Digital Systems with Self-Healing by Hardware Reconfiguration

I. Ştefan, G. Moiş, Sz. Enyedi, L. Miclea
Technical University of Cluj-Napoca
{Iulia.Stefan, George.Mois, Szilard.Enyedi, Liviu.Miclea}@aut.utcluj.ro

***Abstract*-This paper proposes a new approach for distributed system testing and repairing using mobile hardware agents. This way, we obtain a networked reconfigurable system which does not need human intervention for maintenance and testing. The proposed architecture is flexible and re-programmable.**

I. Introduction

Our times see the continuation of an exponential development in computer science and microelectronics, constant growth of systems' integration level, miniaturization and the development of increasingly complex and larger scale integrated digital systems. These advances, as well as the development of electronics used in heterogeneous systems, require new testing and fault elimination methods.

For these systems, classic and local testing and repair does not produce results at overall system level, due to the very large number of elementary subsystems and their heterogeneous nature. On the other hand, a *distributed, decentralized* solution is much easier to design, implement, maintain and utilize [2].

The growth of microelectronics integration level has brought, besides benefits, also the multiplication of the transient and permanent faults, a fact that led to a decrease in reliability. Of course, there are many old and acknowledged solutions for counteracting this problem, but most of them require human intervention, or at least the intervention of an external factor.

II. Agents

An *agent* is a piece of software capable of independent existence within an environment provided for it, is able to communicate with other entities, to unaidedly accomplish the work assigned to it and also to travel between geographically separate locations in its environment. It is an independent, mobile program capable of functioning for task execution in a flexible manner and in continuously changing surroundings.

Any entity that perceives its environment using sensors and takes action modifying those surroundings using actuators can be considered an agent.

A seemingly complete definition was given by J. Ferber (1989): "A real or abstract entity that is able to act on itself and its environment; which has partial representation of its environment; which can, in a multi-agent universe, communicate with other agents; and whose behavior is a result of its observations, its knowledge, and its interactions with the other agents." [1].

Distributed systems, especially ones consisting of a large number of modules or those distributed over a wide geographical area, are easier to monitor, test and maintain with the help of such agents.

The agents travel from device to device, try to detect and repair errors, either by themselves or with the help of other agents or a central database. They can also gather "experience" through their work.

III. Reconfigurable System

This article focuses on the use of reconfigurable systems and tries to find a viable and cost efficient solution for the implementation of such systems. Hardware self-maintenance is a relatively new method and implies that the defective device or the device that is about to be out of order "takes notice" and recovers from this state. Usually, this can be realized through a surplus or a duplication of the present functions, in hardware, and through the use of local software or the use of decision electronics for testing and/or deactivating the defective part and reallocating the tasks to the spare part of the device.

There are many solutions for the decision logic and the replacement of the tasks. For example, if the functionality is implemented in an FPGA, in case of a fault, an unused part of the matrix can be reconfigured, taking over the functionality of the defective part [3,4]. For better results, one can use a method for detecting the faulty parts which is similar to the human immune system: the cells found to be defective are replaced with spare ones that take over their functions [5].

In our research, we tried to achieve this by reconfiguring an entire FPGA in case a critical part of the system is found to be defective.

IV. Experimental Setup

In this section we present the experimental setup of a simple reconfigurable system. The prototype we built is oriented towards a networked infrastructure and is essentially aimed at exploring a particular architectural pattern based on a centralized agent host (a personal computer) and two low-end peripheral nodes with reconfigurable capabilities. This organization is representative for a large class of distributed systems and could be used to model, for example, a distributed control application. We used two XUP V2P Xilinx FPGA boards which communicate with each other and with the computer through a wireless interface using Wi-Fi modules.

This research was funded by the Romanian Government through the research grant PN-II IDEI 93/1.10.2007.

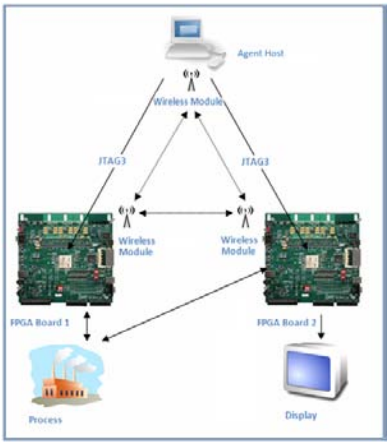

Figure 1. Experimental Setup

One of the boards is programmed to act as a microprocessor in charge of controlling a process. This part has a high importance in our experimental system, because if this microprocessor is defective, the entire system is unable to correctly execute its tasks.

The other FPGA board is programmed to act as a video board for displaying the parameters of the controlled process. This is the part of the system that we can give up in case we need a programmable spare.

The computer is the supervisor and, collaborating with the agents, verifies periodically if the two other components are working without any faults. In fact, the supervisor waits for a message from the mobile agent that “lets it know” whether the two other parts of the systems don’t work correctly. Also, it is continuously backing up the microprocessor’s state. The mobile hardware agent scheme was introduced in the project for pushing the concept of testing and repairing using mobile agents to the hardware level.

The program that tests the correct functionality of the two programmable parts of the system is a mobile hardware agent. Through a wrapper, it uploads its code to the microprocessor board and to the display controller board in order to check their behavior. The agent runs a series of test routines to see if the two parts produce the expected data output. If not, it sends a notice message to the supervisor which will send the required bitstream to reprogram the FPGAs.

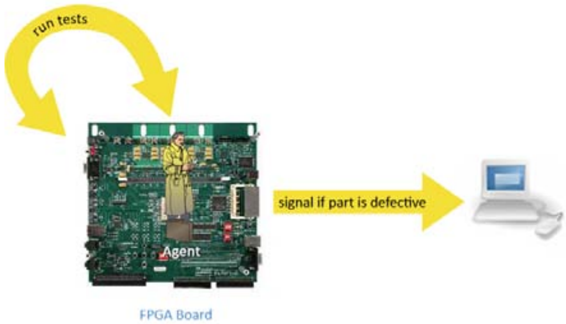

Figure 2. Hardware agent running the tests

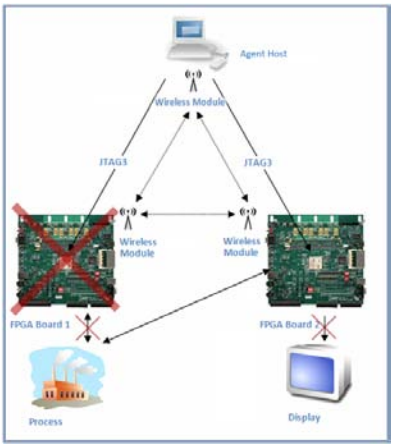

Figure 3. Reconfigured System Structure

If the display board is not working properly, the entire system can execute its tasks without any immediate intervention, but, if the microprocessor is damaged, the process is endangered. At this point, the entire system is reconfigured, reprogramming the video board to act as an identical backup microprocessor. This new microprocessor has the last valid values of the old one written in its registers, for controlling the process. The old microprocessor is no longer used and the one the supervisor has just programmed takes its place in the system.

The experimental setup proves that the concept can be implemented and can be used to solve problem situations without needing human intervention.

V. Conclusions

In this paper, we analyzed how the innovative aspects of reconfigurable systems technology and hardware agents can be exploited to implement efficient and new test and repair strategies.

Our experimental proof of concept setup uses a personal computer as supervisor and reconfiguration manager for the hardware agents and reconfigurable systems, but the use of another FPGA board for this job is planned. The possibility of programming the FPGA boards wirelessly is a solution that will be a great improvement of the design. Currently, we send the bitstream through the boards’ JTAG ports, this being a major drawback.

References

[1] J. Ferber 1989. *Des Objets aux Agents*, Doctoral Dissertation, University of Paris VI.

[2] Sz. Enyedi, L. Miclea, H. Vălean, I. Stoian, *On-line monitoring and testing of a power plant chain*, Proceedings of ICPS'07, Cluj-Napoca, Romania, July 9-11, 2007, ISBN 978-973-713-180-5, pp. 228-233.

[3] A. Benso, A. Cilardo, N. Mazzocca, L. Miclea, P. Prinetto, Enyedi. Sz., *Reconfigurable Systems Self-Healing using Mobile Hardware Agents*, Proceedings of International Test Conference 2005, Austin, TX, USA, November 8 - 10, 2005.

[4] S. Habermann, R. Kothe, H. T. Vierhaus, *Built-in Self Repair by Reconfiguration of FPGAs*, Proceedings of IEEE International On-Line Testing Symposium, 2006, ISBN 0-7695-2620-9/06.

[5] P. K. Lala, B. Kiran Kumar, *An Architecture for Self-Healing Digital Systems*, Proceedings of IEEE International On-Line Testing Workshop, 2002, ISBN 0-7695-1641-6/02.

[6] L. Miclea, Enyedi Sz., A. Benso, *Intelligent Agents and BIST/BISR - Working Together in Distributed Systems*, Proceedings of International Test Conference, Baltimore, USA, 8th–10th October, 2002, pp. 940-946.

[7] L. Miclea, Enyedi Sz., *Distributed Built-In Self-Test using Intelligent Agents*, IEEE European Test Workshop ETW'2002, Corfu, Greece, May 26th–May 29th, 2002.

[8] Dennis Sylvester, David Blaauw, Eric Karl, ElastIC: An Adaptive Self-Healing Architecture for Unpredictable Silicon, IEEE Design & Test of Computers, Nov-Dec 2006, ISSN 0740-7475/06, pp. 484-490.

[9] Philip Garcia, Katherine Compton, *A Reconfigurable Hardware Interface for a Modern Computing System*, Proceedings of International Symposium on Field-Programmable Custom Computing Machines, 2007, ISBN 0-7695-2940-2/07, pp. 73-81.

[10] Wermelinger, M., Koutsoukos, G., Lourenco, H., Avillez, R., Gouveia, J., Andrade, L., Fiadeiro, J.L, *Enhancing Dependability Through Flexible Adaptation to Changing Requirements*, IEEE Computer Society Press, 2003.

[11] Verkest, D., *Machine chameleon*, Spectrum Magazine, IEEE, 2003.

[12] ***FIPA SC00001, Abstract Architecture Specification, http://www.fipa.org/specs/fipa00001/SC00001L.html

[13] A. Benso, S. Chiusano, S. Di Carlo, *HD2BIST: a Hierarchical Framework for BIST Scheduling, Data patterns delivering and diagnosis in SoCs*, ITC International Test Conference, pp. 899-901, 10 - 2000.

[14] Monica Lobetti Bodoni, A. Benso, S. Chiusano, G. di Natale, P. Prinetto, *An Effective Distributed BIST Architecture for RAMs* , Informal Digest of IEEE European Test Workshop ETW 2000, pp. 201-206

[15] F.Corno, M.Sonza Reorda, G.Squillero, M.Violante, *On the Test of Microprocessor IP Cores*, in Proceedings of the Design Automation & Test in Europe 2001, pp.209-213.