

# Fault tolerance in an amplifier system implemented in reconfigurable system on chip platform

Mónica Lovay, Gabriela Peretti, Eduardo Romero

Grupo de Estudio en Calidad en Mecatrónica  
 Facultad Regional Villa María,  
 Universidad Tecnológica Nacional  
 Villa María, Argentina  
 gecam@frvm.utn.edu.ar

Carlos Marqués

Grupo de Desarrollo Electrónico e Instrumental  
 Facultad de Matemática, Astronomía y Física,  
 Universidad Nacional de Córdoba  
 Córdoba, Argentina  
 marques@famaf.unc.edu.ar

**Abstract**—This work address the problem of providing fault tolerance to an analog system embedded in a commercial programmable system on chip. The system presents a functionality that has to be maintained despite the presence of faults, without direct human intervention. For detecting a gain fault, we use a built-in self-test strategy that establishes the actual values of gain achievable by the system. A simulated annealing (SA) algorithm finds the hardware configuration. The simulation results show that the strategy is able to maintain its functionality under the presence of catastrophic and deviation faults. In addition, SA presents better performance than an exhaustive search method.

**Keywords:** *hardware fault tolerance, amplifier system, simulated annealing algorithm*

## I. INTRODUCTION

The need for fault tolerance in hardware is an important issue for critical safety applications or for electronics systems that have to operate in environments where maintenance is difficult to achieve. The use of redundant hardware, i.e. the exchange of a faulty component with an operating spare, is the traditional way for obtaining hardware fault recovery. Assuming that a fault detection test strategy is available, field programmable gate arrays (FPGAs), field programmable analog arrays (FPAAs) and programmable system on chip offer an alternative to traditional fault tolerant schemes because the reconfigurable nature of these devices enables runtime correction [1]. Additionally, although reconfiguration does not always guarantee that complete functionality can be restored, does allows maintaining the operation with a slight degradation of the system [2] and is an alternative for systems with limited free space [3].

In the literature, many researchers exploit the use of reconfiguration for tolerating hardware faults. One alternative is the generation of different versions of logic placement-and-routing information for the same FPGA application circuitry. Once a faulty region is located the system switches between different configurations [4], [5] or use partial reconfiguration [6]. By other way, evolvable hardware [7] combines reconfigurable hardware with evolutionary algorithms. In this methodology, usually a genetic algorithm searches the

possible hardware configurations that present the better performances once a fault is present. Among others, different schemes have been proposed FPGAs [8], FPAAs [2], and for programmable system on chip [9]. In addition to genetic algorithms, the search of possible hardware reconfiguration can be made by using other algorithms. One of them is simulated annealing, one of the first algorithms that extend local search methods with an explicit strategy to escape from local optima [10]. SA is still object of further studies, is used in optimization problems [11], [12], and is component of other algorithms [13].

In this work, we address the problem of providing fault tolerance to an analog system embedded in a commercial programmable system on chip. The system presents a functionality that has to be maintained despite the presence of faults, without direct human intervention. For detecting a gain fault, we use a built-in self-test (BIST) strategy that establishes the actual values of gain achievable by the system. We adopt a SA algorithm for searching the hardware configuration, with the aim of comparing the strategy with the one addressed in [9], which uses an evolvable hardware strategy.

## II. SYSTEM DESCRIPTION

PSoC® device is a programmable system-on-chip platform with an on-chip processor core [14]. It includes configurable blocks of analog and digital circuits, programmable interconnect and configurable IO in a low-cost chip. Analog functions in the PSOC device are organized as groups of general-purpose analog blocks that can be configured into user-determined functions. The control for these blocks is register-based and can be programmed through the design tools or reprogrammed by the user at run-time. Some of the available configurations for the analog arrays are up to 14 bits analog to digital converters (ADC), up to 9 bits digital to analog converters (DAC), and programmable gain amplifiers (PGA).

The amplifier system addressed in this work employs four PGAs. The PGA user module implements a non-inverting amplifier with user-programmable gain, which is established

by an array of resistances (Fig. 1). This amplifier has 33 programmable values for the gain, ranging from 0.062 to 48.

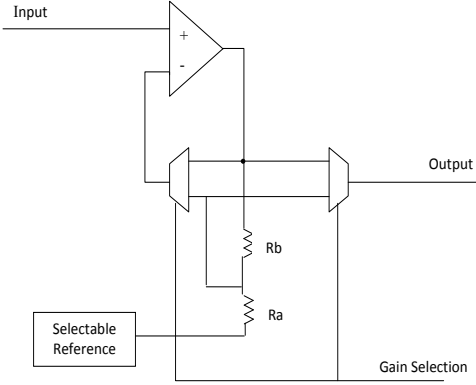


Figure 1. Programmable gain amplifier available in the PSoC® device (simplified diagram)

Fig. 2 shows the amplifier system in normal mode. The four-amplifier chain (PGA1, PGA2, PGA3 and PGA4) is configured in the PSoC® CY8C27443-24PXI.

In this work, it is considered that the redundancy necessary for fault tolerance comes from the multiple values of gain of every amplifier, the use of four amplifiers in the amplifying chain and runtime configuration [15]. A BIST strategy, described in [9], tests the gain of each amplifier during the dead times of the system. If the test process finds a degradation in the overall gain, then establishes that is necessary a system reconfiguration. The BIST strategy is implemented as a new hardware configuration (using on-chip analog resources) that is dynamically loaded while the device is running. The reconfiguration makes use of a SA algorithm running in an external computer. SA find the gain values of the four amplifiers with the goal of maintaining the system overall gain within specifications. The new values of gain are loaded back into the hardware for continuing the normal operation (Fig. 2).

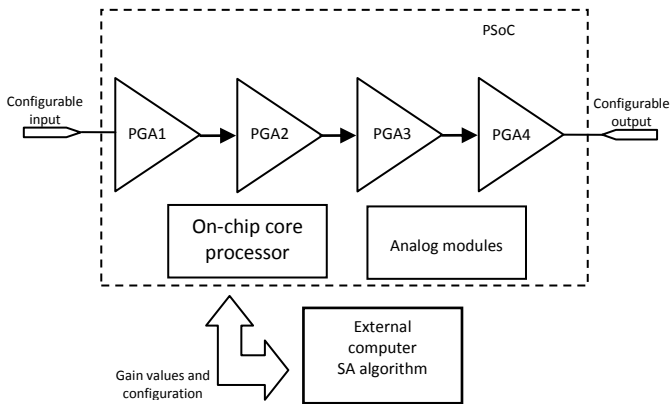


Figure 2. Amplifier system diagram, normal mode hardware configuration

### III. OVERVIEW OF SA AND FORMULATION OF THE OPTIMIZATION PROBLEM

The SA algorithm was originally inspired by formation of crystal in solids during cooling i.e., the physical annealing with solids. The foundation and details of SA can be found elsewhere [10].

In this work, SA has to find the four PGA gain values ( $G_1$  for PGA1,  $G_2$  for PGA2,  $G_3$  for PGA3, and  $G_4$  for PGA4, Fig. 1) that reach the target (desired) gain  $A_{tar}$ . We propose three different values for  $A_{tar}$ : 2, 8 and 15. The aim is to evaluate the ability of SA for finding an acceptable solution in different scenarios. The optimization problem is formulated as follows: the reconfiguration algorithm has to find the values of  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$  that reach the condition:

$$\text{Min}(|A_{tar} - G_1 \cdot G_2 \cdot G_3 \cdot G_4|). \quad (1)$$

The SA algorithm begins with an initial solution that is randomly generated, with an initial temperature parameter,  $T$ . At each iteration, SA compares the values for two solutions, the current and a newly selected in the neighborhood of the actual solution. Improving solutions are always accepted, while non-improving solutions are retained with a probability that depends on  $T$ . The algorithm stops when finds a solution that fulfill the requirements or when reaches a maximum number of iterations. In our work, the initial temperature ( $T_0$ ) is 500, and the maximum number of iterations is 200. The function that is used to update the temperature  $T$  in each iteration  $i$  is the following:

$$T = T_0 / i. \quad (2)$$

### IV. FAULT MODELS USED FOR VALIDATION

The performance of the fault tolerance scheme presented here is evaluated by means of fault injection. Consequently, it is necessary to define a fault model.

If the PGA is well designed, the operational amplifier can present wide deviations in its functional parameters without effects in its closed loop performance. As a result, we consider that the main cause of PGA gain faults comes from faults or degradations in the resistances that establish the gain ( $R_a$  and  $R_b$  in Fig. 1). In each PGA, we consider two different types of faults in the gain determined by its array of resistances. The first one is a catastrophic fault that assume that is not possible to establish one gain value. The second fault is a deviation in the gain values. For deviation faults, we consider that individual gains  $G_1$ ,  $G_2$ ,  $G_3$  and  $G_4$ , deviate their values in a percentage of their nominal values,  $\pm 10\%$ ,  $\pm 20\%$ ,  $\pm 30\%$ ,  $\pm 40\%$  and  $\pm 50\%$ .

### V. EXPERIMENTAL RESULTS

#### A. Fault Free Operation

Fig. 3 shows the relative error for the three target gains (2, 8 and 15) in several runs of SA. The three target gains present relative errors in the range  $[-2,893\%, 3,518\%]$ . Each run is a solution to the optimization problem changing the seed for the random generation of the initial solution.

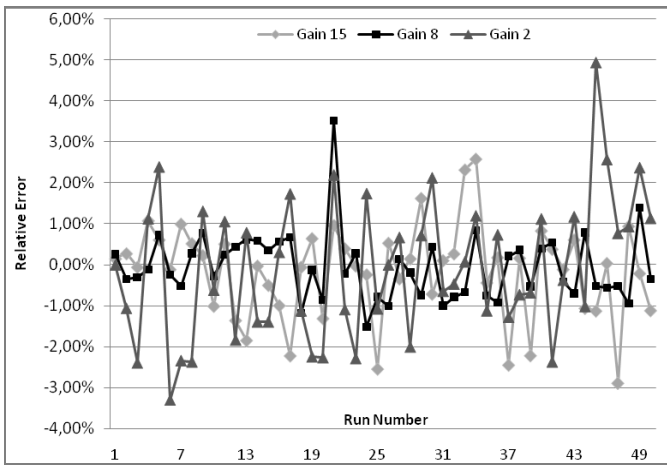


Figure 3. Errors in the target gain. Fault-free operation

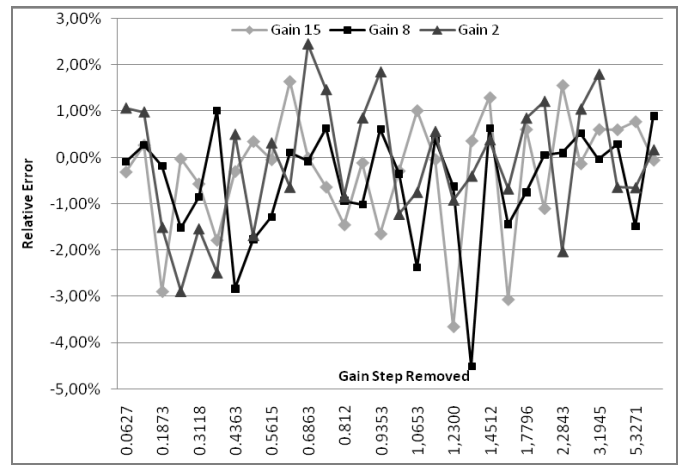


Figure 4. Errors in the target gain under catastrophic fault condition in PGA1

Table I summarizes the relative error characterization for all the gains. In the following, as a measurement of a central tendency, we use the median of the relative errors because the data distribution is not normal. As a measurement of dispersion, we use the range of the error in order to take into account extreme values. We observe that the median and the error range of the target gain 15 are both higher than the values obtained for the other two gains. Target gain 8 presents lower median and higher range than gain 2.

TABLE I. GAIN ERROR CHARACTERIZATION UNDER FAULT-FREE CONDITIONS

Target gain	Median	Minimum error	Maximum error	Error range
15	0,004%	-2,893%	2,583%	5,476%
8	-0,213%	-1,523%	3,518%	5,041%
2	-0,145%	-2,481%	2,471%	4,952%

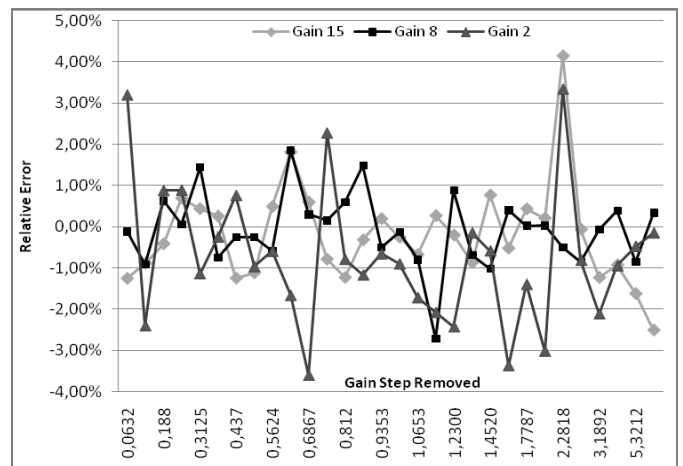


Figure 5. Errors in the target gain under catastrophic fault condition in PGA2

**B. Operation Under Fault Condition**

Figs. 4 to 7 depict the results obtained under catastrophic fault condition. These figures show the relative error for the three target gains versus the removed gain value in the corresponding PGA. In all the experiments, the SA is capable of reaching the target gain, with errors for all the gains in the range [-4,513%, 4,154%].

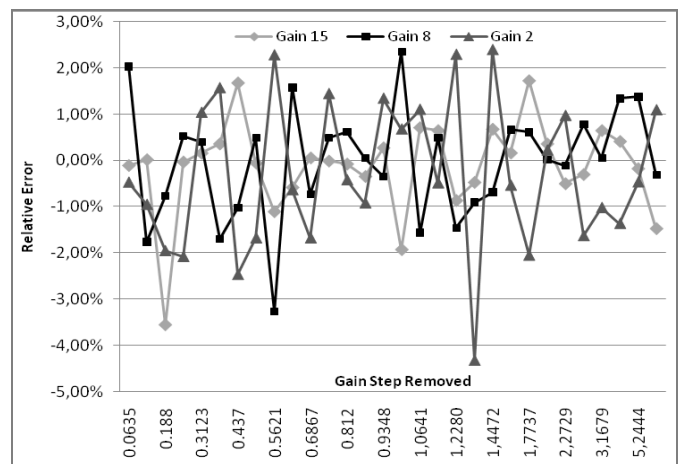


Figure 6. Errors in the target gain under catastrophic fault condition in PGA3

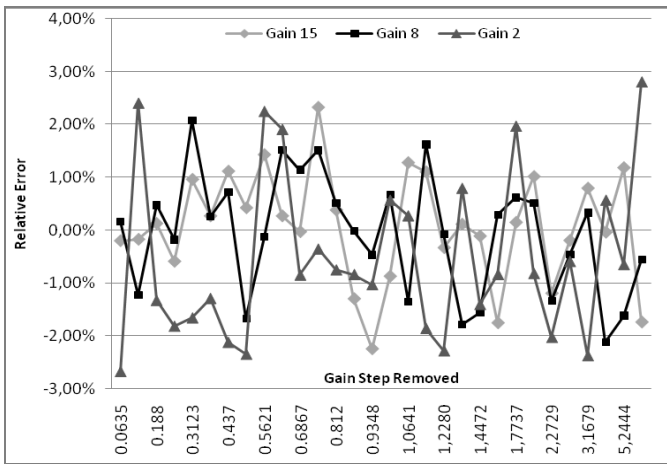


Fig. 7. Errors in the target gain under catastrophic fault condition in PGA4

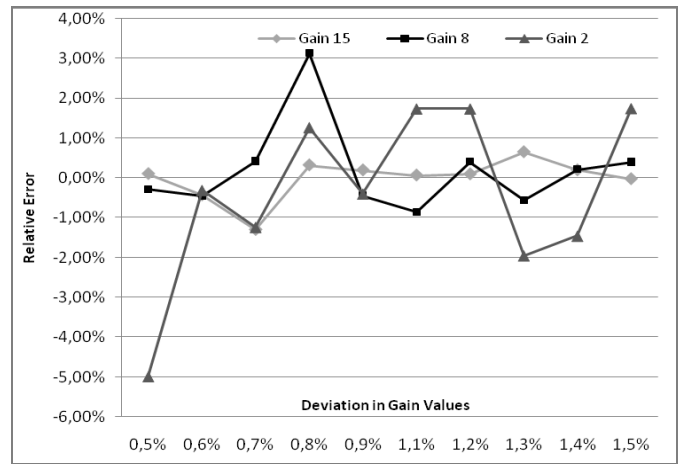


Fig. 9. Errors in the target gain under deviation fault condition in PGA2

Table II summarizes the effects of catastrophic faults in the four PGAs. Comparing the normal (Table I) and catastrophic fault operation (Table II), the faulty system presents as a worst case an increase of 2,329% in the error range for gain 15. For gains 2 and 15, the median is lower than the median in normal operation. For gain 8 the median is higher than the median in normal operation, suggesting in all cases a change in the error distribution between the normal and faulty operation.

TABLE II. GAIN ERROR CHARACTERIZATION UNDER CATASTROPHIC FAULT CONDITION

Target gain	Median	Minimum error	Maximum error	Error range
15	-0,043%	-3,651%	4,154%	7,805%
8	-0,056%	-4,513%	2,344%	6,857%
2	-0,651%	-3,608%	3,345%	6,953%

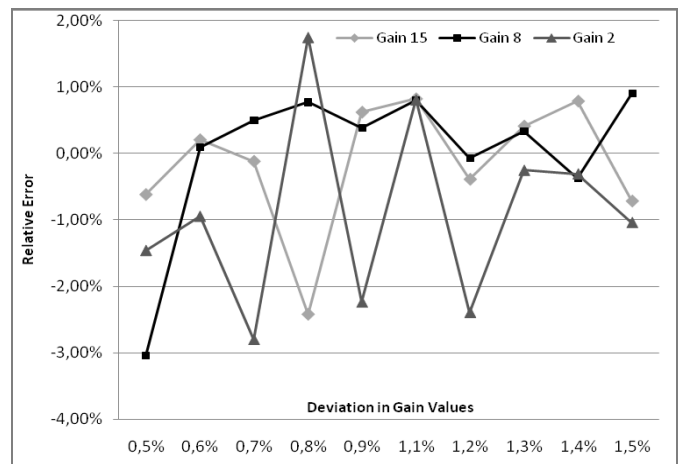


Fig. 9. Errors in the target gain under deviation fault condition in PGA3

Figs. 8 to 10 show the deviation fault simulation results for the fault tolerant system. The figures depict the relative errors in the target gains versus the deviation value in the gain. From the simulation results, it is observed that the SA is able of reaching the target gain with errors for all the gains in the range [-3,045%, 3,113%].

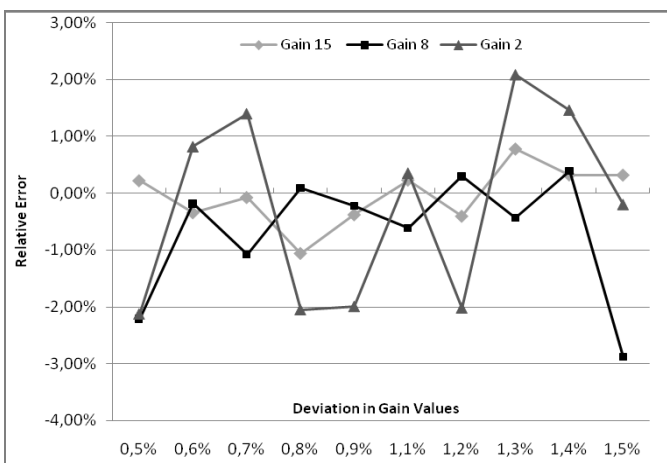


Fig. 8. Errors in the target gain under deviation fault condition in PGA1

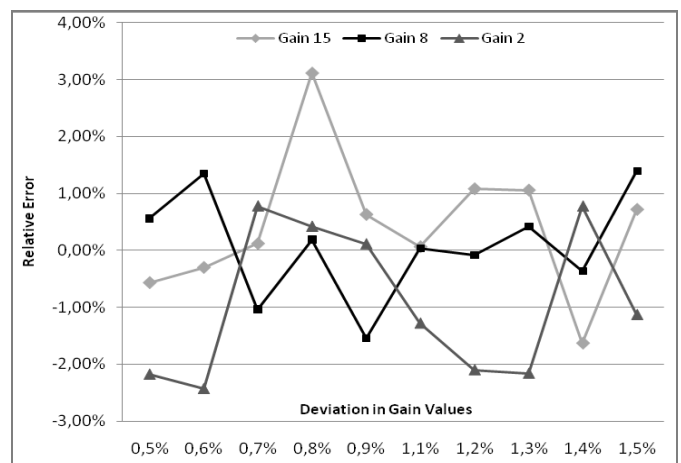


Fig. 10. Errors in the target gain under deviation fault condition in PGA4

Table III summarizes the effects of deviation faults in the four PGAs. Comparing the normal (Table I) and deviation fault conditions (Table III), the faulty system presents as a worst case an increase of 0,056% in the error range for gain 15,

despite the presence of relatively high deviation faults. For target gains 2 and 8 the error range is lower than the error range in operation normal. For target gain 2, the median is lower than the median in normal operation. For target gains 8 and 15 the median is higher. This indicates a slight change in the error distribution between the normal and faulty operation.

TABLE III. GAIN ERROR CHARACTERIZATION UNDER DEVIATION FAULT CONDITION

Target gain	Median	Minimum error	Maximum error	Error range
15	0,112%	-2,419%	3,113%	5,532%
8	-0,019%	-3,045%	1,395%	4,441%
2	-0,351%	-2,799%	2,085%	4,884%

### VI. COMPARISON WITH EXHAUSTIVE SEARCH METHOD

For a better characterization of the efficiency of SA algorithm, we compare the results with those obtained using Exhaustive Search Method (ESM). This method consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem statement [16].

We perform the comparison for fault-free and faulty operations described in Section V, using as parameters the number of objective function evaluation (Table IV) and runtime (Table V). SA and ESM are both implemented in Matlab. For the comparison, we use for SA the median of its runtime in the worst case condition (target gain 15) and the maximum number of function evaluation. These values are chosen because the stochastic nature of SA. Table IV shows that the number of objective function evaluations made by SA is considerably lower than ESM in all the operation conditions. By other way, the SA runtime is lower than the ESM one. These results suggest that the use of SA is preferable to ESM.

With the aim of extending the comparison to other amplifier configurations, we perform new evaluations using three and five amplifiers in the amplifying chain. For a three-amplifier chain, we found that the runtime of the ESM is lower than the runtime of SA, even if the number of ESM evaluations of the objective function is higher. However, for a five-amplifier chain, the SA is almost 8.340 times faster and performs about 120.000 times less objective function evaluations than ESM. These experiments suggest the convenience of using SA in more complex amplifier configurations.

TABLE IV. EXHAUSTIVE SEARCH METHOD (ESM) VERSUS SA. COMPARISON OF NUMBER OF OBJECTIVE FUNCTION EVALUATIONS.

Method	Normal condition	Catastrophic fault condition	Deviation fault condition
SA	200	200	200
ESM	810.000	783.000	810.000

TABLE V. EXHAUSTIVE SEARCH METHOD (ESM) VERSUS SA. COMPARISON OF RUNTIMES (SEC)

Method	Normal condition	Catastrophic fault condition	Deviation fault condition
SA	0,116	0,145	0,149
ESM	0,247	0,211	0,247

### VII. CONCLUSIONS AND FUTURE WORK

We presented an amplifier system with fault tolerance characteristics achieved by reconfiguration of a commercial programmable system on chip, using a built-in self-test strategy that establishes the amplifier gains and start the process of reconfiguration when a fault is detected. The SA algorithm that finds the reconfiguration values for the system is robust for all faults addressed in our evaluation. The fault simulation results show that the system maintains the overall gain within specifications despite the presence of catastrophic and deviation faults. The comparison with an exhaustive search method shows that the SA presents better performance.

### REFERENCES

- [1] P. C. Haddow, M. Hartmann and A. Djupdal, "Addressing the metric challenge: evolved versus traditional fault tolerant circuits", Second NASA/ESA Conference on Adaptive Hardware and Systems, Edinburgh, 2007, pp. 431-438.
- [2] J. Hereford, "Fault-tolerant sensor systems using evolvable hardware", IEEE Trans. Instrum. Meas, vol. 55, 2006, pp. 846-853.
- [3] G. Greenwood, "On the practicality of using intrinsic reconfiguration as a fault recovery method in analog systems", IEEE Trans. Sys. Man & Cyber, 1999, pp. 87-97.
- [4] S.Mitra, W. Huang, N. Saxena, S. Yu, and E. McCluskey, "Reconfigurable architecture for autonomous self-repair", Design & Test of Computers, IEEE, vol. 21, 2004, pp. 228-240.
- [5] P. Garcia, K. Compton, M. Schulte, E. Blem, and W. Fu, "An overview of reconfigurable, Hardware in Embedded Systems", Hindawi Publishing Corporation EURASIP, Journal on Embedded Systems, Volume 2006, pp. 1-19.
- [6] J. Emmert, C. Stroud, and M. Abramovici, "Online fault tolerance for FPGA logic blocks", (VLSI) Systems, IEEE, vol. 15, 2007, pp. 216-226.
- [7] T. Higuchi, Y. Liu, and X. Yao, (Eds.), Evolvable Hardware, Springer, 2006.
- [8] R. Canhoam and A. M. Tyrrell, "Evolved fault tolerance in evolvable hardware", Congress on Evolutionary Computation, Hawaii, 2002, pp. 1267-1272.
- [9] M. Lovay, A. Arregui, J. Gonella, G. Peretti, E. Romero, and M. Lubaszewski, "Fault tolerant amplifier system using evolvable hardware", Proceedings of the Argentine School of Micro-Nanoelectronics, Technology and Applications, 2010, pp. 50-55.
- [10] M. Gendreau and J. Potvin (Editors), Handbook of Metaheuristics, Second edition, Springer, 2010.
- [11] P. Li, J. Lan, D. Li, Q. Liu, "A simulated annealing based technology mapping method for sequential circuits", IEEE, First International Conference on Future Information Networks, 2009.
- [12] H. Shakouri G, Kambiz Shojae, and M. Behnam T, Investigation on the choice of the initial temperature in the Simulated Annealing: A Mushy State SA for TSP, IEEE, 17th Mediterranean Conference on Control & Automation, Greece, 2009, pp. 1050-1059.
- [13] F. Rodríguez-Díaz, C. García-Martínez, and M. Lozano, A GA-based multiple simulated annealing, IEEE Congress of Evolutionary Computation, 2010, pp. 1-7.

- [14] PSoC® Programmable system-on-chip technical reference manual. United States: Cypress Semiconductor Corporation, 2008.
- [15] A. Doholi, P. Kane, and D. Van Ess, Dynamic Reconfiguration in a PSoC Device. International Conference on Field-Programmable Technology 2009, pp.361-363.
- [16] K. Price, R. Storn, J. Lampinen, Differential evolution: a practical approach to global optimization, Springer, 2005.