Implementing Cellular Automata Bio-Inspired Algorithms on Field Programmable Gate Arrays

Nikolaos I. Dourvas, Michail-Antisthenis Tsompanas, Georgios Ch. Sirakoulis
Department of Electrical and Computer Engineering,
Democritus University of Thrace Xanthi, GR 67100, GREECE
Email: {ndourvas,mtsompan,gsirak}@ee.duth.gr

***Abstract*—Biological systems that have been subjected to vast time periods of evolution, have provided numerous times inspiration for problem solving techniques. These systems have evolved to survive in harsh environments, thus they provide more efficient solutions than the conventional methods. The inspiration of this study was based on the computing abilities that *Physarum Polycephalum* performs on graphically represented problems. The behavior of the plasmodium of *P. Polycephalum* was emulated using a model based on Cellular Automata (CAs) principles. The model is CA-based, thus, its implementation on hardware is trivial. The resultant digital circuit employ the inherent parallelism of CAs. As a result, the execution of the proposed algorithm is significantly accelerated compared with the software based model. The proposed system can be implemented in any real-time application solving graph represented problems.**

I. Introduction

Physarum polycephalum is a large amoeba-like cell consisting of a dendritic network of tube-like structures (pseudopodia). It changes its shape as it crawls over a plain agar gel, and if food is placed at two different points, it will extend pseudopodia that connect the two food sources (FSs). Nakagaki *et al.* [1] showed that this simple organism has the ability to find the minimum-length solution between two points in a labyrinth. This resulted in an intensive period of research on this organism that exposed a great range of its computational abilities to spatial representations of various graph problems [2], [3], [4], [5], [6]. On the other hand, the experiments on a living organism may last a lot of hours or maybe some days to provide data [7]. Therefore, it is necessary and meaningful for us to model its behavior precisely as fast as possible. Consider only the plasmodium stage of its life cycle, there is no single model that can describe exactly the behavior of Physarum. So far, there is a variety of modeling approaches which also are implemented by a variety of tools. The bibliography presents some purely spatial Cellular Automaton (CA) models [8], [9], [10], a mathematical representations of flux canalization [11], oscillatory behavior [12], [13], a two-variable Oregonator model of Belousov-Zhabotinsky (BZ) medium [14] and path-length [15].

CA is a very elegant computing model which dates back to John von Neumann [16] and Konrad Zuse [17]. They are models of physical systems, where space and time are discrete and interactions are local. They can capture the essential features of systems, where global behavior comes of the collective effect of simple components, which interact locally. Some significant examples are the simulation of physical systems [18], [19], crowd evacuation [20], computer architectures [21], [22], cryptography [23], computer networks [24] and Quantum CAs [25]. In addition, they can handle complex boundary and initial conditions, inhomogeneities and anisotropies [26] and work in parallel. These characteristics are very convenient for us to describe bio-inspired algorithms and particularly to simulate the dynamics of an organism such as *P. polycephalum* in a parallel environment such as an FPGA.

FPGA-based computation engines appear to be very promising for CA algorithms, since CA comprises a uniform structure composed of many finite state machines, thus matches the inherent design layout of FPGA hardware. There are various recent CA-based applications that have been implemented on FPGA's and achieve significant performance evaluation compared to the corresponding software implementations [27], [28], [29]. Finally, FPGAs can be completely dedicated to a particular function, in several cases they are more energy efficient than general-purpose CPUs. Therefore, although each implementation has its particular features, the merits of CA-based application on FPGA platforms along with low power consumption, compactness and portability, completely justify in many cases such an option [30], [31], [32], [33].

In this paper we revise the CA model that was proposed by Tsompanas and Sirakoulis in [8], and implemented on a FPGA platform, in order to reproduce the behavior of the plasmodium of *P. polycephalum*. The algorithm that was developed based on the CA model produced results that are in accordance with the ones produced by the real plasmodium. As we mentioned before, the experiments with the real plasmodium may last hours or even days. The implementation of modeling its behavior on software may last just some minutes or in best case seconds. A hardware implementation can accelerate this execution time to just some μs. So, in order to maximize the performance of this model, an automated hardware implementation in FPGA was developed. We implemented it by using [Very High Speed Integrated Circuit (VHSIC)] Hardware Description Language (VHDL). Because of the problems' inherent parallel nature and the CA's parallel nature, this hardware implementation showed a very fast computation performance.The aim of this paper is to have a more precise, fast and low-cost virtual laboratory that models the computation behavior of slime mould.

II. Physarum CA Model

In this section, the CA model designed to describe the behavior of *P. polycephalum* is discussed. We consider the biological experiment where the plasmodium was starved and then introduced into a specific place in the space. Moreover, a FS which produces chemo-attractants is placed in another place of the topology. In order to simulate this biological experiment, the area is divided into a matrix of squares with identical areas and each square of the surface is represented by a CA cell. The type of neighborhood that was used in this CA model is the Moore neighborhood which means that we use the north, south, east, west, north-east,north-west, south-east and south-west neighbors.The state of the (i, j) cell at time t, defined as $C^t_{i,j}$ is equal to:

$$C^t_{i,j} = \{Topology_{i,j}, Chem^t_{i,j}, Dir^t_{i,j}, Phys^t_{i,j}, Pseudo^t_{i,j}\} \tag{1}$$

- $Topology_{i,j}$ is a variable which indicates the type of area of the corresponding (i, j) cell. The possible values of this variable are 0,1,2,3 and indicate a free area, the spot of the initially placed FS, the spot of the initially placed plasmodium and the spot which represents a wall of the topology respectively.
- $Chem^t_{i,j}$ represents the concentration of chemo-attractants at time t in the area corresponding to the (i, j) cell. In order to calculate this variable for every cell, we make use of the diffusion equation of the chemo-attractants in each time step:

$$\begin{aligned} Chem^{t+1}_{i,j} = \{&Chem^t_{i,j} + \\ &f1[(Chem^t_{i-1,j} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i+1,j} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i,j-1} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i,j+1} - f3 \times Chem^t_{i,j})] \\ &+ f2[(Chem^t_{i-1,j-1} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i+1,j-1} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i-1,j+1} - f3 \times Chem^t_{i,j}) \\ &+ (Chem^t_{i+1,j+1} - f3 \times Chem^t_{i,j})]\} \end{aligned} \tag{2}$$

  The variables $Chem^t_{i-1,j}$, $Chem^t_{i+1,j}$, $Chem^t_{i,j-1}$, $Chem^t_{i,j+1}$ represent the concentration of the chemo-attractants of the north,south, west and east neighbor of the central cell (i, j), respectively. The variables $f1$, $f2$, $f3$ are the parameters of the diffusion equation (2) which are chosen after many simulations and have the values 0.05, 0 and 1, respectively.
- $Dir^t_{i,j}$ is a variable that indicates the direction of the attraction of the plasmodium by the chemicals produced by the FS. For example, if the area around a corresponding cell has no chemo-attractants, the foraging strategy of the plasmodium is uniform and, thus, these parameters are equal to zero. If there is higher concentration of chemo-attractants in the cell at direction x from the one in direction y, then the parameter corresponding to direction x is positive and the parameter corresponding in the direction y is negative. This happens, in order to more accurately simulate the non-uniform foraging behavior of the plasmodium.
- $Phys^t_{i,j}$ indicates the volume of the cytoplasmic material of the plasmodium in the corresponding (i, j) cell. In order to calculate this variable for every cell, we make use of the diffusion equation of the plasmodium in each time step is given by:

$$\begin{aligned} Phys^{t+1}_{i,j} = &Phys^t_{i,j} + \\ &o1\{[(1 + N^t_{i,j})Phys^t_{i-1,j} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + S^t_{i,j})Phys^t_{i+1,j} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + W^t_{i,j})Phys^t_{i,j-1} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + E^t_{i,j})Phys^t_{i,j+1} - o3 \times Phys^t_{i,j}]\} \\ &+ o2\{[(1 + NW^t_{i,j})Phys^t_{i-1,j-1} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + SW^t_{i,j})Phys^t_{i+1,j-1} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + NE^t_{i,j})Phys^t_{i-1,j+1} - o3 \times Phys^t_{i,j}] \\ &+ [(1 + SE^t_{i,j})Phys^t_{i+1,j+1} - o3 \times Phys^t_{i,j}]\} \end{aligned} \tag{3}$$

  where $N^t_{i,j}$, $S^t_{i,j}$, $W^t_{i,j}$, $E^t_{i,j}$, $NW^t_{i,j}$, $SW^t_{i,j}$, $NE^t_{i,j}$, $SE^t_{i,j}$ correspond to north, south, west, east, north-west, south-west, north-east, south-east directions. The variables $o1$, $o2$, $o3$ are the parameters of the diffusion equation (3) which are chosen once again after many simulations and have the values 0.05, 0 and 1, respectively.
- Finally, $Pseudo^t_{i,j}$ is a variable which can take values [0,1] and illustrates if the (i, j) cell is included in the final path of tubular network that is formed inside the plasmodium's body. This tubular network forms the shortest path between the FS and the cell from where the plasmodium started to expand and it is our final solution.

III. Hardware implementation

Current FPGAs include logic density equivalent to millions of gates per chip and can implement very complex computations. As mentioned above for hardware point of view, CA consist of a uniform n-dimensional structure, composed of many identical synchronous cells where both memory and computation are involved, thus matching the inherent design layout of FPGA Hardware (HW). As a result memory and processing unit are closely related both in CA cells and FPGA configurable logic blocks (CLBs). The structure of a cell consists of a combinational part connected with one or more memory elements in a feedback loop shape while the state of the memory elements is also defined by the inputs and the present state of these elements. For this paper the design produced by using VHDL code has been analyzed and synthesized by Quartus II (32-bit version 12.1 build) FPGA design software of ALTERA Corporation.

Each CA cell is implemented by a hardware block called "PhysarumCell". Each "PhysarumCell" block is connected appropriately with its four neighbors (west, east, south and

north). It uses the inputs from the neighbors and the previous state of itself to produce results that simulate the movement of the plasmodium. An "PhysarumCell" block has 22 inputs and 7 outputs.

The input signals can be categorized in the following categories:

- The circuit signals which are applied globally on all cells simultaneously. These signals are clk, $hold$ and rst and they are signals of 1-bit each. Clk represents the clock of the circuit needed to synchronize all cells in order to communicate at the same time. Rst represents the reset of the circuit. Finally $hold$ is enabled in one specific time step when the procedure of the tubular network formation is initialized. This signal is triggered manually.
- Signals $o1$ and $f1$ have 7-bit width and represent the parameter for the diffusion equation of the cytoplasm of the plasmodium and the chemo-attractants respectively. Moreover, the state signals which show the type of each cell, are included in this group. These are 2-bit signals named $topology$. Furthermore, signals $topologyNorth$, $topologySouth$, $topologyEast$ and $topologyWest$ indicate the type of the north, south, east and west neighbor of the central cell respectivelly.
- The last group is constituted by signals that are received by the cells who are adjacent to the central cell. Firstly, the incoming signals that show which of the neighbors are already part of the tubular network. Those are 1-bit signals, namely $enablePseudoFromEast$, $enablePseudoFromWest$, $enablePseudoFromNorth$ and $enablePseudoFromSouth$ that are enabled when the east, west, north and south neighbor are part of the tubular network, respectively. Moreover, there are signals that indicate the volume of cytoplasm of the plasmodium in each of the four neighbors. These are 19-bit signals, namely $physNorth$, $physSouth$, $physEast$ and $physWest$ that represent the volume of cytoplasm in the are represented by the north, south, east and west neighboring cell, respectively. Finally, there are signals that illustrate the concentration of chemo-attractants in each neighbor. These are, also, 19-bit signals, namely $chemNorth$, $chemSouth$, $chemEast$ and $chemWest$ that represent the concentration of chemo-attractants in the north, south, east and west neighbor, respectivelly.

The output signals are the following. The signals $physResult$ and $chemResult$ are 18-bit signals and represent the result of the diffusion equation of the cytoplasm of the plasmodium and the chemo-attractants, respectively. These two signals indicating the values for the central cell, are routed to the four neighbors of that cell. Moreover, the $pseudoResult$ signal is illustrating when the central cell is a part of the tubular network of the plasmodium. This signal is a 1-bit signal and is used to carry the results of tube forming from each cell of the grid to the outside world. Furthermore, the procedure of tube forming is triggered by a central cell to one of its neighbors, namely the one with the higher concentration of cytoplasm of the plasmodium. That is achieved by the 1-bit signals named $enablePseudoToEast$, $enablePseudoToWest$, $enablePseudoToNorth$ and $enablePseudoToSouth$. Each of these signals is driven to the appropriate neighbor that has the higher concentration of cytoplasm of the plasmodium and is available of participating to the tubular network of the plasmodium. The basic CA cell implemented in FPGA is presented in Fig.1 .

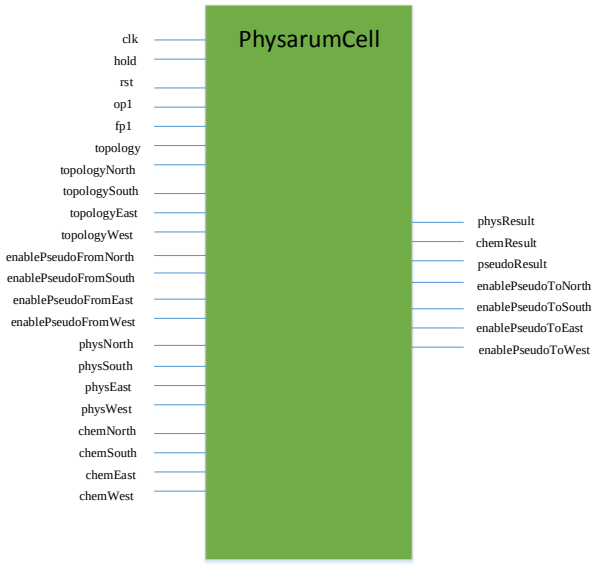

Fig. 1. Physarum CA Cell in FPGA.

TABLE I
FPGA HARDWARE IMPLEMENTATION DETAILS FOR ONE CA CELL.

Quartus II 32-bit Version	12.1 Build 243 01/31/2013 SP 1
Total logic elements	1,739
Total registers	45
Total pins	226

After creating the lattice, the user has to provide only the topology of the experimental area by giving values to the 2-bit signals $topology$ for each individual cell, namely the location of the FS and the location of the initial introduction of the plasmodium to the experimental area and the parameters for the diffusion equations.

The number of logic elements, registers and pins of the CA cell are presented in Table I. Moreover, to illustrate the area needed for a fully interconnected system of a CA grid implementing the proposed bio-inspired model, the results of synthesizing a 10×10, a 15×15 and a 20×20 grid are illustrated in Table II. The circuits are synthesized on several target devices and the results on the Stratix V 5SGXBB are presented here.

So, our hardware implementation seems to use less logic

TABLE II
FPGA HARDWARE IMPLEMENTATION FOR DIFFERENT TOPOLOGY SIZES.

	10×10	15×15	20×20
Total logic elements	161,162	370,447	666,060
Total registers	8,360	18,840	33,520
Total pins	317	692	1217

elements and registers than the previous work. Apparently, we use less physical space to synthesize our IP core. In Table II, it is shown that for every 200 CA cells there is an increment of 300,000 logic elements on average.

IV. CONCLUSIONS

In this paper, the hardware implementation of a bio-inspired CA based model is presented. An IP core that uses smaller physical space, as it uses lower amount of logic elements and registers than the previously proposed versions of the CA model was implemented. In addition, another difference is the verification by the IP core of the biological experiment where the plasmodium is placed on one place of the topology and the FS on the other place of significance. The proposed implementation results in a more clear path connecting the two points of interest in the topology with the shortest path between them. This model can, also, be used in order to solve more complex topologies with multiple paths. Consequently, this improvement shows that a low-cost virtual laboratory with more precise results and a better implementation of the bio-inspired algorithm is provided. Thus, the proposed model can be used more effectively and provide results faster on other engineering applications. Finally, as an aspect of future work, using more environmental parameters that affect the foraging strategy of the plasmodium, will enhance the robustness of simulating the biological experiment. Some examples are the humidity and the terrain morphology of the experimental area and the existence of chemicals that repel the plasmodium.

REFERENCES

[1] T. Nakagaki, H. Yamada and Á. Tóth, Intelligence: Maze-solving by an amoeboid organism, *Nature*, **407(6803)** (2000) 470–470.

[2] A. Adamatzky, Growing spanning trees in plasmodium machines *Kybernetes* **37** (2008) 258-264.

[3] T. Nakagaki, H. Yamada and M. Hara, Smart network solutions in an amoeboid organism, *Biophysical chemistry* **107(1)** (2004) 1–5.

[4] A. Adamatzky, (Ed.), *Physarum machines: computers from slime mould* (Vol. 74) (2010) World Scientific.

[5] A. Adamatzky and J. Jones, Road planning with slime mould: if Physarum built motorways it would route M6/M74 through Newcastle, *International Journal of Bifurcation and Chaos* **20(10)** (2010) 3065–3084.

[6] A. Adamatzky, G. J. Martnez, S. V. Chapa-Vergara, R. Asomoza-Palacio and C. R. Stephens, Approximating Mexican highways with slime mould, *Natural Computing* **10(3)** (2011) 1195–1214.

[7] A. Adamatzky, *Bio-Evaluation of World Transport Networks* (2012) World Scientific

[8] M.-A. I. Tsompanas and G. Ch. Sirakoulis, Modeling and hardware implementation of an amoeba-like cellular automaton, *Bioinspiration & Biomimetics* **7 (3)** (2012) 036013.

[9] Y. P. Gunji, T. Shirakawa, T. Niizato and T. Haruna, Minimal model of a cell connecting amoebic motion and adaptive transport networks, *Journal of theoretical biology* **253(4)** (2008) 659–667.

[10] Y. P. Gunji, T. Shirakawa, T. Niizato, M. Yamachiyo and I. Tani, An adaptive and robust biological network based on the vacant-particle transportation model, *Journal of theoretical biology* **272(1)** (2011) 187–200.

[11] A. Tero, R. Kobayashi and T. Nakagaki, Physarum solver: A biologically inspired method of road-network navigation. *Physica A* **363(1)** (2006) 115–119.

[12] R. Kobayashi, A. Tero and T. Nakagaki, Mathematical model for rhythmic protoplasmic movement in the true slime mold, *J. Math. Biol.* **53** (2006) 273-86.

[13] T. Nakagaki, H. Yamada and M. Ito, Reaction diffusion advection model for pattern formation of rhythmic contraction in a giant amoeboid cell of the Physarum plasmodium *J. Theor. Biol.* **197** (1999) 497-506.

[14] A. Adamatzky,If BZ medium did spanning trees these would be the same trees as Physarum built, *Phys. Lett. A* **373** (2009) 952-956.

[15] J. Jones, Approximating the behaviours of Physarum polycephalum for the construction and minimisation of synthetic transport networks, *In Unconventional computation* (Springer Berlin Heidelberg, 2009),191–208.

[16] J. von Neumann and A. W. Burks, *Theory of Self-reproducing Automata* (Urbana: University of Illinois Press, 1996)

[17] K. Zuse, Rechnender Raum, Vieweg, Braunschweig (1969); translated as "Calculating Space," *Tech. Transl. AZT-70-164-GEMIT* (1970) MIT Project MAC.

[18] I. Karafyllidis, Design of a dedicated parallel processor for the prediction of forest fire spreading using cellular automata and genetic algorithms., *Eng. Appl. Artif. Intell.* **17** (2004) 19-36.

[19] G. Ch. Sirakoulis, I. Karafyllidis and A. Thanailakis, A cellular automaton methodology for the simulation of integrated circuit fabrication processes., *Future Gener. Comput. Syst* **18** (2002) 639-657.

[20] I. G. Georgoudas, G. Ch. Sirakoulis, and I. Andreadis, An Anticipative Crowd Management System Preventing Clogging in Exits during Pedestrian Evacuation Processes, *IEEE Systems* **5(1)** (2011) 129–141.

[21] N. B. Wilding, A. S. Trew, K. A. Hawick and G. S. Pawley, Scientific modeling with massively parallel SIMD computers, *Proc. IEEE.* **79** (1991) 574-585.

[22] J. Jendrsczok, P. Ediger and R. Hoffmann, A scalable configurable architecture for the massively parallel GCA model, *Int. J. Parallel Emergent Distrib. Syst.* **24** (2009) 275-91.

[23] S. A. Chatzichristofis, D. A. Mitzias, G. Ch. Sirakoulis and Y. S. Boutalis, A novel cellular automata based technique for visual multimedia content encryption, *Optics Communications* **283(21)** (2010) 4250-4260.

[24] V. Mardiris, G. Ch. Sirakoulis, C. Mizas, I. Karafyllidis and A. Thanailakis, A CAD system for modeling and simulation of computer networks using cellular automata, *IEEE Trans. Syst. Man Cybern. C* **38(2)** (2008) 253-264.

[25] V. A. Mardiris and I. G Karafyllidis, Universal cellular automaton cell using quantum cellular automata, *Electron. Lett* **45** (2009) 607-609.

[26] G. Ch. Sirakoulis and S. Bandini, (Eds.) Cellular Automata - 10th International Conference on Cellular Automata for Research and Industry, ACRI 2012, *Proceedings, Lecture Notes in Computer Science*, **7495** (2012) Santorini Island, Greece, September 24–27, 2012, Springer.

[27] I. G. Georgoudas, G. Ch. Sirakoulis, E. M. Skordilis and I. T. Andreadis, On-chip earthquake simulation model using potentials, *Natural hazards* **50(3)** (2009) 519–537.

[28] G. Ch. Sirakoulis, A TCAD system for VLSI implementation of the CVD process using VHDL, *Integration, the VLSI Journal* **37(1)** (2004) 63–81.

[29] R. Porter, J. Frigo, A. Conti, N. Harvey, G. Kenyon, and M. Gokhale, A reconfigurable computing framework for multi-scale cellular image processing, *Microprocessors and Microsystems* **31(8)** (2007) 546–563.

[30] I. G. Georgoudas, P. Kyriakos, G. Ch. Sirakoulis and I. T. Andreadis, An FPGA implemented cellular automaton crowd evacuation model inspired by the electrostatic-induced potential fields, *Microprocessors and Microsystems* **34(7)** (2010) 285–300.

[31] G. Kalogeropoulos, G. Ch. Sirakoulis and I. Karafyllidis, Cellular automata on FPGA for real-time urban traffic signals control, *J. Supercomputing* **65(2)** (2013) 1–18.

[32] I. Vourkas and G. Ch. Sirakoulis, FPGA based Cellular Automata for Environmental Modeling, *in Proc, of the 2012 IEEE International Conference on Electronics, Circuits, and Systems* (2012) 308–313.

[33] P. Progias and G. Ch. Sirakoulis, An FPGA Processor for Modelling Wildfire Spread, *Mathematical and Computer Modeling* **57(5-6)** (2013) 1436–1452.