Artificial Immune System Implementation upon Embryonic Machine for Hardware Fault-tolerant Industrial Control Applications

Géza HUSI[1] Csaba SZÁSZ[2]

Virgil CHINDRIŞ[3]

GJCST Classification
B.8.1, J.7, I.2.9, I.5.1

***Abstract*- Living organisms demonstrate through millions of years evolution remarkably fault-tolerance, robustness and self-healing abilities. Taking inspiration from biological immune systems and embryonic processes which acquire some of these fault-tolerant properties, the paper presents the implementation of an embryonic machine with FPGA-based multi cellular architecture, which is able to imitate cells or artificial organism operation mode, with similar robustness and fault-tolerance properties like their biological equivalents from nature. This VLSI hardware structure was upgraded through specially developed algorithms, provided with strongly network communication capabilities and self-healing behaviors. Several casual faults were considered also through the test operations. Detection and localization of these hardware faults were achieved through special reconfiguration operations of active and spare artificial cells inside the embryonic array. Laboratory experiments prove high validity of the considered background theoretical approaches, the developed hardware immune system express remarkable surviving and self-healing capabilities.**

Keyword-artificial cell, embryonic machine, hardware immune system, POE architecture, FPGA circuit

I. Introduction

As the new generation programmable hardware systems becomes more complex, it becomes increasingly difficult to avoid manufacturing errors or occasional internal faults, and to determine the validity of the system. No electronic components - like diodes or transistors - will function for ever, and these faults can manifest themselves as internal errors, or can ultimately cause a system to fail. Therefore, the ability of a system to function in the presence of faults, and more, to become fault tolerant, is a highly increasing research area for engineers from informatics and microelectronic sciences.

In case of VLSI hardware systems, the traditional fault detection methodologies seem to be very inefficient and also expensive. The alternative for all these problems was inspired from living biological organisms, which are provided with remarkable surviving and fault-tolerance properties. Enhanced with remarkable abilities during a long evolution process, they are under continuous attack from other living entities, can survive infectious pathogens, injury, or several diseases. By adapting these mechanisms and capabilities from nature, scientific approaches have helped researchers understand related phenomena and associated with principles to engine complex novel digital systems and improve their capability. As a result of these efforts, bio-inspired techniques are now frequently used in VLSI digital systems design and development.

In last decade several projects were started on the POE model theoretical background, which means the creation of a hardware system that exhibits learning, evolutionary diversity, and multi-cellular organization [1], [2], [3], [4]. Phylogeny (P) is basically concerned in species genetically evolutionary. In engineering sciences, this corresponds to the genetic algorithms and evolvable hardware. Ontogeny (O) involves multi-cellular organization, cellular division and differentiation from the mother to the daughter cell (each cell owns a copy of the original genome). Finally epigenesist (E) is concerned with learning and adaptation processes (for example: nervous system, immune system) [5], [6], [7].

The embryonic systems were born as a result of the above mentioned research effort, on support of the cellular embryology basic terminology. Usually an embryonic system is considered to be a homogenous array of logic units (called artificial cells), on which is built a hardware multilayered artificial immune system, and which can then accommodate the faults [8]. Such systems, similar to their biological equivalents will posses cellular architecture properties: multi-cellular organization, cellular division, cellular differentiation, and they will virtually mimic every aspect of a living organism to achieve the above mentioned POE features [9], [10].

II. Biological Organisms as Model for Artificial Immune Systems Hardware Implementation

As it is known, the immune system found in higher evolutional level biological organisms is a distributed and multilayered system that is robust and able to identify infectious pathogens, injury, diseases, or other harmful effects. Therefore, their properties and abilities - like self-healing or surviving - would be more advantageous in many applications were often are imposed robustness and also high security operation requirements. The basic goal of these research efforts is to take inspiration from biological organism's immune system and embryonic processes to acquire these fault tolerant properties in hardware circuits.

About-[1]H. G. is with the Department of Electrical Engineering and Mechatronics, University of Debrecen, Ótemető u. 2-4, 4028 Debrecen, HU (phone: 36-52- 415-455, e-mail: husigeza@mk.unideb.hu).

For this reason, the artificial immune systems have been applied to many different application areas, such as: hardware fault tolerance, industrial process monitoring, fault tolerant software, pattern recognition, electrical drives control, neural networks implementation, optimization and industrial control processes [9], [10], [11].

The artificial immune system presented in this paper is modeled on a POE-type embryonic structure developed in analogy with the evolutionary processes of biological systems. In accordance with this model, these embryonic systems derive from the multi-cellular structure of complex living organisms with strong hierarchical organization from molecular to population levels, as shown in figure 1.

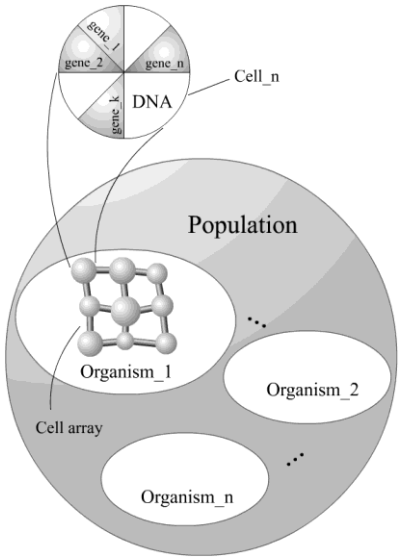

Fig. 1. Living organism's evolution process as model for POE-type artificial immune hardware systems development

All multi-cellular organisms start their life as a single cell, which divides then repeatedly to generate numerous identical copies of itself. Each cell contains all the information necessary to create the entire entity – the genotype, or named DNA, as it is expressed in figure 2.

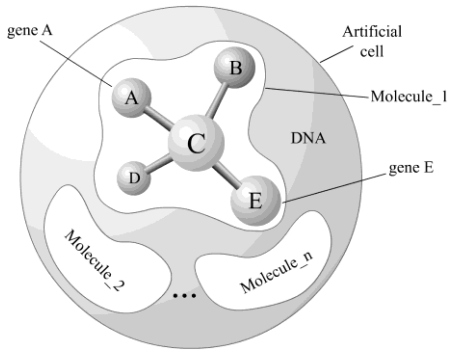

Fig. 2. A single cell structure from a multi-cellular organism

As the number of cells grows, cell differentiation takes place, when some of them start to change to provide different or specialized functionality. In this case the appropriate gene (or genes) is selected based upon the cell's position inside the cell network as well as other factors.

III. FPGA CIRCUITS-BASED ARTIFICIAL IMMUNE SYSTEM MODEL DEVELOPMENT

Considering as starting point and background the theoretical approaches from the previous paragraph, a new model of hardware multilayered artificial immune system based on embryonic array structure is proposed to develop. In this idea, is considered a homogeneous array of programmable logic units (called artificial cells) that use their location within the network to extract appropriate configuration data. Each artificial cell contains all the configuration details of all cells and hence can perform any cell's function as required. Due to avoid complex structures presentation, let's consider a model of entities composed each from 9 cells, organized in macro-groups of cell networks, and named shortly clusters. Figure 3 shows the structure of this conception, limiting the cell network area at first to case of just one cluster.

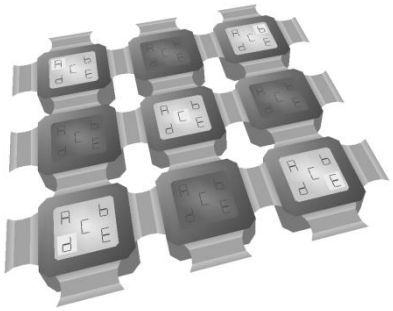

Fig.3. Artificial cells organized in a cluster structure

For more simplicity, in the model, the cell DNA is designed only with 5 genes (*A, B, C, D,* and *E*), showing active at the same time just one of them (highlighted in the figure). The 4 cells without active genes (dark in the figure) are considered spare cells in the network. The implemented genes are generically labeled in the model with *A, B, C, D,* and *E,* but they can represent in fact a wide range of control algorithms and programs (industrial process control, electrical motors control, etc.) defined by the software implementation.

Previous estimations for hardware implementation seem to evince an increased processing power and network communication abilities for the model. In this context, the FPGAs (Field Programmable Gate Arrays) because of their specific internal architecture circuits seem to be the most appropriate elements for such types of implementation. Thus, it means that each cell will be considered an autonomous FPGA array, with special functions inside the organism, defined through an instruction set (program), and called the cell's gene. Each cell has a copy of all genes from the organism (operative genome), and depending on the

cell's position inside the organism, only one cell has an operative gene (the cell's differentiation properties).

By extending the foregoing ideas, it was easy to conclude that the next stage of development should be expanding the previous model with several cell clusters. Thus, it was developed the artificial organism model shown in figure 4.

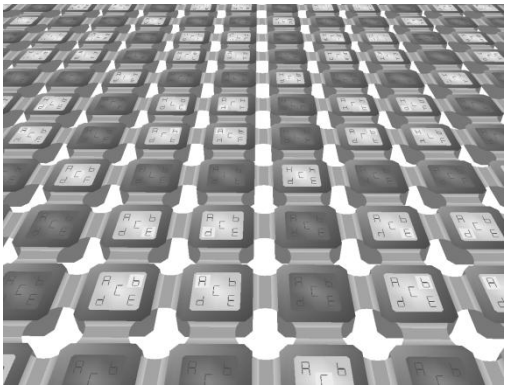

Fig. 4. Artificial organism model

In this structure, there are more interconnected groups (each of 9 cells), which can make up a square matrix architecture or lop-sided with a different number of rows and columns. This setting does not affect in any way the theoretical point of view or model functionality, all artificial cells are going to work together inside a homogenous embryonic array.

Figure 5 gives an example of network structure for the artificial immune system (the cells are numbered after their row and column position).

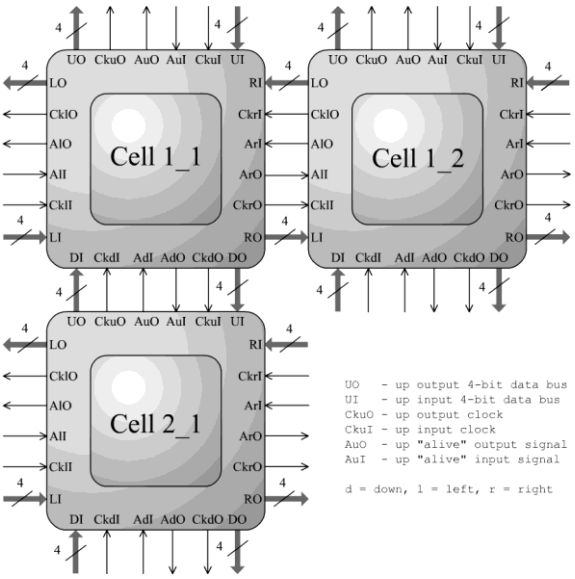

Fig. 5. Network structure of the embryonic system

There are 4-bit data buses on each lattice for source-cluster identification *X* and *Y* coordinates code, destination-cluster identification *X* and *Y* coordinates code, destination- and source-cell identification code, and for the implemented genes code (*A, B, C, D,* and *E*). Cell operation state (active, or faulty) is indicated through *Alive*-type signals (*Alive=1* active, *Alive=0* means faulty cell, e.g.: *AuI=1*), and all communication data are synchronized by *Clk*-type signals.

In figure 6 it is shown a typical fault detection process, where the neighbor cells that discover damage (*Alive=0* on each lattice of one cell) pass through a sequential process this information to entire network. In this way, all artificial cells become aware almost instantaneously of this error or damage.

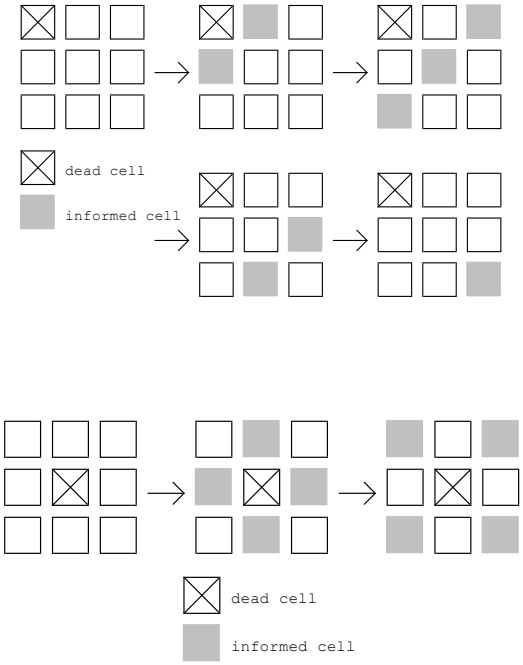

Fig. 6. Detection and information about occurred fault

As it is known, a biological immune system never tries to provide a fault free functionality, typically killing of infected cells. The biological entity can easily accommodate this due to the huge quantity of redundancy inside the organism. In contrast with the above, the hardware immune systems require some inherent fault tolerance, where as many faults inside the system, there are spare cells which can handle the occurred physical errors and damages. These theoretical observations are imitated briefly below in figure 7.

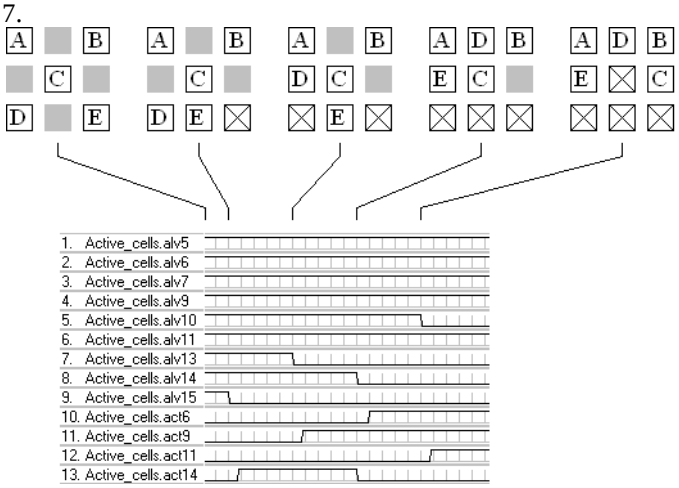

Fig. 7. Immunity of embryonic hardware system

The first cluster from left-upper side of figure, contains 4 spare cells (the cells without active gene) that are not utilized until a fault occurs. In case of fault, for example the cell which shows gene *E*, all buses on the 4 lattice of the cell are released and thus the neighboring spare cell takes over the functionality of faulty cell. It should be reiterated that no configuration data has to be recalculated or moved; just the change in coordinate is all that is required for the cell to reconfigure themselves. This elimination process can be repeated more and more, until are not enough spare cells inside the cluster. But in any case it can be observed that the embryonic array keeps during this process its immunity, showing active the same genotype (A, B, C, D, and E). The faulty cells replacing process with spare cells is presented also through the time diagram of Alive signals shown below the considered clusters. The above presented methodology can increase considerably the fault tolerance and self-healing properties of the embryonic array, and is well suitable for reconfigurable hardware implementation.

IV. Artificial Immune System Simulation

As it is known, an immune system is composed of a huge number of cells which protects an organism from infection and pathogens. From this point of view biological organisms are observed to have an amazing stability. The cost to have such properties is huge; a reliable embryonic system usually has redundant information about itself. These observations are briefly illustrated also in figure 8, where an artificial organism (composed by 6 cell clusters) with multiple faults is considered for computer-aided simulation.

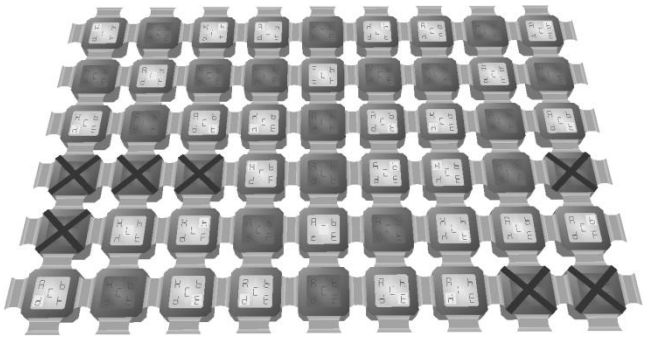

Fig. 8. Multiple faults in the artificial organism

Several alleatory faults of active or spare cells are considered in this simulation example. Faulty of redundant cells (spare, or without active gene) do not mean in any manner a threat for the organism functionality. But the faulty of one active cell, for example with gene B in cluster 2_1 (left-down side of the organism) is a real problem. As it is observed from the figure, this out of order cell was replaced already by a redundant spare cell which shows the same active gene B (highligted in the picture). In the cluster 2_3 (right-down side) three cells are faulted one after the other: one spare cell, and two active with genes B and E. After these unwanted events, from the four spare cells of the cluster, two cells become instantaneuosly active showing the same gene B and E. The result is: the cluster 2_3 kept its immunity, showing active the same genotype (A, B, C, D, and E), and remains immune, with high fault-tolerance ability. If the clusters can maintain individually their high immunity (in strong relationship with the available number redundant or spare cells), it means that the whole organism structure is also protected against any occasional faults. These events are presented also by a signals time-diagram (the *Alive* signals of each cells in the mentioned clusters), as given in figure 9. A huge number of similar computer-aided simulations is possible to investigate using the artificial organism model presented already in figure 4. In each case, the result was the same robustness and fault-tolerance, proving the viability of the developed artificial immune system model presented in the previous paragraph.

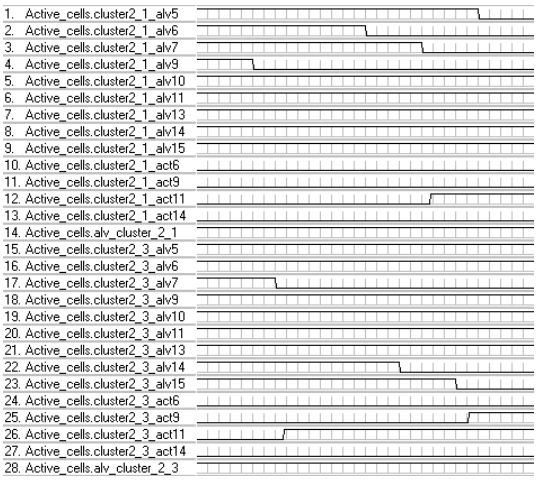

Fig. 9. Time-diagram of multiple faults in the cluster

It is important to mention, that there are important differences between natural biological systems and human-made systems. In the human-made systems, if a small piece does not operate properly, the system is not very reliable. In biological systems even an essential fault cannot stop operability of the full system because information about structure of the whole system and algorithms to self-healing are kept in a decentralized manner. The artificial hardware immune system model presented here follows also the above mentioned decentralization strategy.

V. Artificial Immune System Implementation On FPGA-Based Reconfigurable Hardware

The hardware implementation part of the project is motivated by two main goals: to perform a preliminary study of the ability of the presented artificial immune system model to imitate as close as possible the living organism's remarkable adaptation, surviving and fault-tolerance properties, and to develop fast implementation to explore the reconfigurable architecture hardware systems flexibilities. To use FPGAs is perfect for such an application, because the computation involved in the system is a massively parallel problem. Furthermore, an FPGA-based system can be easily

reconfigured to test a several theoretical approaches and hypotheses [12].

For implementation purpose it is defined at first the FPGA circuit-based artificial cell hardware structure (figure 10). This operation starts from consideration that each artificial cell from the presented models is built on using an autonomous FPGA array. Basic communication rules between cells inside the network are defined after the model described in paragraph III, and presented in figure 5.

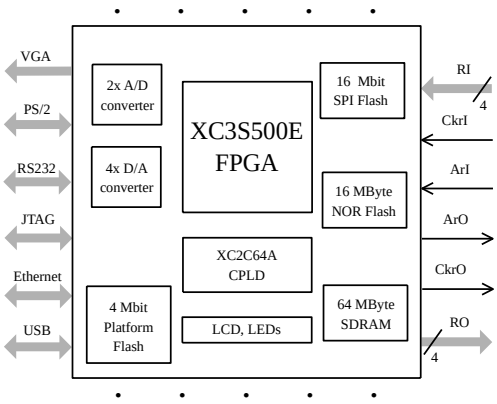

Fig. 10. Artificial cell block diagram

The proposed implementation, which is synthesized in an FPGA XC3S500E circuit, employs reconfigurable pipeline, 4 Mbit Platform Flash Memory, 16 Mbit SPI Flash, 16 MByte NOR Flash, 64 MByte DDR SDRAM Memory, 64-macrocell XC2C64A CoolRunner CPLD, SHA-1 1-wire serial EEPROM, 4-output SPI-based D/A Converter (DAC), 2-input SPI-based A/Digital Converter (ADC), and ChipScope™ SoftTouch debugging port [13]. To communicate with external devices is available for users also one PS/2 port, VGA display port, 10/100 Ethernet PHY, two RS-232 ports, and USB-based FPGA/CPLD download/debug interface. A photography overview of the Spartan-3E Starter Kit development board built on the above presented FPGA XC3S500E circuit is shown in figure 11 [12], [13].

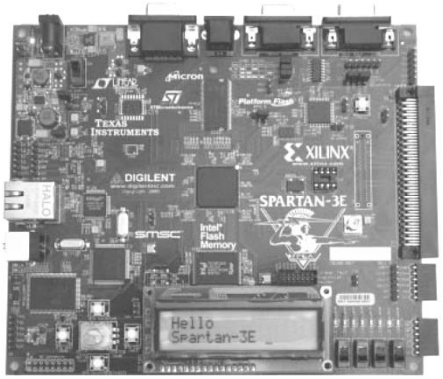

Fig. 11. Spartan-3E Starter KIT development board [13]

It's important to mention that the hardware system development strategy follows some special purpose requirements and implementation guidance. One of them is to avoid artificial cells hardware resources and memory utilization for initialization, configuration, or other auxiliary routines and algorithms implementation. The main goal is to maintain all their capability for complex software processing by algorithms that are able to imitate with high fidelity the biological organism's sophisticated adaptation, robustness, and immunity properties. This is the basic reason of choosing one stand-alone FPGAs for each artificial cell, with the above mentioned powerful hardware and software resources built in an FPGA XC3S500E circuit. The block diagram given in figure 12 shows this implementation strategy.

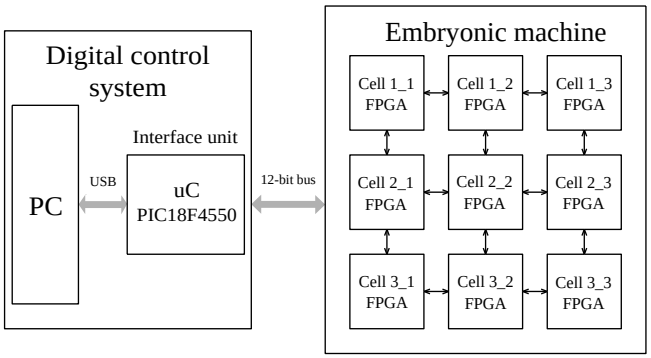

Fig. 12.The hardware system block diagram

The main goal was to design and construct a versatile framework system. There are two basic novel hardware structures: in the right side, it is depicted the developed embryonic machine (hardware immune system) with its array structure, and in the left the supervisor digital control system. No other functions or tasks are executed by the embryonic machine, just the regarded network communication abilities and specially developed fault-tolerance algorithms, in order to reproduce artificial hardware immune system behaviors. All auxiliary functions or drivers like interfacing, initialization, or data acquisition are processed by the supervisor digital control system, built on a personal computer and PIC18F4550-type microcontroller. A general view of the laboratory experimented test system is shown in figure 13.

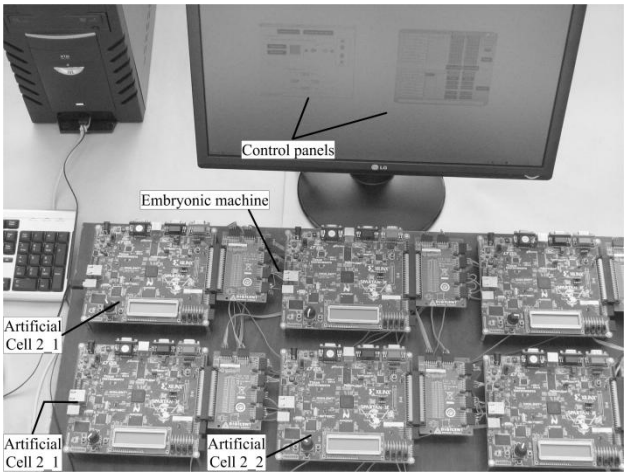

Fig. 13. The embryonic system

Data acquisition operations and initialization tasks are performed by the PIC18F4550 controller which operates like an intelligent interface unit between the personal computer and embryonic array. The main role of the computer consists of monitoring the network communication and data transfer rate inside the artificial cell network, through specially designed software control panels. Also each occurred fault in the embryonic machine is detected in real-time and represented in the panel's window.
The experimental results are focused upon monitoring the communication waveforms and data transfer protocol between the personal computer and embryonic machine via PIC18F4550 microcontroller. In this way there are checked continuously the right interoperation tasks between artificial cells, information about faulty cells and their replacing process, the cells coordinate detection inside the network, and the adequate genotype transmission for neighboring cells. The first example from figure 14 presents through the most relevant signal waveforms the above mentioned network operations time-diagram.

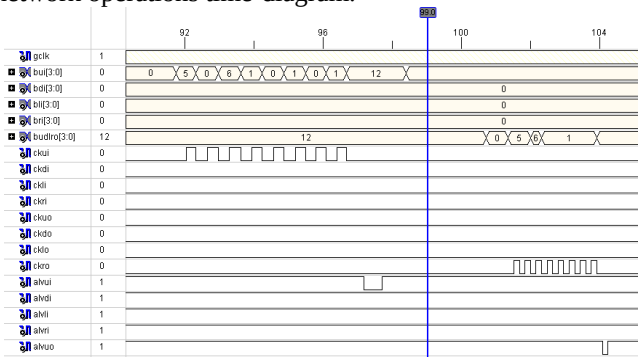

Fig. 14. Network communication waveforms (example_1)

In left-side of figure there are shown the 8 clocks of the *ckui* (*clk up in*, means PIC18F4550 controller connected to the upper side of the cell) signal which synchronizes the data transfer through the *bui(3:0)* data bus from controller to cell 1_1. This cell processes the captured instructions, and via the *alvui* signal marks the end of this operation. In right-side of figure cell 1_1 passes toward the captured instructions to cell 1_2, which in this example performs the position of destination cell.

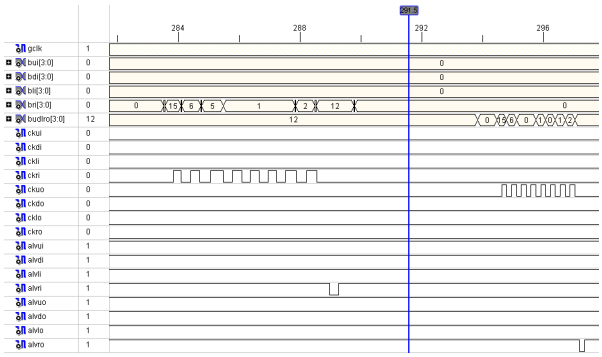

Fig. 15. Network communication waveforms (example_2)

In the second example from figure 15, cell 1_2 is already in role of the source cell, uploaded before with the data packages mentioned in example_1. After data processing over, cell 1_2 returns the result of computing via cell 1_1 to the interface unit built with the PIC18F4550 controller. In this case, the signals involved in communication are *ckri*, which synchronizes the data (signal *bri*) received by cell 1_1 from cell 1_2, and *ckuo* which synchronizes the data (signal *budlro*) from cell 1_1 to personal computer. The data transfer rate on the laboratory experimented embryonic machine was programmed during tests sufficiently low (3-5Hz) to ensure all network communication operations visual observation.

VI. Conclusions

The developed artificial hardware immune system may become helpful support for future developments in embryonic systems, in order to founding the theoretical basis, design models or development methods of this relatively new science domain named embryonic systems. The presented theoretical approaches and proposed models were carefully tested and implemented on a new generation of FPGA-based development system with a generous hardware resource with high-level programming and upgrading possibilities.
The importance of these research efforts consist basically of the influence of embryonic systems regarding the evolution of other engineering sciences from microelectronics and informatics. One of the most spectacularly application of embryonic systems are in high performance industrial control processes, where the system hardware security and maintenance is a major criteria. Not at last, through implementing the basic properties of living organisms on digital systems it becomes possible to realize high performance fault-tolerant and self-healing hardware architectures.

VII. References

1) A.Stauffer, D. Mange, G. Tempesti, "Bio-inspired computing machine with self-repair mechanism" Springer Berlin/Heidelberg, ISSN 0302-9743, vol. 3853, 2006.
2) A.Stauffer, D. Mange, J. Rossier, "Design of self-organized bio-inspired systems", 2nd NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 413-419, 2007.
3) Tempesti, D. Roggen, E. Sanchez, Y. Thoma, R. Canaham, A. Tyrrell, J.M. Moreno, "A POEtic Architecture for bio-inspired Systems" (Artificial life VIII), Sydney, Australia, December 2002.
4) R.O.Canham, A.M. Tyrrell, "A multilayered immune system for hardware fault tolerance within an embryonic array" Proc. of 1st International Conference on Artificial Immune Systems (ICARIS 2002), Canterbury, UK, September 2002.
5) A.Upequi, Y. Thoma, E. Sanchez, A. Perez-uribe, J.M. Malho, J. Manderas, "The perplexus bio-inspired reconfigurable circuit" 2nd NASA/ESA Conference on Adaptive Hardware and Systems (AHS), pp. 600-605, 2007.
6) R.O.Canham, A.M. Tyrrell, "A hardware artificial immune system and embryonic array for fault tolerant systems" Genetic programming and evolvable machines 4, pp. 359-382, 2003.
7) R.O.Canham, A.M. Tyrrell, "A learning multi-layered hardware artificial immune system implementation upon an embryonic array" ICES 2003, LNCS 2606, pp. 174-185, 2003.
8) X.Zhang, G. Dragffy, A.G. Pipe, Q.M. Zhin, "Artificial innate immune system: an instant defense layer of embryonic" ICARIS 2004, LNCS3239, pp. 302-315, 2004.
9) Zhang, G. Dragffy, A.G. Pipe, Q.M. Zhin, "Partil DNA supported artificial life in an embryonic array" Conference on Embedded Systems and Applications, Las Vegas, USA, 2004.
10) Y.Thoma, G. Tempesti, E. Sanchez, J.M. Arostegni, "An electronic tissue for bio-inspired cellular applications" Biosystems, vol. 76, no. 1-3, pp. 191-200, August-October 2004.
11) D.Mange, A. Stauffer, E. Petraglio, G. Tempesti, "Artificial cell division" Biosystems, vol. 76, no 1-3, pp. 157-167, Aug.-October 2004.
12) Xilinx Co., Spartan-3E Families User's Guide (www.xilinx.com) 2008.
13) Xilinx Spartan-3E Starter Kit Development Board, User Guide –www.xilinx.com, 2006.