



Mycelium-Based ELM Digital Twin Implemented in FPGA

Ioannis K. Chatzipaschalis^{1,2(✉)}, Ioannis Tompris¹, Konstantinos Rallis¹,
Theodoros Panagiotis Chatzinikolaou¹, Iosif-Angelos Fyrigos¹,
Michail-Antisthenis Tsompanas³, Andrew Adamatzky³, Phil Ayres⁴,
Antonio Rubio², and Georgios Ch. Sirakoulis¹

¹ Democritus University of Thrace, DUTH University Campus, 67100 Xanthi, Greece
ichatzip@ee.duth.gr

² Universitat Politècnica de Catalunya, 08034 Barcelona, Spain

³ University of the West of England, Bristol, UK

⁴ Chair for Biohybrid Architecture, Institute of Architecture and Technology,
Royal Danish Academy, 1435 København K, Denmark

Abstract. Engineered Living Materials (ELMs) based on fungal mycelium offer a promising solution to the challenges posed by environmental disruption, resource scarcity, and rising material demands, while also exhibiting interesting computational capabilities. These materials are cost-effective, widely available, and environmentally beneficial due to their biodegradable nature. Using a Field-Programmable Gate Array (FPGA) to create a digital twin could make it much easier to predict growth patterns, find the best conditions for development, and look into possible uses for mycelial ELMs. Digital twins with advanced hardware integration are particularly effective, providing deep insights into the physical and mathematical aspects of these materials. This paper describes the use of Cellular Automata (CAs) and reaction-diffusion systems to model these processes due to their ability to handle the behavior of complex systems with scalability and parallelism. This approach has enabled the high-fidelity simulation of ELMs behaviors and the successful prototype implementation on a FPGA, making it a significant step towards practical applications of mycelial ELMs.

Keywords: Cellular Automata · Reaction-Diffusion · Mycelium-based ELMs · Reconfigurable Hardware · FPGA

1 Introduction

Engineered Living Materials (ELMs), particularly those based on fungal mycelium, represent a novel class of materials inspired by living organisms. These mycelium-based ELMs offer a sustainable solution to the escalating societal challenges posed by human-induced environmental disruption, resource scarcity, and increasing material demands. They stand out for their affordable manufacturing process, natural ubiquity, and significant environmental benefits like biodegradability [1, 11]. However, due to the complex biological mechanisms they encompass, their environmental sensitivity, the slow supply chain, and regulations, these materials present significant challenges for reproduction [15].

A digital twin would help scientists and engineers accurately predict growth patterns, find the best conditions for mycelial ELM development, and look at possible uses [21]. A digital twin [19] is a virtual representation that serves as the nearly identical digital equivalent of a real-world physical object, system, or process. This model is used for practical purposes including simulation, integration, testing, monitoring, and maintenance, effectively mirroring its real-world counterpart.

In this context, to effectively design a digital twin of the ELMs, it is crucial to select a modeling framework that can be easily ported into hardware while also taking full advantage of the available resources, as analyzed in Sect. 2. More specifically, Cellular Automata (CAs) have been strategically selected for this task due to their robust ability to simulate complex systems, as well as being scalable, modular, and showcasing massive parallelism [7,8]. CAs have been extensively utilized in the literature, from unconventional computing [17,20] to intelligent applications [5,14]. In this framework, CAs are specifically utilized to create a detailed grid that models the reaction-diffusion processes, providing a dynamic mapping of the morphological activities of the ELMs [4]. Leveraging this hardware-compatible modeling framework, along with their robust ability to simulate complex systems, has enabled the successful implementation of a mycelium-based ELM prototype on a Field-Programmable Gate Array (FPGA), elaborated in Sect. 3. The simulation results are presented in Sect. 4, while Sect. 5 concludes this work.

2 Hardware-Compatible ELM Model

2.1 Reaction-Diffusion Processes

The Reaction-Diffusion (RD) system employed consists of activator-inhibitor pairs that engage in dynamic interactions throughout the designated space, denoted as u and v , respectively [18]. These interactions significantly influence a third component, c , which characterizes the concentration of ELM in the specified region. In more detail, the activator u facilitates the growth of ELM by promoting its concentration c in regions where the activator is sufficiently present. On the other hand, the inhibitor v impedes the growth of ELM by constraining the concentration of the activator.

It is important to note that the reaction-diffusion model used, which includes the activator and inhibitor mentioned above, is not meant to be a representation of real chemical species. Instead, it is a mathematical framework used to show how the system works in a way that is generic enough and fits the mycelium evolution. Instead of directly modeling specific chemical entities, the involved abstract variables are used to mimic the patterns and behaviors that are seen in the ELM bio-inspired system.

This system has been utilized to model various processes and patterns by leveraging the complex dynamics of the activator-inhibitor pairs. In this study, the variable c denotes the concentration of the ELM. It signifies the degree of presence or absence of the material formed through the interplay between the

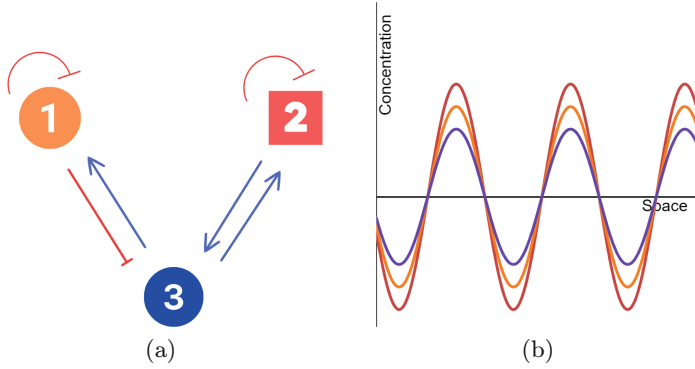


Fig. 1. (a) Activator (3), inhibitor (1) and main body (2) dynamics. Blue arrows signify promotion, while red flat-head links indicate suppression. (b) Concentration-Space graph for the concentrations of the activator (blue), the inhibitor (orange) and the main body (red). Adapted from [12]. (Color figure online)

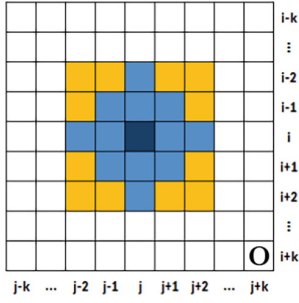
activator and the inhibitor. In this case, the activator u is strategically limited to a certain radius from the ELM's main body c . This radius is in the space shown as Ω_c , which shows the mycelium's plasma membrane. The inhibitor, on the other hand, is free to spread out in this space. The RD system employed, comprising of three components, describes a three-way interaction between the activator, the inhibitor, and the main body. More specifically, the activator diffuses within the main body and promotes its growth when its concentration reaches a set amount indicated by Eq. 4. Once the main body is generated, the activator's morphology is no longer affected directly by the inhibitor, which diffuses freely and demotes the activator's concentration. This interaction leads to an in-phase evolution of the system's components, as depicted in Fig. 1(b), resulting in a Mycelium-like pattern rather than a traditional Turing pattern generated by Turing instability. The presence of the main body component disturbs the phenomenon of Turing instability [23]. This results to a final pattern that depends on the combined effects of diffusion, growth rates, and the dynamics of the components. The dynamics of the activator-inhibitor system are illustrated in Fig. 1 and are mathematically expressed in Eqs. (1)–(4), which refer to a reaction-diffusion model for bio-inspired evolution of structures:

$$\frac{\partial u}{\partial t} = \nabla^2 u + n\zeta(\kappa u + u^2 - \lambda uv) \epsilon \Omega_c \quad (1)$$

$$\frac{\partial v}{\partial t} = d\nabla^2 v + \zeta(\mu u^3 - v) \epsilon \Omega \quad (2)$$

$$\frac{dc}{dt} = \zeta \nu c(a(u) - c)(c - 1) \epsilon \Omega \quad (3)$$

$$a(u) = \begin{cases} \xi, & \text{if } u \geq \text{threshold} \\ \xi - \theta(u - \text{threshold}), & \text{otherwise} \end{cases} \quad (4)$$



$$\boxed{\mathbf{O}} = \begin{cases} u(i+k, j+k), & u \in \Omega_c \\ v(i+k, j+k), & v \in \Omega \\ c(i+k, j+k), & c \in \Omega \end{cases}$$

Fig. 2. Variable spaces Ω (whole grid) and Ω_c (colored cells) in the CA grid employed. The cells of the evolution enable matrix N are categorized in orange cells that have reduced weights (equal to $1/10$), in light blue colored cells (equal to $7/20$), and in the central dark blue colored cell (with weight equal to $1/9$) to support circular growth. (Color figure online)

On the right side of Eq. (1) and Eq. (2), the initial term featuring the Laplacian operator delineates how the solutions u and v diffuse. The subsequent term, a polynomial of u and v , explicates the interactions between these solutions, referred to as the reaction term. Equation (3) describes the time derivative of the ELM's main body c , which correlates with the previous state of c and the function $a(u)$, indicating the dependence of the cell's growth on the activator. This dependency is detailed in Eq. (4). Additionally, the parameters d , ζ , κ , λ , μ , and ν are reaction coefficients that modify the corresponding interactions and significantly impact the resulting pattern, while n represents the nutrient concentration of the substrate. Finally, the parameters ξ and θ exert direct control over $a(u)$, indirectly affecting the evolution of c .

2.2 CA Framework

The foundational principles of CA concerning the cells within the utilized grid are the CA cell's state, its rule, and its neighborhood [2]. Also, the grid employed in this implementation is illustrated in Fig. 2 as a square grid, consisting of i rows and j columns. Each cell within the grid encapsulates the continuous values of u , v , and c , which represent the state of the CA. These values are updated in accordance with Eqs. (1)–(3), embodying the rule principle. The updates are performed conditionally, as indicated by the spaces Ω and Ω_c , shown in Fig. 2. Propagation can occur around cells where the concentration of $c(i, j)$ is greater than zero, in particular within the region defined by Ω_c , which is highlighted in color in Fig. 2. The set of points where propagation is permitted constitutes the evolution enable matrix N .

Furthermore, the neighborhood defined by the Laplacian operator delineates the diffusion processes of u and v across space. As shown in [13], the Laplacian operator is discretized. This makes it possible for the operation, which is based on partial differential equations, to be carried out in the discrete framework of

the CA. This discrete form of the Laplacian resembles a Moore neighborhood but with adjusted weights, placing reduced emphasis on the diagonal neighbors of the central cell. This adjustment facilitates a more circular diffusion pattern, a phenomenon commonly observed in natural processes [16]. The Laplacian for the center cell $\ast(i, j)$, which pertains to the concentrations of either u or v , is expressed in Eq. (5).

$$\begin{aligned} \nabla^2 \ast(i, j) = & \frac{1}{\Delta x^2} \left(\frac{7}{20} \left[\ast(i-1, j) + \ast(i+1, j) + \ast(i, j-1) + \ast(i, j+1) \right] \right. \\ & + \frac{1}{10} \left[\ast(i-1, j-1) + \ast(i+1, j-1) + \ast(i-1, j+1) \right. \\ & \left. \left. + \ast(i+1, j+1) \right] - \frac{1}{9} \ast(i, j) \right) \end{aligned} \quad (5)$$

Consequently, the extracted kernel that emerges from Eq. 5 is $L1$ and in the same way, the second kernel $L2$ is reproduced to represent the two different CA rules. The two kernels are seen as follows:

$$L1 = \begin{pmatrix} 1/10 & 7/20 & 1/10 \\ 7/20 & -1/9 & 7/20 \\ 1/10 & 7/20 & 1/10 \end{pmatrix}, \quad L2 = \begin{pmatrix} 0 & 1/20 & 0 \\ 1/20 & -1/5 & 1/20 \\ 0 & 1/20 & 0 \end{pmatrix} \quad (6)$$

Two separate rules have been set up to get a dense biomass in the middle and hyphal tips that are spreading out around the edges, which looks a lot like mycelium-based ELMs. More specifically, this evolution pattern is notably similar to the fungus *Rhizoctonia Solani* [3]. As iterations progress, these rules alternate with the two different kernels $L1$ and $L2$, leading to the emergence of a final evolution pattern characterized by this dual feature.

3 System Design and Architecture

The CA's principles (state, rule and neighborhood) are fully compatible with digital hardware implementations, as they can be effectively represented using memories and logic. These memory units function as matrices which can be modified at each timestep, dictated by the programmed logic. This setup facilitates easy reading from/writing to the memory (state and rule), and efficiently manages local interactions (neighborhood) within each cell of the CA. Moreover, this method is not only highly compatible with existing hardware, but also enhances computational efficiency through significant parallelism [6], where all calculations occur within a narrowly defined time window, specifically the simulation's time step, set by the clock. As a result, the FPGA is a very promising digital platform to implement CAs, incorporating the hardware to effectively support all of the above requirements. The proposed CA was implemented using High-Level-Synthesis (HLS) [10] using the AMD Vitis Software and the Xilinx ZedBoard Zynq 7000 ZC702 Evaluation Board (xc7z020clg484-1).

The algorithm used for the HLS, written in C++ and described in Algorithm 1. The model initiates a loop phase where it's continuously updated at each iteration. Initially, evolution-enabled matrix N is established. Subsequent updates are governed by the *codeselector* variable, a binary vector comprising sequences of 0s and 1s, which determines when the rule update of the CA is altered. The first rule (lines 1–4) facilitates a more circular diffusion by employing a kernel with diminished weights on the diagonal, whereas the second rule utilizes a kernel with equal weights on adjacent and diagonal positions. Equations (1)–(4) and the right Laplacian kernel are used to make changes to the values of the activator u , the inhibitor v , and the main body c . The neighborhood, which is like a circular front, is also multiplied by the RD processes' diffusion term to aid the activator's growth in a radial direction adjacent to the main body. Limiters are then applied to both the activator and inhibitor to prevent excessively large values that could impede the diffusion process. Finally, the states of the activator, inhibitor, and main body are updated.

The memory needed to uphold the state of the CA is divided into 3 two-dimensional matrices containing the activator, the inhibitor, and the main body's values, which utilize the board's BRAM blocks. The rule principle of the CA is implemented in the FPGA, utilizing the board's look-up-tables (LUTs), which are reprogrammable combinational logic blocks containing logic gates capable of handling the complex RD processes. As seen in Fig. 3, the design comprises an *Evolution Module*, which is responsible for computing of the next state of the mycelium's evolution, and a *Top Module*. The *Evolution Module* takes as input the *Evolution Parameters* first initialized in the *Top Module*, and also returns the state of the evolution as *data out*, and the three state tables u , v , and c to it in order to get renewed for the next iteration of the evolution. So, there is a feedback mechanism between the *Evolution Module* and the *Top Module* to take the previously computed state as an input to the next iteration. Finally, it is noted that the values of the utilized parameters are shown in Table 1.

Table 1. Utilized Evolution Parameters.

Parameter	Value	Parameter	Value
θ	2.5	ν	1
ζ	625	ξ	0.49
κ	0.5	a_{max}	20
λ	0.8	i_{max}	30
μ	2.6	<i>threshold</i>	0.5
d	30	ρ	0.7

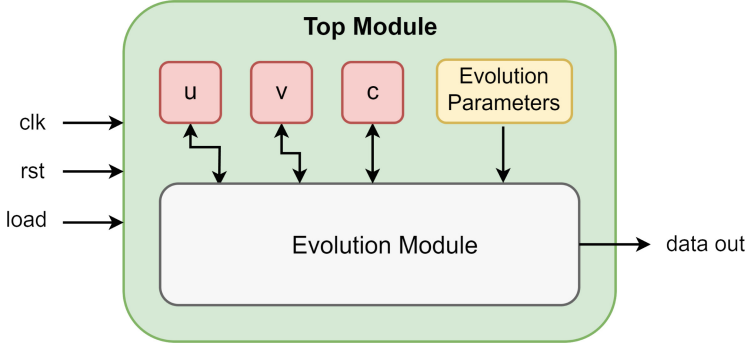


Fig. 3. Qualitative block diagram of the design in the FPGA. (Color figure online)

Algorithm 1. CA RD Model - Iterative Process

```

1:  $N \leftarrow \text{evolution enable matrix}$ 
2:  $n \leftarrow \text{substrate nutrient matrix}$ 
3:  $\text{codeselector} = 1 ? L \leftarrow L1 : L \leftarrow L2$ 
4:  $u_{\text{new}} \leftarrow u_{\text{old}} + N \circ (\text{conv}(u_{\text{old}}, L)) + n\zeta(\kappa u_{\text{old}} + u_{\text{old}}^2 - \lambda u_{\text{old}} v_{\text{old}})$ 
5:  $v_{\text{new}} \leftarrow v_{\text{old}} + d(\text{conv}(v_{\text{old}}, L)) + \zeta(\mu u_{\text{old}}^3 - v_{\text{old}})$ 
6:  $c_{\text{new}} \leftarrow c_{\text{old}} + \zeta \nu c_{\text{old}}(a(u_{\text{old}}) - c_{\text{old}})(c_{\text{old}} - \rho)$ 
7:  $u_{\text{old}} \geq \text{threshold} ? a(u_{\text{old}}) \leftarrow \xi : a(u_{\text{old}}) \leftarrow \xi - \theta(u_{\text{old}} - \text{threshold})$ 
8:  $a(u_{\text{old}}) < 0 ? c_{\text{new}} \leftarrow 1$ 
9:  $u_{\text{new}} < 0 ? u_{\text{new}} \leftarrow 0$ 
10:  $u_{\text{new}} > a_{\text{max}} ? u_{\text{new}} \leftarrow a_{\text{max}}$ 
11:  $v_{\text{new}} > i_{\text{max}} ? v_{\text{new}} \leftarrow i_{\text{max}}$ 
12:  $v_{\text{old}} \leftarrow v_{\text{new}}, u_{\text{old}} \leftarrow u_{\text{new}}, c_{\text{old}} \leftarrow c_{\text{new}}$ 

```

4 Simulation Results

In Figs. 4(d)–(h), the growth of the CA-based mycelium is depicted at various simulation steps, highlighting a dense biomass center marked by both mycelium cells and an activator, with hyphae visible on the stem. The inhibitor's density increases toward the edges and decreases toward the center, illustrating its inverse relationship with both the activator and cell states. Figures 4(a), (b), and (h) show that the patterns of the activator, inhibitor, and the mycelium's main body are closely aligned. Following, Fig. 4(c) displays the substrate, which can affect the mycelium's ability to grow; more conducive areas (indicated by white/bright colors) promote a higher activator concentration and a denser mycelium body.

It is important to note that in order to achieve the most accurate outcomes, a central zone was initialized with minimal concentrations of the activator, the inhibitor, and the main body, representing the initial state of the mycelium, along with all the corresponding RD parameters. It should be also considered that the initialization in the central area is conducted randomly. Considering

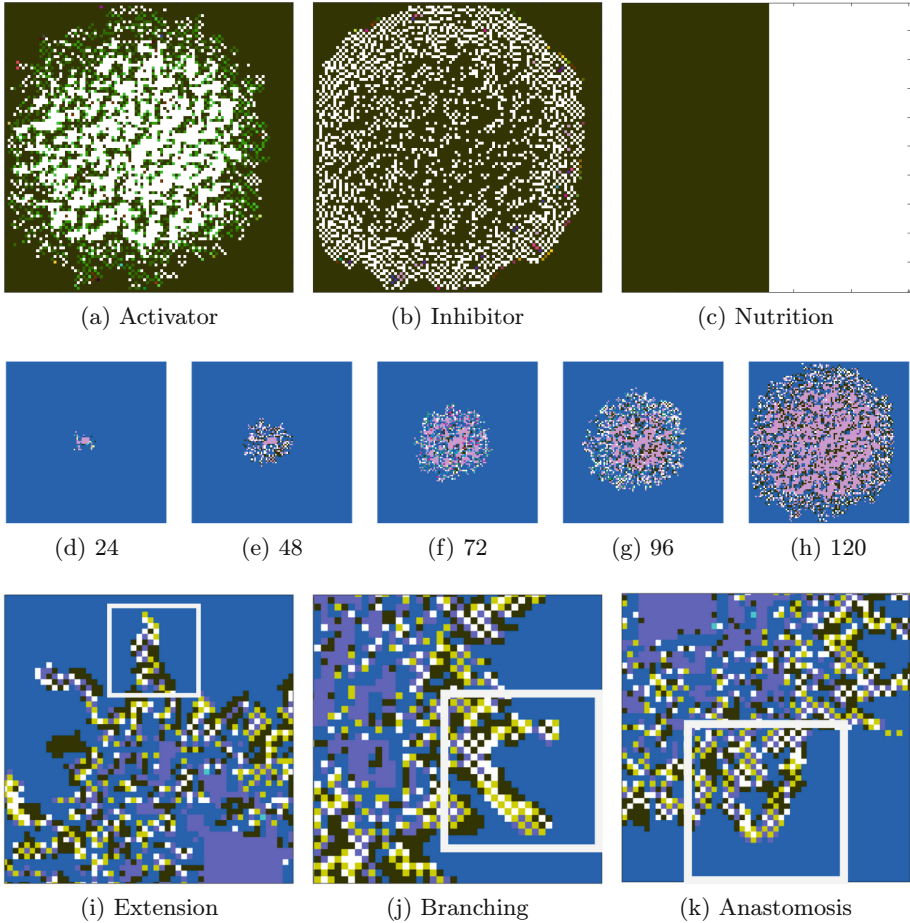


Fig. 4. (a) Activator u , (b) Inhibitor v , and (c) Nutrient concentrations across the grid at 120 simulation's iterations. (d)–(h) Mycelium main body concentration c at different simulation's iterations. In (a)–(h), the lighter in color a cell is, the higher its concentration. (i)–(k) Mycelium's tip mechanisms. Simulations were conducted for 120 iterations at a 100×100 grid. (Color figure online)

the sensitivity of reaction-diffusion systems [22], this initial randomness is propagated across the evolving pattern with each iteration.

Reaction-diffusion models featuring inhibitors and activators offer a compelling framework for simulating the complex behaviors of mycelial growth, including tip extension, branching, and anastomosis, that are readily observable in the mycelium's hyphae. The validation of the result is based on the appearance of these three characteristics (extension, branching, and anastomosis) in every run of the simulation, as it is initialized with different u and v tables. It is noted that the proposed model is a generic reaction-diffusion model that is able,

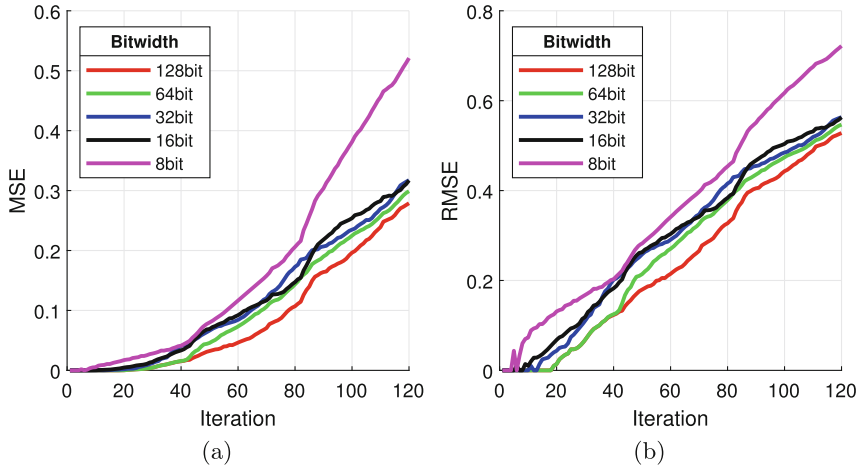


Fig. 5. (a) Mean Squared Error (MSE) and (b) Root Mean Square Error (RMSE) observed during the iterations of mycelium's evolution between software and hardware simulation using different bitwidths. (Color figure online)

with the proper tuning of its parameters, to replicate the evolution of mycelial networks denoted by the aforementioned three characteristics. As referred in Sect. 2, the activator promotes growth and propagation, while the inhibitor regulates and constrains this growth. Tip extension is naturally simulated as the activator diffuses outward, driving the elongation of mycelial hyphae. Branching emerges when local variations in activator and inhibitor concentrations lead to spatial gradients, prompting the formation of new growth points. Anastomosis, the fusion of hyphal strands can be effectively simulated through the interaction of activator fronts, where collisions inhibit further propagation. These mechanisms are observed and showcased in Figs. 4(i)–(k). These processes have been regulated due to mechanisms for nutrient uptake, as indicated by matrix n , which also indicates the density of the activator u .

Other than that, the modeling scale involves discretizing space into square lattice sites, each with a side length of 100 μm [9], encompassing a simulated grid area of 1 cm^2 . With a small tradeoff in accuracy and adjustment of parameters, even larger mycelium structures can be emulated on the same grid, offering flexibility in modeling various mycelial configurations.

There have been various evolution implementations of the proposed design using different bitwidths for the computations. As one can notice in Fig. 5, using 16 bits, the evolution performs really well with similar MSE and RMSE metrics compared to the higher bitwidth options during the iterations. For 16 bit bitwidth, the hardware resources are seen in Table 2. It has been observed that using 32 bits, the accuracy of the evolution was almost the same as 16 bits, but the utilization of BRAMs and DSPs was doubled, as was the consuming power. For this reason, the 16 bit bitwidth option is considered optimal for the proposed design.

Table 2. Utilization of the FPGA’s resources after implementation.

Utilization of the FPGA’s resources after implementation			
Resource	Utilization	Total Available	Utilization %
LUT	5787	53200	10.88
LUTRAM	308	17400	1.77
FF	2142	106400	2.01
BRAM	66	140	47.14
DSP	63	220	28.64
IO	23	200	11.50
BUFG	1	32	3.13

Finally, the proposed algorithm has been tested on a CPU (Intel i9 9900k) and also on a GPU (GeForce RTX 2060 Super), as referred to in Table 3. However, FPGA’s parallelization capabilities, along with its low power consumption and portability, make it the most efficient candidate. To elaborate more, for the same computation load, FPGA appears to have the lowest energy need between CPU and GPU implementations, while utilizing the smaller area footprint as well. It is noted that the clock of the board is set to 10 ns, providing a running frequency of 100 MHz. Further rationale for the FPGA suitability, including specific advantages and future applications, is discussed in the concluding Sect. 5 of this paper.

Table 3. Efficiency, execution time, and total energy metrics measured on CPU, GPU, and FPGA.

Platform	Power (<i>Watts</i>)	Execution Time (<i>ms</i>)	Total Energy (<i>mJ</i>)
CPU	43.1	40	1724
GPU	84.5	4.2	354.9
FPGA	0.37	35.1	12.9

5 Conclusions and Future Work

In this paper, a digital twin of a mycelium-based Engineered Living Material (ELM) within a Field-Programmable Gate Array (FPGA) has been successfully implemented. The model successfully mimics key behaviors of mycelium, such as tip growth, branching, merging, and nutrient absorption, while replicating typical bio-inspired patterns and also utilizing core features of the ELM, which are a dense central biomass and distinctive tips at the mycelium’s stem. The proposed implementation showcases a significant speed boost and enhanced energy

efficiency compared to basic CPU-powered software execution. This development marks a significant step forward in the simulation and understanding of mycelium-based materials within a controlled digital twin environment and provides an ideal framework for simulating such biological organisms. The current implementation primarily simulates and observes ELM behavior without allowing interactive feedback between biological and digital systems.

In future work, exploring the establishment of a bidirectional communication between the mycelium-based ELM and the FPGA holds great promise. By incorporating analog-to-digital (ADCs) and digital-to-analog converters (DACs) within the FPGA, it could become feasible not only to decode biological signals into digital data for detailed analysis but also to transmit operational commands back to the living material, enabling real-time adjustments based on the ELM's continuous monitoring. The compact form of the FPGA board, along with its showcased efficiency, can make the proposed implementation a really versatile and mobile tool that can easily be deployed for its integration with ELMs. Also, by utilizing the analog ports of the FPGA, the interference with environmental sensors can be exploited to measure metrics like humidity, temperature, and soil quality. So, the digital twin could dynamically reflect and adapt to the conditions affecting the biological counterpart, ensuring a high-fidelity emulation. The model's superior insights can be enhanced by using memristive nanocircuits, which enable non-discrete, continuous computation suitable for analog biological signals. In the end, the FPGA prototype could facilitate the implementation of a specific integrated circuit (ASIC) as an embedded low-power proposal with parallel core processing for mycelium-based ELM emulation.

Acknowledgments. This work has been supported by the framework of the FUNGATERIA project, which has received funding from the European Union's HORIZON-EIC-2021-PATHFINDER CHALLENGES program under grant agreement No. 101071145.

References

1. Adamatzky, A.: *Fungal Machines: Sensing and Computing with Fungi*, vol. 47. Springer, Heidelberg (2023). <https://doi.org/10.1007/978-3-031-38336-6>
2. Adamatzky, A., Alonso-Sanz, R., Lawniczak, A.: *Automata-2008: Theory and Applications of Cellular Automata*. Luniver Press (2008)
3. Boswell, G.P.: Modelling mycelial networks in structured environments. *Mycol. Res.* **112**(9), 1015–1025 (2008)
4. Bozzini, B., Lacitignola, D., Sgura, I.: Morphological spatial patterns in a reaction diffusion model for metal growth. *Math. Biosci. Eng.* **7**(2), 237–258 (2010)
5. Chatzinikolaou, T.P., Karamani, R.E., Fyrigos, I.A., Sirakoulis, G.C.: Handling sudoku puzzles with irregular learning cellular automata. *Nat. Comput.* **23**(1), 41–60 (2024). <https://doi.org/10.1007/s11047-024-09975-4>
6. Dourvas, N., Tsompanas, M.A., Sirakoulis, G.C., Tsalides, P.: Hardware acceleration of cellular automata Physarum polycephalum model. *Parallel Process. Lett.* **25**(01), 1540006 (2015)

7. Dourvas, N.I., Sirakoulis, G.C., Adamatzky, A.I.: Parallel accelerated virtual physarum lab based on cellular automata agents. *IEEE Access* **7**, 98306–98318 (2019)
8. Dourvas, N.I., Sirakoulis, G.C., Tsalides, P.: A GPGPU physarum cellular automaton model. *Appl. Math.* **10**(6), 2055–2069 (2016)
9. Fricker, M.D., Heaton, L.L., Jones, N.S., Boddy, L.: The mycelium as a network. *Fungal Kingdom* 335–367 (2017)
10. Gajski, D.D., Dutt, N.D., Wu, A.C., Lin, S.Y.: *High-Level Synthesis: Introduction to Chip and System Design*. Springer, Heidelberg (2012)
11. Houette, T., Maurer, C., Niewiarowski, R., Gruber, P.: Growth and mechanical characterization of mycelium-based composites towards future bioremediation and food production in the material manufacturing cycle. *Biomimetics* **7**(3), 103 (2022)
12. Landge, A.N., Jordan, B.M., Diego, X., Müller, P.: Pattern formation mechanisms of self-organizing reaction-diffusion systems. *Dev. Biol.* **460**(1), 2–11 (2020)
13. O'Reilly, R.C., Beck, J.M.: A family of large-stencil discrete Laplacian approximations in three-dimensions. *Int. J. Numer. Methods Eng.* 1–16 (2006)
14. Pavlidis, N., Perifanis, V., Chatzinikolaou, T.P., Sirakoulis, G.C., Efraimidis, P.S.: Intelligent client selection for federated learning using cellular automata (2023)
15. Peeters, E., Saluena Martin, J., Vandeloek, S.: Growing sustainable materials from filamentous fungi. *Biochemist* **45**(3), 8–13 (2023)
16. Sirakoulis, G.C., Karafyllidis, I., Thanailakis, A.: A cellular automaton for the propagation of circular fronts and its applications. *Eng. Appl. Artif. Intell.* **18**(6), 731–744 (2005)
17. Sirakoulis, G., Karafyllidis, I., Mizas, C., Mardiris, V., Thanailakis, A., Tsalides, P.: A cellular automaton model for the study of DNA sequence evolution. *Comput. Biol. Med.* **33**(5), 439–453 (2003)
18. Sugimura, K., Shimono, K., Uemura, T., Mochizuki, A.: Self-organizing mechanism for development of space-filling neuronal dendrites. *PLOS Comput. Biol.* **3**(11), 1–12 (2007)
19. Tao, F., Xiao, B., Qi, Q., Cheng, J., Ji, P.: Digital twin modeling. *J. Manuf. Syst.* **64**, 372–389 (2022)
20. Tsompanas, M.A., Chatzinikolaou, T.P., Sirakoulis, G.C.: Cellular automata application on chemical computing logic circuits. In: Chopard, B., Bandini, S., Dennunzio, A., Arabi Haddad, M. (eds.) *ACRI 2022. LNCS*, vol. 13402, pp. 3–14. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-14926-9_1
21. Udugama, I.A., Lopez, P.C., Gargalo, C.L., Li, X., Bayer, C., Gernaey, K.V.: Digital twin in biomanufacturing: challenges and opportunities towards its implementation. *Syst. Microbiol. Biomanuf.* **1**, 257–274 (2021)
22. Van Gorder, R.A.: A theory of pattern formation for reaction-diffusion systems on temporal networks. *Proc. Roy. Soc. A: Math. Phys. Eng. Sci.* **477**(2247), 20200753 (2021)
23. Zheng, Q., Shen, J., Xu, Y.: Turing instability in the reaction-diffusion network. *Phys. Rev. E* **102**(6), 062215 (2020)