important quality factor for use in some applications. The preliminary results, using the picoJavaII as a test case, demonstrate that a low-redundant system, with spares at the "right-level" of granularity and also equipped with low-cost BIST or on-line checking support to ensure a low-enough component error probability, could achieve a much higher reliability than a high-redundant system that primarily relies on the fault-tolerant scheme to mask errors.

References

[1] K. Constantinides *et. al.*, "BulletProof: A Defect Tolerant CMP Switch Architecture," in *Proc. of High-Performance Computer Architecture Symposium*, pp. 5–16, Feb. 2006.

[2] T.-Y. Hsieh *et. al.*, "An Error-Oriented Test Methodology to Improve Yield with Error-Tolerance," in *Proc. of VLSI Test Symposium*, pp. 130–135, April 2006.

[3] K.-J Lee *et. al.*, "A Novel Test Methodology Based on Error-Rate to Support Error-Tolerance," in *Proc. of International Test Conference*, pp. 1136-1144, Nov. 2005.

[4] M. A. Breuer *et. al.*, "Defect and Error Tolerance in the Presence of Massive Numbers of Defects," in *IEEE Design and Test*, Vol. 21, No. 3, pp. 216–227, May-June 2004.

[5] M. A. Breuer, "Determining Error Rate in Error Tolerant VLSI Chips," in *Proc. of International Workshop on Electronic Design, Test and Applications*, pp. 321–326, Jan. 2004.

[6] B. Benware *et. al.*, "Impact of Multiple-Detect Test Patterns on Product Quality," in *Proc. of International Test Conference*, pp. 1031–1040, Sep. 2003.

[7] W. J. Townsend *et. al.*, "Quadruple Time Redundancy Adders," in *Proc. of International Symposium on Defect and Fault Tolerance in VLSI*, pp. 250–256, Nov, 2003.

[8] Z. Jiang and S.K. Gupta, "An ATPG for Threshold Testing: Obtaining Acceptable Yield in Future Processes," in *Proc. of International Test Conference*, pp. 824–833, Oct. 2002.

[9] M. L. Shooman, "Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design," *A Wiley-Interscience Publication*, 2002.

[10] T. Austin, "DIVA: A Dynamic Approach to Microprocessor Verification," in *J. of Instruction-Level Parallelism*, Vol. 2, May 2000.

[11] "picoJava-II Microarchitecture Guide," Sun Microsystems Inc., March 1999.

[12] H. Al-Asaad *et. al.*, "Online BIST for Embedded Systems," in *IEEE Design & Test*, Vol. 15, No. 4, pp. 17–24, Oct.–Dec. 1998.

[13] Paul H. Bardell *et. al.*, "Built-In Test for VLSI: Pseudorandom Techniques," *A Wiley-Interscience Publication*,1987.

[14] J. Savir *et. al.*, "Random Pattern Testability," in *IEEE Trans. on Computers*, Vol. C-33, pp. 1041-1045, Jan. 1984.