

A Progressive Approach for Fault Tolerance Improvement in Digital IPs

Tian BAN, Lirida NAVINER

{tian.ban,lirida.naviner}@telecom-paristech.fr

Institut TELECOM, TELECOM-ParisTech, LTCI-CNRS

46, rue Barrault, 75634 - Paris Cedex 13, France

Abstract

Reliability is becoming an important concern in nanoelectronic systems. Different fault-tolerant strategies to achieve reliable nanoelectronic architectures have been widely researched. Motivated by the need of economical designs, a progressive modular redundancy method is proposed to select the best subset among possible redundant architectures. It builds upon the block-grading concept that takes into account the inherent sensitivity and eligibility of constituent blocks of a digital circuit. Experiment results demonstrate the advantages and efficiency of the proposed method.

1. Introduction

Nanoelectronic systems face the challenge of providing reliable computation. Fault tolerant techniques have been widely used to correct errors of the faulty components and achieve reliability enhancement. A representative approach proposed by von Neumann is N-modular redundancy (NMR) [1].

However, hardware redundancy implies huge area overheads. Designers are constantly looking for methods to alleviate the performance degradation while keep the reliability improvement at an accepted level.

As digital circuits are consisted of a set of assembled sub-blocks, the reliability depends on the reliabilities of these individual blocks, and as well assembling rule of the sub-blocks. Therefore, it is very interesting to find the best candidates (sub-blocks) whose reliabilities play more important roles in the overall reliability. Under the same design constraint, protecting or improving reliabilities of these candidates would be more efficient in design-for-reliability.

In this work, constituent blocks are graded based on the analysis of their “**Sensitivity**” or “**Eligibility**”. Afterwards, modular redundancy adding is implemented on these ranked blocks in a progressive way.

The paper is organized as follows. Section 2 first introduces the concepts of sensitivity and eligibility, followed by how to grade blocks based on these two concepts. The method of progressive modular redundancy (PMR) is described in section 3. Section 4 presents examples of the PMR method, together with both results analysis and comparison. Finally, section 5 outlines some conclusions.

2. Block Grading Concept

2.1. Sensitivity and eligibility concept

Consider a circuit C with reliability R and being constituted of K independent blocks b_i . Let $B = b_1, b_2, \dots, b_K$ be the set of K blocks in C and is $Q = q_1, q_2, \dots, q_K$ the set of their respective reliabilities (i.e. is q_i the reliability of b_i).

We define the **sensitivity** of C 's reliability with respect to b_i 's reliability as a metric giving impacts on R by changing q_i . i.e., partial derivative of function R with respect to variable q_i , expressed as $s(b_i) = \left| \frac{\partial R}{\partial q_i} \right|$. We

define the **eligibility** of a block b_i , noted $e(b_i)$, as the metric expressing how reliability improvement of this block is meaningful. Let us denote $\Delta_i = |R_i^* - R|$ as the reliability change resulting from q_i improvement. The eligibilities of two blocks b_i and b_j are then defined such that they satisfy $R_i^* \geq R_j^* \Leftrightarrow e(b_i) \geq e(b_j)$. For a circuit such as C , $e(b_i)$ are integers in $[1, K]$, where 1 and K represent the less and the most eligible blocks, respectively. Eligibility values depend on the techniques adopted to improve the reliability of the blocks.

Sensitivity and eligibility are two concepts closely related, but addressing different analysis objectives. Sensitivity is useful to determine which blocks are critical in C , which are blocks whose reliability degradation would lead to significant degradation of C 's reliability. Eligibility determines the order in which blocks should have their reliability improved to obtain the best gain in C 's reliability. Therefore, these two concepts imply the relationship between design requirement and block grading, and therefore reveal opportunities for more efficient design according to different block weights W_i .

2.2. Examples of grading

Sensitivity and eligibility analysis could be obtained in a mathematical way if the circuit reliability could be expressed by a closed-form equation, as addressed in [2]. In general cases, sensitivity and eligibility analysis are realized by simulations.

1) Experimental metric for sensitivity classification: In order to get the rank of the blocks by their sensitivities, a fault simulation platform [3] is used to inject faults due to Single Event Upsets (SEUs). The results produced by the original and the faulty circuits are compared by a fault injection and fault analysis platform. If these results are different, it is concluded that the effects of the fault injected have been propagated to the outputs. On the contrary, it is considered that the fault has been masked (there is a "mask").

Each b_i contributes to a number of masks, noted as m_i , which is obtained by calculating the number of masks after fault injection based on the hypothesis that only the corresponding block b_i is faulty. In this way, each m_i is directly related to the sensitivity of block b_i . The block corresponding to the minimum number of masks is defined as the most sensitive block, i.e., $s(b_i) \geq s(b_j) \Leftrightarrow m_i \leq m_j$. An example is the implementation of the ISCAS 85' benchmark C17 [4]. The results are in Tab. 1.

2) Experimental metric for eligibility classification: Eligibilities of the constituent blocks can be obtained by comparing the results of $\Delta_i = |R_i^* - R|$ for each block b_i considering the same technique. The highest Δ_i value indicates that the block b_i has the highest eligibility, that is, $e_i = K$. We still consider circuit C17, each NAND gate is supposed to have reliability $q_i = q = 0.99$. After TMR implementation on each NAND gate, reliability improvement results are those shown in Tab. 1, thus e_i is also decided here.

Tab.1 - Results of sensitivity and eligibility rank of C17

b_i	m_i	R_i^*	e_i
NAND1	12	95.765%	2
NAND2	8	95.882%	3
NAND3	2	95.065%	4
NAND4	12	95.763%	1
NAND5	0	96.117%	6
NAND6	0	96.114%	5

3. Progressive Modular Redundancy Approach

3.1. The workflow

As a matter of fact, reliability enhancement problem is addressed as an optimization task where two variables should be kept in balance: the reliability improvement ΔR and the overhead ΔC generated by this reliability improvement. The analysis of this problem depends on the improvement objective: 1) Determine the architecture that can satisfy a constraint of reliability, while minimizing the overhead. It means minimizing of ΔC , while respecting $R \geq R_{\min}$. 2) Determine the best gain in reliability due to a maximum-tolerant additional cost. It means maximizing of ΔR , while respecting $\Delta C \leq \Delta C_{\max}$.

Benefitting from the ranks of graded constituent blocks, we propose to solve this problem step by step, which is in a progressive way. The basic idea is to act progressively on the blocks: 1) Starting with improving the reliability of a single block, then two blocks, and so on until cover all the blocks. 2) Starting with improving the reliability of the blocks with inexpensive technique, then gradually moves to techniques increasingly expensive (and so increasingly efficient).

On the other hand, weights of each block that determine the order for executing in blocks are already possible from sensitivity and eligibility analysis. For reliability improvements, $W_i = e_i$. Assume the design criterion is to reach a required reliability R_{req} , while respecting the overhead should be $\Delta C \leq \Delta C_{\max}$.

The workflow corresponding to the PMR method is described in Fig.1. Main functions and procedures are explained below. **Description** defines how the blocks b_i are Fig. 1, connected together to form the circuit. This can be a structural HDL description of C . **Library** provides useful parameters of each block b_i , like area A_i , power consumption P_i , delay T_i and reliability q_i . **RA(C)** refers to reliability analysis of the original circuit C . It is done to check whether C already meets the constraints. If $R \geq R_{req}$, the original C is selected. **BG** grades the blocks according to their different weights. All the constituent blocks are then donated with priorities (W_i)

to decide their priorities when adding redundancy. **PI** stands for reliability improvement with progressive modular redundancy. Improvements are implemented in an iterative way. (*PTMR* and *PMMR*, as will be explained in next sections.)

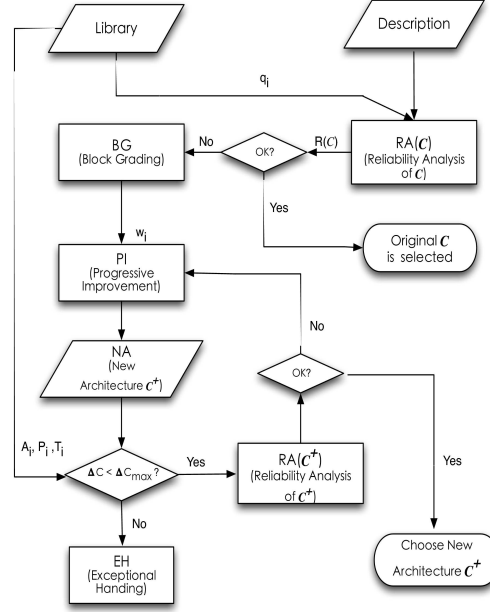


Fig. 1 – The workflow

NA are new versions of C related to use of progressive TMR (C_{PTMR}) or progressive MMR (C_{PMMR}). $RA(C^+)$ refers to reliability analysis of the new architectures that have passed cost constraint checking. **EH** functions when both of the new architectures C_{PTMR} and C_{PMMR} exceed the area cost limit. It either relaxes ΔC_{max} or selects the architecture of the last iteration.

3.2. Progressive triple modular redundancy: PTMR

Now we present the method for the procedure **PI** in Fig.1. Considering NMR in this work, we investigate the reliability impact of having $N = 3$ (TMR) and $N = 5$ (QMR), respectively. We first begin with TMR. The idea is to implement TMR first on the blocks that have higher weights (eligibilities) and then (if necessary) on the blocks with lower weights. The values W_i that are required for the proposed method can be obtained from eligibility analysis. Notice that sometimes W_i come directly from circuit's topology.

According to the proposed method, only the block with the highest W_i benefits from redundancy adding in the first step. If the obtained reliability improvement is considered insufficient compared with the requirements, a second block is considered for redundancy adding (this is the block with the second highest W_i value). The procedure is carried on until the reliability requirement is satisfied or maximum redundancy is used. Table 1 in [5] shows how new architectures are produced under this progressive method.

3.3. Progressive mixed modular redundancy: PMMR

We have also explored the use of combined Triple and Quintuple modular redundancy with progressive redundancy adding. This involves the progressive mixed modular redundancy (PMMR), which is an extension of PTMR. Table 2 in [5] shows the execution of PMMR.

Considering the ordered set of blocks, the first one has weight K and the last one has weight 1 . In addition to architecture based on PTMR, PMMR approach produces a new architecture as follows. When m is odd, this comes from QMR on the first p blocks ($p = \left\lfloor \frac{m}{2} \right\rfloor$) and TMR on the next r blocks ($r = m - p$). When m is even, it results in only QMR on the first p blocks. Advantages of PMMR will be found in the next section.

4. Implementations and Results

In this section, we present two implementations of the proposed progressive redundancy adding method. Reliability evaluation is performed with SPRMP tool proposed in [6].

1) PTMR implementation: The PTMR approach has been applied on circuit C17. Each NAND gate is supposed to have reliability value $q_i = q = 0.99$ and the area cost S_{NAND} . Requirements of the design are as follows: minimum required reliability is $R_{req} = 0.97$ and maximum redundancy area overhead is set as $4 \bullet S_{NAND}$. After TMR implementation on each NAND gate, block grading results are those shown in Tab.1. As shown in Tab.2, in the first step, $RA(C) = 95.20\% \leq R_{req}$. In the same way, implementation of TMR only on the block NAND5 ($W_5 = 6$) is still insufficient. Finally, TMR on NAND5 and NAND6 in the second step, $RA(C^+) = 97.047\%$ which satisfies the design requirements. Although the total number of candidate architectures under the same area cost is $C_6^2 = 15$, we find a shortcut to reach the best one.

2) PMMR implementation: We implemented the PMMR approach on the 8-bit ripple carry adder (RCA-8). The basic blocks (FA) are supposed to have reliability value $q_i = q = 0.999$ and the area cost S_{FA} . Requirements of the project are as follows: minimum required reliability is $R_{req} = 0.955$ and maximum area overhead is set as $4 \bullet S_{FA}$. The results are presented in Tab.3.

The adder here has a cascade structure and all blocks have the same reliability, block eligibilities are given as $s(i) = w(FA_i) = 8 - i + 1$. Initially, $RA(C) = 94.06\% \leq R_{req}$. The first iteration generates only one architecture with TMR on FA1. The reliability of this new architecture is $R = 94.85\% \leq R_{req}$. The second iteration generates two architectures; one is TMR on FA1 and FA2, another is QMR on FA1. Both architectures satisfy reliability requirements, so we select the architecture that results in less area overhead ($4S_{FA} + S_V$). Notice that mixed modular redundancy is a better solution than TMR in this case. MMR brings into higher reliability. Furthermore, if R_{req} is set higher, for example, 96%, we don't need further iterations, either.

A similar technique named selective TMR has been proposed in [7], where selective insertion of TMR is implemented on the blocks that are considered sensitive. It ignores the fact that reliability of the circuit highly relates with the combinational logic. Block grading concept in this work takes combinational logic masking into account and shows that PTMR is more efficient than STMR. For example, in the circuit C17, STMR decides to implement TMR on gates NAND3, 4, 5 and 6. However, under the same area cost, the PTMR chooses gates NAND5, 6, 3 and 2 to implement TMR, which increases the overall reliability.

Tab.2 - Results of PTMR on C17

Step	Architecture	Reliability	Area Overhead
0	1-1-1-1-1-1	95.20%	0
1	3-1-1-1-1-1	96.12%	$2S_{NAND} + S_V$
2	3-3-1-1-1-1	97.05%	$4S_{NAND} + 2S_V$

Tab.3 - Results of PMMR on RCA-8

Step	Architecture	Reliability	Area Overhead
0	1-1-1-1-1-1-1-1	94.06%	0
1	3-1-1-1-1-1-1-1	94.85%	$2S_{FA} + S_V$
2	3-3-1-1-1-1-1-1	95.64%	$4S_{FA} + 2S_V$
	5-1-1-1-1-1-1-1	97.14%	$4S_{FA} + S_V$

5. Conclusion

The effective method that was presented in this paper is adaptable with all existing fault-tolerant techniques based on redundancy. Blocks are graded by their significance and then redundancy techniques could realize a more efficient performance. As many large circuits contain a limited number of components that are used

repeatedly, block grading could then be carried out in a similar modular approach. Designers could make more judicious decisions and unnecessary overhead is avoided.

6. Acknowledgement

The authors would like to thank STIC AmSud Program for the financial support given to this work.

7. References

- [1] J. von Neumann, "Probabilistic logics and synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. Shannon and J. McCarthy, Eds. Princeton University Press, 1956, pp. 43–98.
- [2] L. Naviner, J.-F. Naviner, T. Ban, and G. Junior, "Reliability analysis based on significance," in *Argentine School of Micro-Nanoelectronics Technology and Applications (EAMTA)*, 2011, aug. 2011, pp. 1–7.
- [3] L. Naviner, J.-F. Naviner, G. Jr., E. Marques, and N. Jr., "Fifa: A fault-injection-fault-analysis-based tool for reliability assessment at rtl level," *Microelectronics Reliability*, vol. 51, no. 9-11, pp. 1459–1463, 2011.
- [4] "ISCAS'85 Benchmark Circuits Information [Online]." Available: <http://www.cbl.ncsu.edu/benchmarks/ISCAS85/>
- [5] T. Ban and L. Naviner, "Progressive module redundancy for fault-tolerant designs in nanoelectronics," *Microelectronics Reliability*, vol. 51, no. 9-11, pp. 1489–1492, 2011.
- [6] D. T. Franco, M. C. Vasconcelos, L. Naviner, and J.-F. Naviner, "Signal probability for reliability evaluation of logic circuits," *Microelectronics Reliability*, vol. 48, no. 8-9, pp. 1586–1591, 2008.
- [7] P. Samudrala, J. Ramos, and S. Katkoori, "Selective triple modular redundancy based single-event upset (seu) tolerant synthesis for fpgas," *Nuclear Science, IEEE Transactions on*, vol. 51, no. 5, pp. 2957–2969, 2004.