01 Dec 2004

Evaluating the Repair of System-on-Chip (SoC) using Connectivity

Minsu Choi
Missouri University of Science and Technology, choim@mst.edu

Nohpill Park

Vincenzo Piuri

Fabrizio Lombardi

Recommended Citation

M. Choi et al., "Evaluating the Repair of System-on-Chip (SoC) using Connectivity," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, no. 6, pp. 1464-1472, Institute of Electrical and Electronics Engineers (IEEE), Dec 2004.
The definitive version is available at https://doi.org/10.1109/TIM.2004.834603

Evaluating the Repair of System-on-Chip (SoC) Using Connectivity

Minsu Choi, *Member, IEEE*, Nohpill Park, *Member, IEEE*, Vincenzo Piuri, *Fellow, IEEE*, and Fabrizio Lombardi, *Member, IEEE*

***Abstract*—This paper presents a new model for analyzing the repairability of reconfigurable system-on-chip (RSoC) instrumentation with the repair process. It exploits the connectivity of the interconnected cores in which unreliability factors due to both neighboring cores and the interconnect structure are taken into account. Based on the connectivity, two RSoC repair scheduling strategies, *Minimum Number of Interconnections First (I-MIN)* and *Minimum Number of Neighboring Cores First (C-MIN)*, are proposed. Two other scheduling strategies, *Maximum Number of Interconnections First (I-MAX)* and *Maximum Number of Neighboring cores First (C-MAX)*, are also introduced and analyzed to further explore the impact of connectivity-based repair scheduling on the overall repairability of RSoCs. Extensive parametric simulations demonstrate the efficiency of the proposed RSoC repair scheduling strategies; thereby manufacturing ultimately reliable RSoC instrumentation can be achieved.**

***Index Terms*—Configurability, connectivity, reconfigurable system-on-chip (RSoC), reliability, repair, repairability.**

I. INTRODUCTION

THE INCREASING demand on operation speed, integration density, and customizability for tomorrow's high-performance instrumentation has motivated high performance system development. System-on-chip (SoC) technology provides potential advantages of high integration density, small interconnection delay and high system performance [8], [12]–[14], [16]–[18], [21], and [23]. Thus, SoC is one of the key technology choices for high-performance instrumentation development [9]. For the purpose of customizability and repairability, embedding reconfigurable components along with ordinary cores with fixed functionality are commonly practiced [1]–[3], [7], [10], [11], [15], [19], [20], and [22]. The SoC with reconfigurable resources is commonly referred to as *reconfigurable system-on-chip (RSoC)*. In this paper, connectivity-driven repair algorithms for an RSoC which exploits the reconfigurable redundancy will be proposed. *Test* and *repair* are essential processes for achieving high-yielding SoCs. After the fabrication phase, each SoC undergoes a test phase where defective cores are diagnosed and identified. Usually, defective cores on RSoCs are deemed to be reworked, which means that defective cores can be repaired by reconfigurable redundancy. The overall quality of the repair process significantly affects the final quality of the repaired RSoC. However, the repair process is not free from penalty, since faulty core isolation and reconfiguration processes may affect the overall system integrity, the reconfigured interconnect structure routability, and the neighboring cores' functionality due to the serious interconnection network reconfiguration and associated programmable logic gate programming. For the extra long interconnects, even signal boosters are required to guarantee the integrity of the routed signals [5]. For more structured and reliable operations, protocol-based interconnction networks are commonly implemented for SoCs as well [4].

How densely a *repair-candidate core* (i.e., the core to be repaired since it is diagnosed as defective) is connected to the neighboring parts of the RSoC is referred to as *connectivity* [6]. If a repair-candidate core's connectivity is high, the repair process applied to the core may impair the reliability of the RSoC while it repairs the core because of the physically associated components with the core. Thus, selection of a repair-candidate core that results in the least negative effects on the overall reliability at each repair cycle seems to be crucial in the RSoC repair process.

The objective of this paper is to extensively investigate the effect of the connectivity of repair-candidate cores on the overall yield (i.e., ratio of the total number of functional RSoCs out of the total number of fabricated RSoCs) of the repaired RSoC and to propose various repair scheduling strategies. Also, how improper repair scheduling could degrade the overall quality of repaired RSoCs will be extensively studied. Thus, results and findings from this research will be beneficial to RSoC-based digital instrumentation developers.

The organization of the paper is as follows. In Section II, review and preliminaries related to this research work will be given. In Section III, analytical characteristics of the RSoC repair process for the proposed repair scheduling strategies will be discussed. Section IV will describe details of the proposed RSoC repair scheduling strategies. In Section V, extensive parametric analysis and simulations will be provided to demonstrate and verify the accuracy and efficiency of the proposed approaches.

II. REVIEW AND PRELIMINARIES

In this work, an RSoC is modeled as a set of cores, their interconnect structure, reconfigurable interconnects, and reconfigurable logic redundancy is shown in Fig. 1, in which the

---

Manuscript received June 15, 2003; revised June 30, 2004.

M. Choi is with the Department of Electrical and Computer Engineering, University of Missouri, Rolla, MO 65409-0040 USA (e-mail: choim@umr.edu).

N. Park is with the Department of Computer Science, Oklahoma State University, Stillwater, OK 74078-1053 USA (e-mail: npark@a.cs.okstate.edu).

V. Piuri is with the Department of Information Technology, University of Milan, 26013 Crema, Italy. (e-mail: piuri@dti.unimi.it).

F. Lombardi is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: lombardi@ece.neu.edu).

Digital Object Identifier 10.1109/TIM.2004.834603

RSoC has six cores and corresponding interconnect structure. The repair process induces faulty core isolation, programmable logic reconfiguration, and interconnection rerouting. Although the process repairs the RSoC, the unreliability induced by the reconfiguration process (i.e., imperfect faulty core isolation, programmable logic reconfiguration and interconnection rerouting) may have negative effects on the overall quality of the repaired RSoC.

Once fabricated, embedded cores cannot be physically replaced. Thus, embedded redundancy must be practiced for better yielding SoCs. Since a number of embedded hybrid cores are usually involved to design an SoC, a legacy modular redundancy scheme (i.e., embedding of extra cores to repair faulty cores) may require significant die area investment and its redundancy utilization also may be very low (i.e., unused spare cores are likely). The proposed reconfigurable redundancy architecture for SoC repair consists of two key components: *reconfigurable logic redundancy* and *reconfigurable interconnect redundancy*. The embedded cores are tested in order to identify faulty cores, if any. Then, the faulty cores and their interconnects are emulated by the reconfigurable logic and interconnect redundancy to restore the original functionality of the RSoC. The following case study clarifies the proposed RSoC core repair scheme based on the reconfigurable redundancy. Suppose that an RSoC shown in Fig. 1 is tested and diagnosed, and its core 4 is identified as faulty. Then, an emulated core $4'$ is implemented by using the reconfigurable logic redundancy and core $4'$'s interconnects are rerouted to the core $4'$ via the reconfigurable interconnect redundancy. As a result, the repaired RSoC is shown in Fig. 2.

Upon proper fault simulation and analysis, the optimized amount of the reconfigurable redundancy can be determined prior to the fabrication of the RSoC. Thereby, both minimization of the die area overhead due to the redundancy and maximization of the RSoC yield can be achieved. Customized circuits can also be implemented by the reconfigurable redundancy, of course.

The following assumptions are made in this paper.

- RSoC is fabricated with embedded cores and each core can be tested and diagnosed as faulty or not.
- No escaped cores are considered (i.e., 100% test coverage is assumed).
- Repair process, including defective core isolation, redundancy reconfiguration, and interconnect reconfiguration can be applied to the RSoC.
- Each core may have an uneven number of ports which connects to other core(s) via intrachip interconnects.
- Reconfigured and rerouted interconnects are considered as less dependable than the original interconnects due to the complexity of the resulting interconnect configuration.

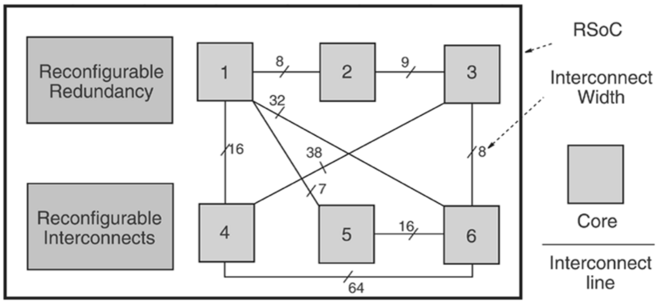

Fig. 1. RSoC model with reconfigurable resources.

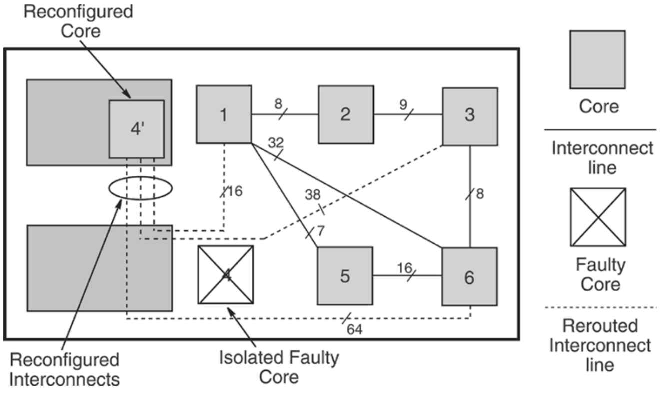

Fig. 2. Example of repaired RSoC.

As clearly addressed in the assumptions given above, the repair procedure of an RSoC has not only an advantage but also a disadvantage. Proper testing and diagnosis of the embedded cores and reconfigurable redundancy utilization may enhance the overall yield of the RSoC, since faulty cores can be replaced by reconfigured cores and rerouted reconfigurable interconnects. However, the reconfigured and rerouted interconnects may be less reliable due to the complexity of the resulting interconnect configuration. The unreliability associated with the reconfigured redundancy is modeled as the *unreliability impact factor*(uif). For example, in Fig. 1, repairing core 4 is assumed to affect neighboring (i.e., interconnected) cores 1, 3, 6, and associated interconnect structure. To accurately and effectively model the effect of both advantageous repair process and disadvantageous reconfigured and rerouted interconnects' unreliability at the same time, the following parameters are used to model the overall reliability of RSoC under repair:

N	number of cores in the RSoC;
$r(i)$	probability that the ith individual core is functioning, which is called *reliability*;
φ	maximum number of *repair* cycles;
rc	overall reliability of cores in the RSoC;
ri	overall reliability of interconnect structure;
r	overall reliability of the RSoC which takes into account both the cores and the interconnect structure;
uif	base unreliability impact factor due to the repair process penalty;
uif_{inc}	incremental rate of uif per each repair cycle;
$\text{cuif}(i,j)$	unreliability impact factor of the neighboring jth core due to repair of the ith core;
α	core unreliability impact factor coefficient. $0 \leq \alpha \leq 1$. It is fully dependent on the repair technology used. As $\alpha \rightarrow 1$, more reliability degradation due to the repair process is assumed to be applied to neighboring cores of the repair-candidate core;

$\lambda_{i,j}$	expected number of interconnect lines between the ith core and the jth core;
$\text{iuif}(i)$	interconnect structure unreliability impact factor due to repair of the ith core;
β	interconnect structure unreliability impact factor coefficient. $0 \leq \beta \leq 1$. It is fully dependent on the repair technology used. As $\beta \rightarrow 1$, more reliability degradation due to the repair process is assumed to be applied to the interconnect structure;
λ_i	expected number of interconnect lines of the ith core;
r_{inc}	reliability increase rate of a core due to repair;
$\eta(i)$	number of interconnect lines from the ith core;
$\eta(i,j)$	number of interconnect lines between the ith and the jth cores;
$F(\eta(i,j);\lambda_{i,j})$	cumulative Poisson probability function of $\eta(i,j)$ (i.e., $\sum_{y=0}^{\eta(i,j)}\left(e^{-y}\lambda_{i,j}^{y}/y!\right)$);
R	success rate of overall repair process, which is called *repairability* (e.g., if 8 out of 10 defective RSoCs are repaired during the repair process, $R = 8/10 = 80\%$);
Y	overall yield of RSoCs in which repair process is taken into account.

III. CONNECTIVITY-BASED RSOC REPAIR PROCESS

The repair process of a core enhances the overall reliability of the RSoC, but the process is also likely to introduce reliability degradation due to the complication of the reconfiguration process and is also prone to impair its neighboring (i.e., interconnected) cores' reliability since serious rerouting of connectivity would be experienced afterwards. Thus, the unreliability impact factor is modeled to be mainly determined by the number of interconnect lines between the repair-candidate core and its neighboring cores.

The characteristics of the repair process, so called *connectivity-based repair*, analyzed in this paper are given as follows:

1) The ith core is assumed to have initial reliability of $r(i)$. Then, rc (i.e., overall reliability of cores) of the given RSoC is initially determined by

$$rc = \prod_{i=1}^{N} r(i). \tag{1}$$

Then, the overall initial reliability of RSoC (denoted by r) is

$$r = rc \cdot ri \tag{2}$$

where ri is the reliability of the interconnect structure of RSoC.

2) The test and repair processes are performed after the fabrication phase.

3) Repair of a core degrades the reliability of the neighboring cores and the interconnect structure of the RSoC under repair. It is assumed that the unreliability impact factor (denoted by uif) due to the repetition of repair cycles increases as the RSoC undergoes a number of repair cycles. uif_n at the nth repair cycle is given as

$$\text{uif}_n = \text{uif}_{n-1} + (1 - \text{uif}_{n-1})\text{uif}_{\text{inc}} \tag{3}$$

where uif_{inc} is the incremental rate of uif at each repair cycle due to the increasing complexity of the repair process as the number of repair cycles increases.

4) The repair of the ith core is assumed to affect the neighboring (i.e., interconnected) core j, if it exists. The unreliability impact factor of the jth core due to the repair of the ith core is denoted by $\text{cuif}(i,j)$. $\text{cuif}(i,j)$ is modeled as a function of $\eta(i,j)$. The probability that the interconnect lines between the ith and the jth cores consist of exactly $\eta(i,j)$ lines is

$$P(\eta(i,j);\lambda_{i,j}) = \frac{e^{-\eta(i,j)}\lambda_{i,j}^{\eta(i,j)}}{\eta(i,j)!}. \tag{4}$$

The increase in the possibility of having more degradation due to an increment of one interconnect line from $\eta(i,j) - 1$ to $\eta(i,j)$ is also modeled to be determined by (4), since the occurrence of degrading repair is directly influenced by the number of interconnect lines attached to the repair candidate core. Thus, without loss of generality, the cumulative Poisson probability function of $\eta(i,j)$ (i.e., $\sum_{y=0}^{\eta(i,j)}\left(e^{-y}\lambda_{i,j}^{y}/y!\right)$) is the reasonable one to simulate an incremental rate of uif imposed by the number of interconnect lines between the ith and jth cores. Thus, unreliability impact factor of the neighboring jth core due to repair of the ith core is

$$\text{cuif}(i,j) = \text{uif}_n + (1 - \text{uif}_n) \cdot \alpha \cdot F(\eta(i,j);\lambda_{i,j}) \tag{5}$$

where α is a technology-dependent core unreliability impact factor coefficient and $F(\eta(i,j);\lambda_{i,j})$ is a cumulative Poisson probability function of $\eta(i,j)$. The parameter α simulates the efficiency of the repair process technology. As α approaches 1 and $\eta(i,j)$ increases, more reliability degradation is assumed to take place on the jth core since the $\alpha \cdot F(\eta(i,j);\lambda_{i,j})$ part approaches 1. Thus, the reliability of the jth core after the repair of the ith core can be formulated as

$$r(j)_n = r(j)_{n-1}(1 - \text{cuif}(i,j)_n). \tag{6}$$

5) Repair of the ith core is assumed to impair the interconnect structure as well. The reliability impact factor of the interconnect structure due to the repair of the ith core is denoted by

$$\text{iuif}(i)_n = \text{uif}_n + (1 - \text{uif}_n) \cdot \beta \cdot F(\eta(i);\lambda_i) \tag{7}$$

where β is a technology-dependent interconnect structure damage coefficient and $F(\eta(i);\lambda_i)$ is a cumulative *Poisson* probability function $\left(= \sum_{y=0}^{\eta(i)}\left(e^{-y}\lambda_i^{y}/y!\right)\right)$ which simulates the incremental rate of the reliability degradation due to the number of interconnect lines of the ith core. As β approaches 1 and $\eta(i)$ increases, more

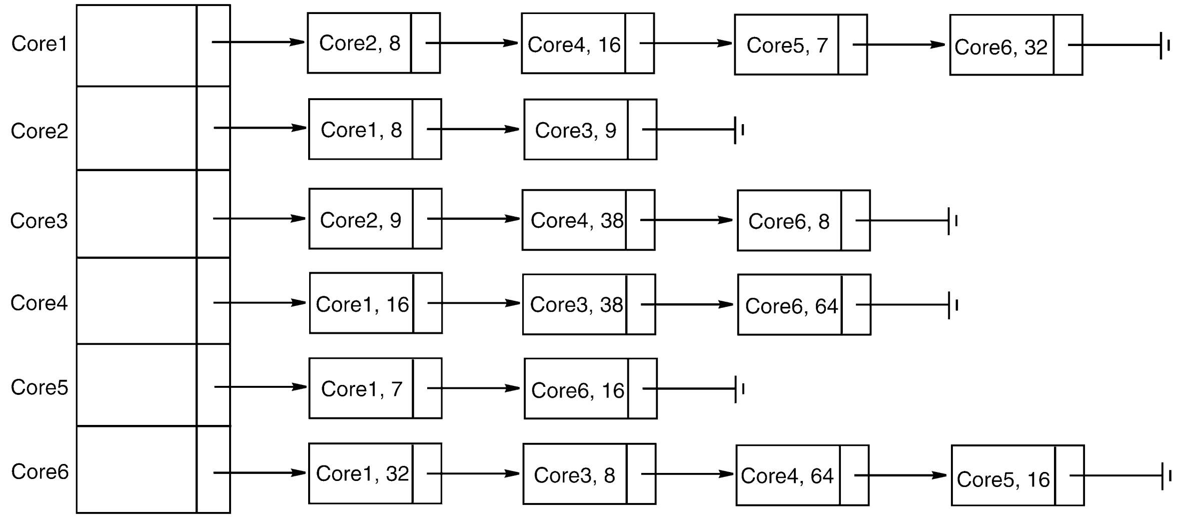

Fig. 3. Adjacency list representation of Fig. 1.

reliability degradation is assumed to be applied to the jth core since the $\beta \cdot F(\eta(i,j);\lambda_{i,j})$ part approaches 1.

6) The reliability of the ith core after the repair process is given by

$$r(i)_n = r(i)_{n-1} + (1 - r(i)_{n-1})r_{\text{inc}}. \tag{8}$$

7) The overall reliability of the cores on RSoC after the nth repair cycle becomes

$$rc_n = \prod_{i=1}^{N} r(i)_n. \tag{9}$$

8) The overall reliability of the interconnect structure on RSoC after the nth repair cycle can be formulated as

$$ri_n = ri_{n-1}(1 - \text{iuif}(i)_n). \tag{10}$$

9) The overall reliability of the RSoC after the nth repair cycle then becomes

$$r_n = rc_n \cdot ri_n. \tag{11}$$

IV. CONNECTIVITY-BASED RSOC REPAIR SCHEDULING

Every core on an RSoC is tested after the fabrication phase. The ith core can be tested and diagnosed as nonfaulty with the probability of $r(i)$ and as faulty with the probability of $\bar{r}(i)$. If there is only one faulty core detected during the test phase, the core will be isolated and repaired. If more than one faulty core is detected during the test phase, the order of repair (referred to as the *repair schedule*) must be properly arranged. In each repair cycle, in other words, selecting an appropriate repair-candidate core which has the least impact on the overall RSoC reliability is a natural choice for optimal scheduling.

The RSoC structure shown in Fig. 1 can be viewed as a weighted graph with six vertices and nine weighted edges. A simple way to represent the graph is to use a two-dimensional array called an *adjacency matrix* representation. The equivalent adjacency matrix of Fig. 1 is shown in Table I. The space requirement of the representation is $O(N^2)$ where N is the number of cores on the RSoC.

TABLE I
ADJACENCY MATRIX REPRESENTATION OF FIG. 1

	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6
Core 1	0	8	0	16	7	32
Core 2	8	0	9	0	0	0
Core 3	0	9	0	38	0	8
Core 4	16	0	38	0	0	64
Core 5	7	0	0	0	0	16
Core 6	32	0	8	64	16	0

If the RSoC is sparsely interconnected, a better solution is the *adjacency list* representation as shown in Fig. 3. The space requirement for this representation is $O(N + E)$, where N is the number of cores and E is the number of edges between cores on the RSoC. For RSoCs with a greater number of cores which are sparsely interconnected, the adjacency list representation can save the space requirement. For RSoCs with a fewer number of cores which are densely interconnected, the adjacency matrix is the choice. One of the two representations can be chosen accordingly, in practice.

For the proposed RSoC model, the number of interconnect lines and the number of neighboring cores attached to a repair-candidate core determines the resulting yield of the RSoC after each repair cycle. Four possible repair scheduling strategies are proposed as follows:

- *Minimum Number of Interconnects First (I-MIN)*—Among those diagnosed as faulty cores, the one which has the smallest number of interconnect lines is to be repaired first.
- *Maximum Number of Interconnects First (I-MAX)*—Among those diagnosed as faulty cores, the one which has the largest number of interconnect lines is to be repaired first.
- *Minimum Number of Neighboring Cores First (C-MIN)*—Among those diagnosed as faulty cores, the one which has the smallest number of neighboring cores is to be repaired first.

- *Maximum Number of Neighboring Cores First (C-MAX)*— Among those diagnosed as faulty cores, the one which has the largest number of neighboring cores is to be repaired first.

Since I-MAX and C-MAX repair scheduling strategies are supposed to repair the most reliability degrading core first, they do not have advantages in practice. However, they are also analyzed to be compared with the I-MIN and C-MIN repair scheduling strategies. The conceptual processes of I-MIN and C-MIN RSoC repair strategies are depicted in the flowchart shown in Fig. 4.

V. PARAMETRIC ANALYSIS

In this section, the effects of the connectivity-based RSoC repair scheduling are investigated through numerical experiments. An RSoC system with $N = 15$, $r(i) = 0.99$ for all i, and $ri = 0.9999$ is considered. The yield of the RSoC before an application of the repair process can be calculated as a series product of the $r(i)$ of all the cores [i.e., $\prod_{i=1}^{N} r(i)$] and the yield of the interconnect structure (i.e., ri). In Table II, the overall RSoC yield r and $\overline{r}$ (i.e., $1 - r$) are given where $\overline{r}$ is subdivided into six categories according to the number of defective cores on the RSoC (denoted by dc), in which $\overline{r}(dc = i)$ is inverse yield of RSoCs containing exactly i defective core(s). The following can be observed from Table II.

- Among those 14.0028% RSoCs with defects, 13.0298% of them have one defective core identified, 0.9213% of them have two defective cores identified, 0.000 403% of them have three defective cores identified, 0.000 012% of them have four defective cores identified, and 0.000 016% of them have more than five defective cores identified.
- Since RSoCs with $dc > 5$ are very few and then almost ignorable, the maximum allowed number of repair cycles $\varphi = 5$ is applied.
- 0.000 86% RSoCs in the category di (i.e., defective interconnect structure) does not have defective cores, but they have a defective interconnect structure. Since RSoCs in the di category are very few, no repair process is applied in this example.

To compare the proposed repair scheduling strategies, the values of r_{inc}, uif, and uif$_{\text{inc}}$ are set to 0.1 and the value of λ_i is set to 9, arbitrarily. In Tables III and IV, the repair performances of those proposed strategies, measured in the percentage of repaired RSoCs at each repair cycle, are shown. For example, 13.0298% of RSoCs contain one defective core and 62.4108% of them are repaired in the first repair cycle of I-MIN, and 0.9213% of RSoCs contain two defective cores and 27.4829% of them are repaired in the second repair cycle, and so on.

By comparing the results shown in Tables III and IV, the following can be observed.

1) Even with relatively small values of α and $\beta = 0.05$ (i.e., less core and interconnect degradation due to the repair process), the repair scheduling plays an important role in the RSoC repair process. Thus, it is shown that I-MIN and C-MIN outperform I-MAX and C-MAX at every repair cycle.

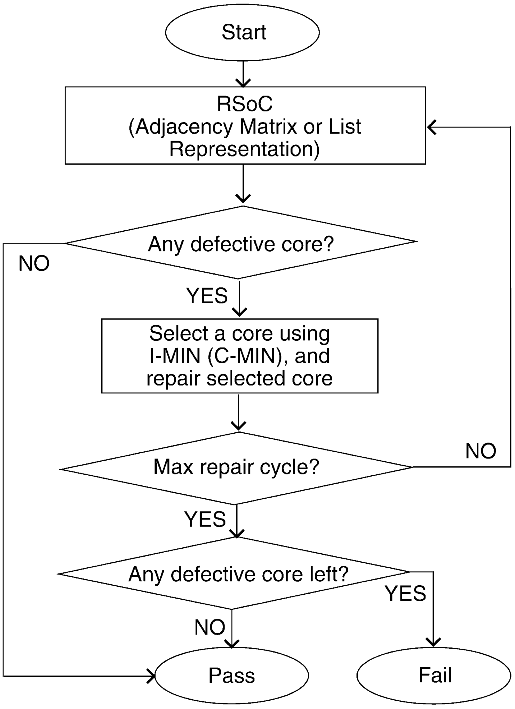

Fig. 4. I-MIN (C-MIN) flowchart.

TABLE II
r AND $\overline{r}$ OF THE GIVEN RSOC WITHOUT REPAIR

r	$\overline{r}$					
85.9972%	14.0028%					
	$\overline{r}(dc=1)$	$\overline{r}(dc=2)$	$\overline{r}(dc=3)$	$\overline{r}(dc=4)$	$\overline{r}(dc\geq 5)$	di
	13.0298%	0.9213%	0.000403%	0.000012%	0.000016%	0.000086%

TABLE III
PERFORMANCE COMPARISON OF THE PROPOSED REPAIR STRATEGIES AT EACH REPAIR CYCLE WHERE $\alpha, \beta = 0.05$

	$\varphi=1$	$\varphi=2$	$\varphi=3$	$\varphi=4$	$\varphi=5$
I-MIN	62.4108%	27.4829%	0.0661%	0%	0%
I-MAX	33.7273%	0.09249%	0.5%	0%	0%
C-MIN	67.2667%	33.2356%	8.1886%	0%	0%
C-MAX	36.3137%	9.2478%	1.4888%	0%	0%

TABLE IV
PERFORMANCE COMPARISON OF THE PROPOSED REPAIR STRATEGIES AT EACH REPAIR CYCLE WHERE $\alpha, \beta = 0.5$

	$\varphi=1$	$\varphi=2$	$\varphi=3$	$\varphi=4$	$\varphi=5$
I-MIN	59.3532%	14.6424%	0.2481%	0%	0%
I-MAX	0.7552%	0.0109%	0%	0%	0%
C-MIN	45.1135%	9.8665%	0.4963%	0%	0%
C-MAX	2.5941%	0.0868%	0%	0%	0%

2) As relatively larger values of α and $\beta = 0.5$ are applied (i.e., more core and interconnect degradation due to the repair process), the difference in the repairability between I-MIN (C-MIN) and I-MAX (C-MAX) becomes even more clear.

3) An appropriate selection of the repair schedule definitely affects repairability regardless of the values of α and β.

Repairability at each repair cycle is denoted by $R(\varphi)$, in which φ is the index of the repair cycle. Then, the overall success rate of the whole repair process, denoted by R, and can be calculated as follows:

$$R = \frac{\text{rate of repaired test-as-bad RSoCs}}{\text{rate of tested-as-bad RSoCs}} = \frac{\sum_{\varphi=1}^{n} \overline{r}(\varphi) \cdot R(\varphi)}{\overline{r}} \tag{12}$$

where n is the total number of repair cycles.

Repairability at each repair cycle (i.e., $R(\varphi)$, in which φ is the index of the repair cycle) and R of I-MIN and C-MIN repair strategies at different values of α and β are more extensively experimented with and the results are shown in Tables V and VI. The values of α and β are arbitrarily set to be equal for the simplicity of the analysis.

Upon the available values r, $\overline{r}$, and R, it is possible to calculate the overall yield of the RSoCs. The overall yield of the RSoCs denoted by Y can be calculated by

$$\underline{Y = r + \overline{r} \cdot R}. \tag{13}$$

In Figs. 5–8, Y (i.e., overall yield of the RSoC) of I-MIN at different values of N (i.e., 5, 10, and 15), α and β (i.e., 0.05, 0.1, 0.25, and 0.5), and $r(i)$ for all i (i.e., 0.8–1.0) are shown versus I-MAX. In Figs. 9–12, Y of C-MIN at different values of N (i.e., 5, 10, and 15), α and β (i.e., 0.05, 0.1, 0.25, and 0.5), and $r(i)$ for all i (i.e., 0.8–1.0) are shown versus C-MAX. By comparing the results of Figs. 5–12, the following observations can be drawn.

1) Using proper connectivity-based repair scheduling strategies (i.e., I-MIN and C-MIN), a higher Y of RSoC can be achieved.
2) I-MIN and C-MIN always outperform I-MAX and C-MAX.
3) As α and β increase, the difference between Y of I-MIN and Y of I-MAX increases. It is the same for C-MIN and C-MAX.
4) With relatively smaller α and β values, Y of both I-MIN and C-MIN perform similarly. However, I-MIN performs better than C-MIN as α and β increase.
5) In practice, C-MIN is likely to be the choice when smaller α and β values are applied, since it counts only the number of neighboring cores which is simpler than counting the number of interconnect lines.
6) In practice, I-MIN is likely to be the choice when larger α and β values are applied since it has less impact on Y of RSoC than C-MIN.

TABLE V
REPAIR PERFORMANCE OF I-MIN FOR THE GIVEN PARAMETERS

α, β	$R(\varphi=1)$	$R(\varphi=2)$	$R(\varphi=3)$	$R(\varphi=4)$	$R(\varphi=5)$	R
0.0	62.7546%	27.0378%	5.4590%	0%	0%	60.23%
0.2	61.3831%	21.3936%	1.7369%	0%	0%	58.57%
0.4	60.0254%	16.9108%	0.4962%	0%	0%	57.01%
0.6	58.6839%	12.7862%	0%	0%	0%	55.49%
0.8	57.3569%	9.6059%	0%	0%	0%	54.04%
1.0	56.0461%	7.1746%	0%	0%	0%	52.66%

TABLE VI
REPAIR PERFORMANCE OF C-MIN FOR THE GIVEN PARAMETERS

α, β	$R(\varphi=1)$	$R(\varphi=2)$	$R(\varphi=3)$	$R(\varphi=4)$	$R(\varphi=5)$	R
0.0	69.7278%	37.0888%	10.4218%	0%	0%	67.40%
0.2	59.8819%	23.3257%	3.7220%	0%	0%	57.31%
0.4	50.0361%	13.5135%	0.9925%	0%	0%	43.20%
0.6	40.1909%	6.9467%	0%	0%	0%	37.88%
0.8	30.3450%	2.9632%	0%	0%	0%	28.46%
1.0	20.4999%	0.9009%	0%	0%	0%	19.15%

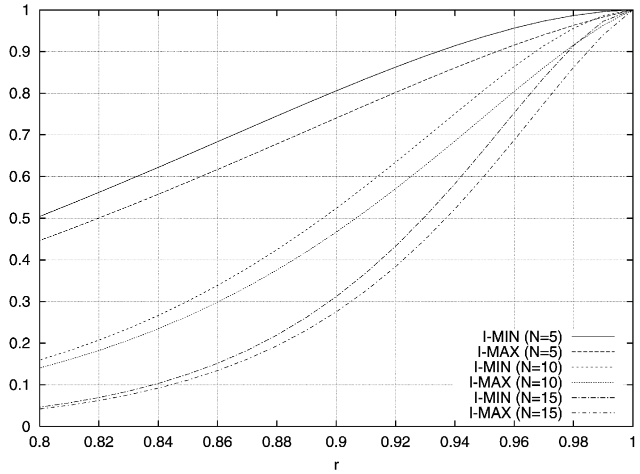

Fig. 5. Repairability of I-MIN and I-MAX at $\alpha, \beta = 0.05$.

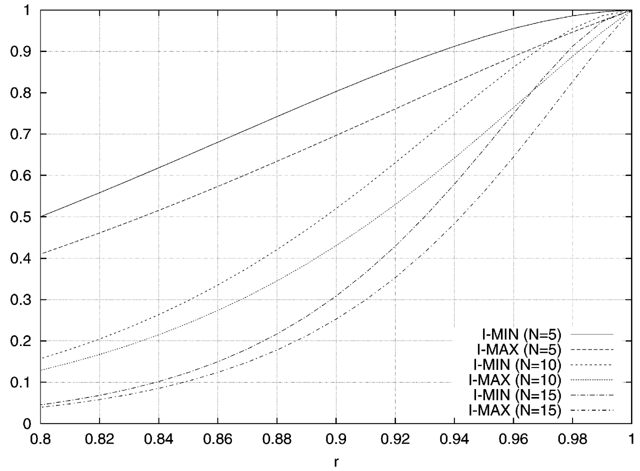

Fig. 6. Repairability of I-MIN and I-MAX at $\alpha, \beta = 0.1$.

VI. DISCUSSION

The overall complexity of SoC-based instrumentation is exponentially increasing as more cores are being embedded and the supporting interconnect structure is becoming more complex. At the same time, more efficient testing and repair of such devices are exigently required.

Thus, this paper has presented a new model for analyzing the repairability of RSoC instrumentation with repair processes based on the effect of the connectivity of the repair-candidate core where reliability degradation of both neighboring cores and interconnect structure due to the complexity of the reconfigured logic and interconnect redundancy is taken into account. Two approaches, I-MIN and C-MIN have been proposed. Two other scheduling policies, I-MAX and C-MAX also have been introduced and analyzed, and it has been shown

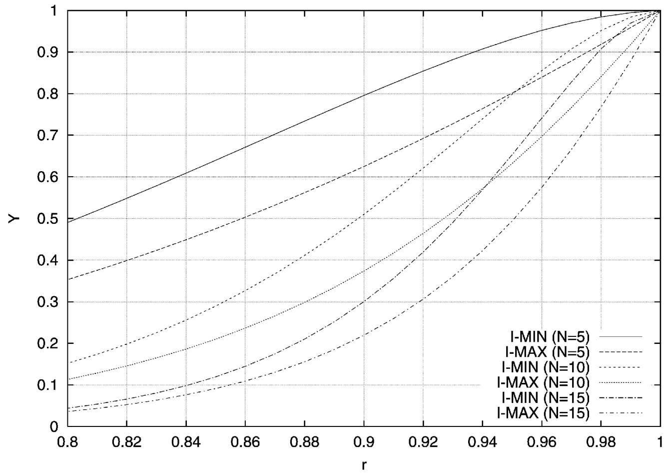

Fig. 7. Repairability of I-MIN and I-MAX at $\alpha, \beta = 0.25$.

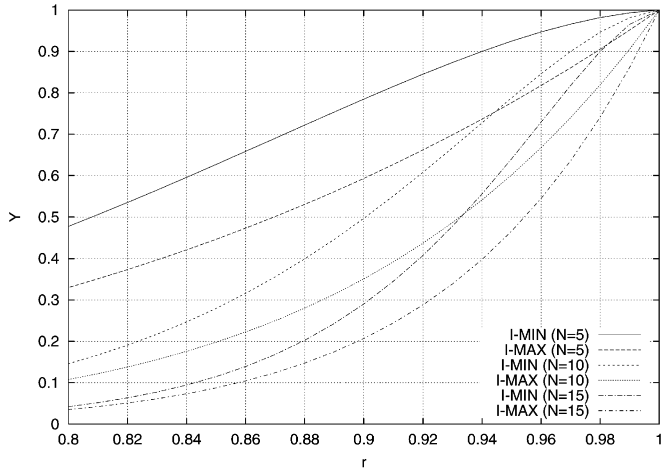

Fig. 8. Repairability of I-MIN and I-MAX at $\alpha, \beta = 0.5$.

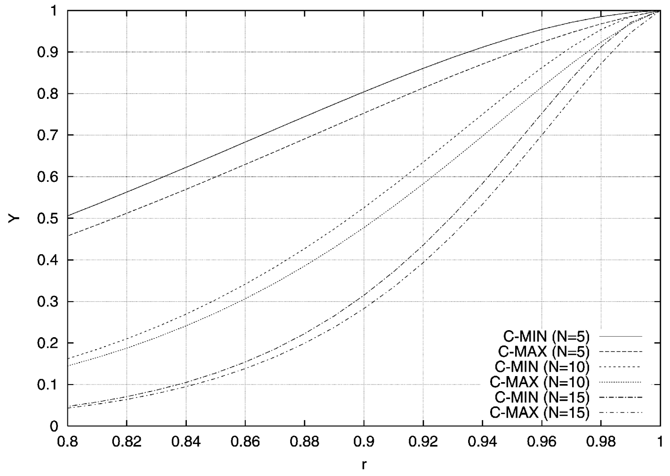

Fig. 9. Repairability of C-MIN and C-MAX at $\alpha, \beta = 0.05$.

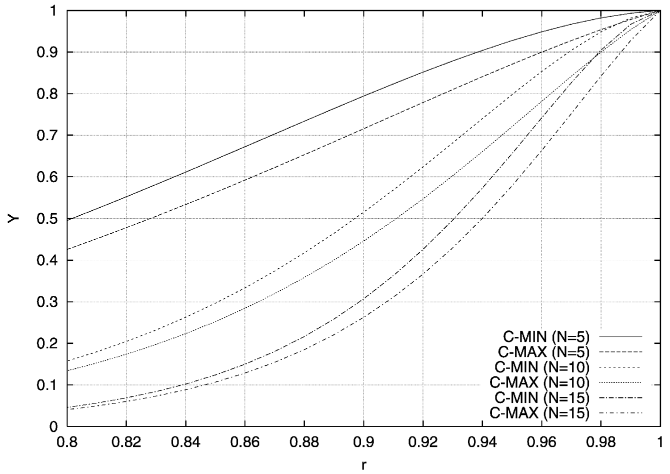

Fig. 10. Repairability of C-MIN and C-MAX at $\alpha, \beta = 0.1$.

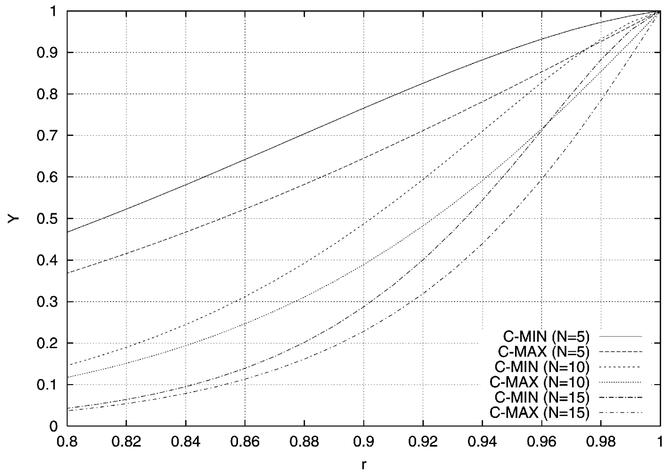

Fig. 11. Repairability of C-MIN and C-MAX at $\alpha, \beta = 0.25$.

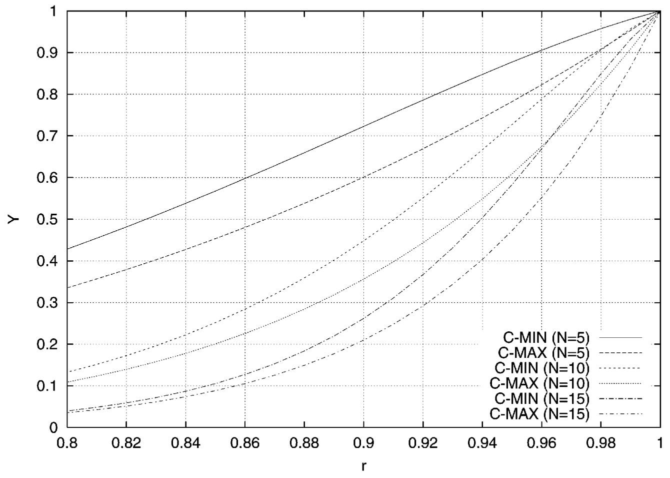

Fig. 12. Repairability of C-MIN and C-MAX at $\alpha, \beta = 0.5$.

how improperly scheduled repair processes could impair the overall repairability of RSoCs under repair. Extensive parametric analysis and comparison of the proposed approaches have demonstrated the efficiency of the proposed RSoC repair scheduling strategies (i.e., I-MIN and C-MIN). From the results, it is obvious that a higher repairability of RSoCs can be expected when proper RSoC repair scheduling strategies such as I-MIN and C-MIN are applied. Also, it has been shown that I-MIN tolerates a higher core and interconnect reliability degradation due to the repair process (i.e., higher α and β) than C-MIN does, while C-MIN results in a higher repairability when less core and interconnect reliability degradation due to the repair process (i.e., lower α and β) is assumed.

The effect of the connectivity of repair-candidate cores on the overall yield (i.e., ratio of the total number of functional RSoCs out of the total number of fabricated RSoCs) has been thoroughly investigated and various repair scheduling strategies have been proposed. Also, how improper repair scheduling could degrade the overall quality of repaired RSoCs will be extensively studied. Thus, results and findings from this research will be beneficial to RSoC-based digital instrumentation *developers*.

REFERENCES

[1] J. Greenbaum, "Reconfigurable logic in SoC systems," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 2002, pp. 5–8.

[2] S. Lee, S. Yoo, and K. Choi, "Reconfigurable SoC design with hierarchical FSM and synchronous dataflow model," in *Proc. Tenth Int. Symp. Hardware/Software Codesign*, May 2002, pp. 199–204.

[3] B. Lewis, I. Bolsens, R. Lauwereins, C. Wheddon, B. Gupta, and Y. Tanurhan, "Reconfigurable SoC—what will it look like," in *Proc. Design, Automation Test Europe Conf. Exhibition*, Mar. 2002, pp. 660–662.

[4] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, pp. 70–78, Jan. 2002.

[5] A. Nalamalpu, S. Srinivasan, and W. P. Burleson, "Boosters for driving long onchip interconnects—Design issues, interconnect synthesis, and comparison with repeaters," *IEEE Trans. Computer-Aided Design*, vol. 21, pp. 50–62, Jan. 2002.

[6] M. Choi, N. Park, F. Meyer, and F. Lombardi, "Connectivity-based multichip module repair," in *Proc. IEEE Pacific Rim Int. Symp. Dependable Computing*, Dec. 2001, pp. 19–26.

[7] R. Hartenstein, "Reconfigurable computing: A new business model-and its impact on SoC design," in *Proc. Euromicro Symp. Digital Systems Design*, Sep. 2001, pp. 103–110.

[8] R. A. Bergamaschi, S. Bhattacharya, R. Wagner, C. Fellenz, M. Muhlada, F. White, J.-M. Daveau, and W. R. Lee, "Automating the design of SoCs using cores," *IEEE Des. Test Comput.*, vol. 18, pp. 32–45, Sep.–Oct. 2001.

[9] M. Oberle, R. Reutemann, R. Hertle, and Q. Huang, "A 10-mW two-channel fully integrated system-on-chip for eddy-current position sensing [in biomedical devices]," *IEEE J. Solid-State Circuits*, vol. 37, pp. 916–925, Sep. 2001.

[10] B. I. Hounsell and T. Arslan, "Programmable multiplierless digital filter array for embedded SoC applications," *Electron. Lett.*, pp. 735–737, June. 2001.

[11] S. J. E. Wilton and R. Saleh, "Programmable logic IP cores in SoC design: Opportunities and challenges," in *Proc. IEEE Conf. Custom Integrated Circuits*, May 2001, pp. 63–66.

[12] T. Bautista and A. Nunez, "Quantitative study of the impact of design and synthesis options on processor core performance," in *Proc. 19th IEEE VLSI Test Symp.*, Apr.–May 2001, pp. 169–175.

[13] S.-Y. Chiang, "Foundries and the dawn of an open IP era," *IEEE Computer*, vol. 34, pp. 43–46, Apr. 2001.

[14] S. Ravi, G. Lakshminarayana, and N. K. Jha, "Testing of core-based systems-on-a-chip," *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 426–439, Mar. 2001.

[15] Z. Huang and S. Malik, "Managing dynamic reconfiguration overhead in systems-on-a-chip design using reconfigurable datapaths and optimized interconnection networks," in *Proc. Design, Automation Test Europe Conf. Exhibition 2001*, Mar. 2001, pp. 735–740.

[16] F. J. Meyer and N. Park, "Predicting the yield efficacy of a defect-tolerant embedded core," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Systems*, Oct. 2000, pp. 30–38.

[17] S. Dey, D. Panigrahi, L. Chen, C. N. Taylor, K. Sekar, and P. Sanchez, "Using a soft core in a SoC design: Experiences with picoJava," *IEEE Des. Test Comput.*, vol. 17, pp. 60–71, Jul.–Sep. 2000.

[18] I. Ghosh, S. Dey, and N. K. Jha, "A fast and low-cost testing technique for core-based system-chips," *IEEE Trans. Computer-Aided Design* , vol. 19, pp. 863–877, Aug. 2000.

[19] H. Singh, M.-H. Lee, G. Lu, F. J. Kurdahi, N. Bagherzadeh, and E. M. Chaves Filho, "MorphoSys: An integrated reconfigurable system for data-parallel and computation-intensive applications," *IEEE Trans. Comput.*, vol. 49, pp. 465–481, May 2000.

[20] S. Knapp and D. Tavana, "Field configurable system-on-chip device architecture," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 2000, pp. 155–158.

[21] R. K. Gupta and Y. Zorian, "Introducing core-based system design," *IEEE Des. Test Comput.*, vol. 14, pp. 15–25, Oct.–Dec. 1999.

[22] S. Winegarden, "Bus architecture of a system on a chip with user-configurable system logic," *IEEE J. Solid-State Circuits*, vol. 35, pp. 425–433, May 1999.

[23] A. M. Rincon, C. Cherochetti, J. A. Monzel, D. R. Stauffer, and M. T. Trick, "Core design and system-on-a-chip integration," *IEEE Des. Test Comput.*, vol. 14, pp. 26–35, Oct.–Dec. 1997.

Minsu Choi (M'02) received the B.S., M.S., and Ph.D. degrees in computer science from Oklahoma State University, Oklahoma City, in 1995, 1998, and 2002, respectively.

He is currently with the Department of Electrical and Computer Engineering, University of Missouri-Rolla as an Assistant Pprofessor. His research mainly focuses on computer architecture and VLSI, embedded systems, fault tolerance, testing, quality assurance, reliability modeling and analysis, configurable computing, parallel and distributed systems, dependable instrumentation and measurement, autonomic computing, and nanotechnology.

Dr. Choi was the recipient of the 2000 Don and Shelley Fisher Scholarship, the 2001 Korean Consulate Honor Scholarship, and the 2002 Graduate Research Excellence Award.

Nohpill Park (M'99) received the B.S. and M.S. degrees in computer science from Seoul National University, Seoul, Korea, in 1987 and 1989, respectively, and the Ph.D. degree from the Department of Computer Science,Texas A&M University, College Station, in 1997.

He is currently an Associate Professor in the Computer Science Department, Oklahoma State University, Oklahoma City. His research interests include computer architecture, defect and fault-tolerant systems, testing and quality assurance of digital systems, parallel and distributed computer systems, multichip module systems, programmable digital systems, and reliable digital instrumentation.

Vincenzo Piuri (F'01) received the Ph.D. degree in computer engineering from the Politecnico di Milano, Milan, Italy, in 1989.

From 1992 to September 2000, he was an Associate Professor of Operating Systems at the Politecnico di Milano. Since October 2000, he has been a Full Professor of Computer Engineering at the University of Milano, Milan, Italy. He was a Visiting Professor at the University of Texas at Austin during summers 1993–1999. His research interests include distributed and parallel computing systems, computer arithmetic, application-specific processing architectures, digital signal processing architectures, fault tolerance, neural network architectures, theory and industrial applications of neural techniques for identification, prediction, and control, and signal and image processing. His original results have been published in more than 150 papers in book chapters, international journals, and proceedings of international conferences.

Prof. Piuri is a member of ACM, the International Neural Network Society, and AEI. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and the *Journal of Systems Architecture*. He is Vice President for Publications of the IEEE Instrumentation and Measurement Society, Vice President for Member Activities of the IEEE Neural Networks Society, and a Member of the Administrative Committees of both the IEEE Instrumentation and Measurement Society and the IEEE Neural Networks Society.

Fabrizio Lombardi (M'82) received the B.Sc. (Hons.) degree in electronic engineering from the University of Essex, Essex, U.K., in 1977, the M.S. degree in microwaves and modern optics and the Diploma in Microwave Engineering, from the Microwave Research Unit, University College London, London, U.K., in 1978, and the Ph.D. degree from the University of London, London, U.K., in 1982.

He is currently the Chairperson of the Department of Electrical and Computer Engineering and holder of the International Test Conference (ITC) Endowed Professorship at Northeastern University, Boston, MA. Prior to this, he was a faculty member at Texas Tech University, Lubbock, the University of Colorado-Boulder, and Texas A&M University, College Station. His research interests are fault-tolerant computing, testing and design of digital systems, configurable computing, defect tolerance, and CAD VLSI. He has extensively published in these area and edited six books.

Dr. Lombardi received the Visiting Fellowship at the British Columbia Advanced System Institute, University of Victoria, Canada (1988), twice the TEES Research Fellowship (1991–1992, 1997–1998), the Halliburton Professorship (1995), and an International Research Award from the Ministry of Science and Education of Japan (1993–1999). He was the recipient of the 1985/1986 Research Initiation Award from the IEEE/Engineering Foundation, a Silver Quill Award from Motorola-Austin (1996), and a Distinguished Visitor of the IEEE Computer Society for the period 1990–1993. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTERS (1996–2000). Currently, he is the Associate Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTERS. He has been involved in organizing many international symposia, conferences, and workshops sponsored by organizations such as NATO and the IEEE, as well as acting as guest editor for archival journals and magazines such as the IEEE TRANSACTIONS ON COMPUTERS, IEEE MICRO, and IEEE DESIGN AND TEST OF COMPUTERS.