1-1-2003

Biology Goes Digital: An array of 5,700 Spartan FPGAs brings the BioWall to "life"

Christof Teuscher
Portland State University

Gianluca Tempesti

Citation Details

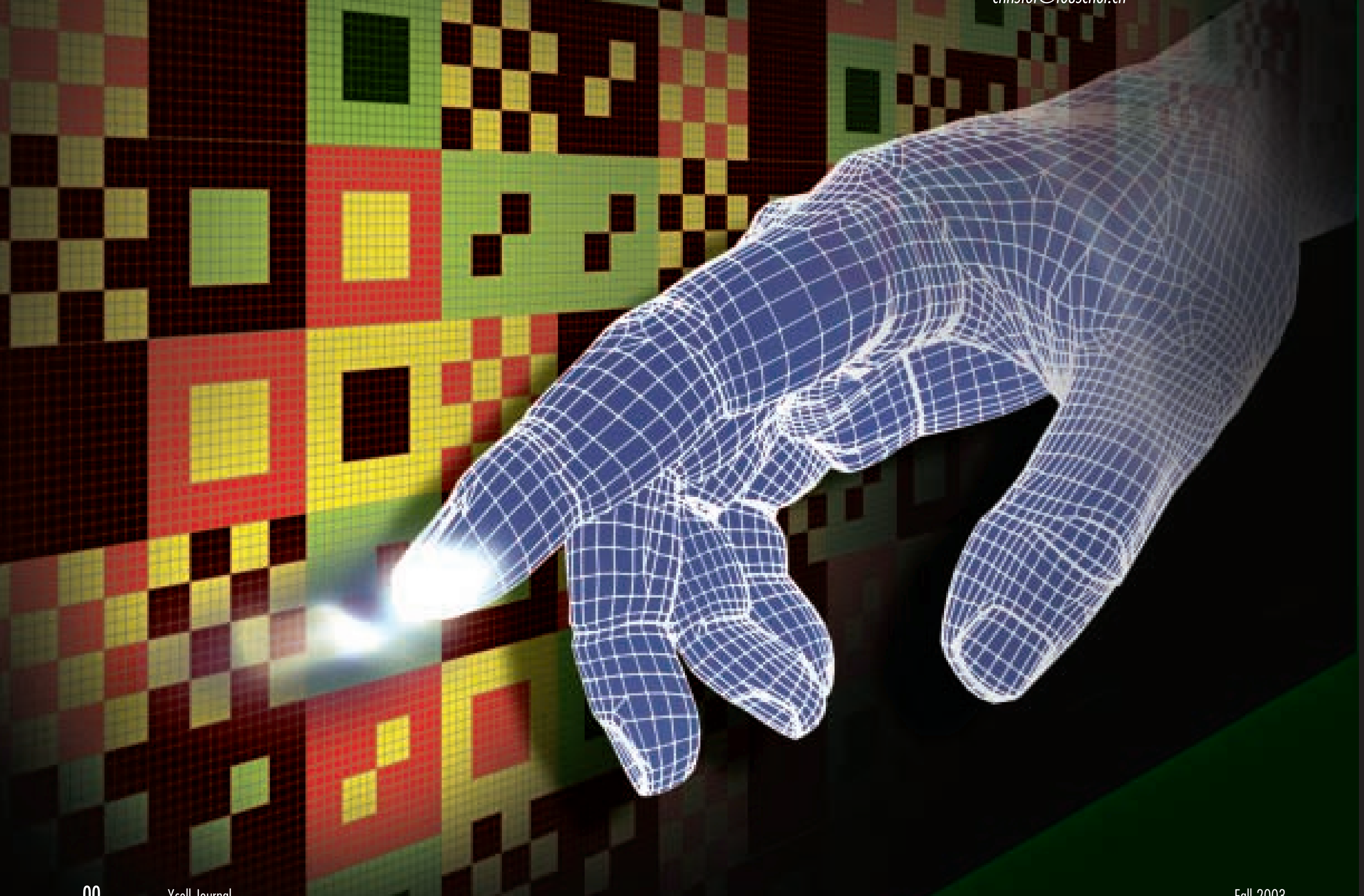
G. Tempesti and C. Teuscher. Biology Goes Digital: An array of 5,700 Spartan FPGAs brings the BioWall to "life."Xilinx Xcell Journal, Fall 2003.

Biology Goes Digital

An array of 5,700 Spartan FPGAs brings the BioWall to "life."

by Gianluca Tempesti, Ph.D.
Assistant Professor
Swiss Federal Institute of Technology
gianluca.tempesti@epfl.ch

Christof Teuscher
Research Assistant
Swiss Federal Institute of Technology
christof@teuscher.ch



A unique, scientific research instrument and art piece, the BioWall models bio-inspired electronic tissues capable of evolution, self-repair, self-replication – and learning.

Much pure and applied scientific research has focused on replicating biological functions in digital hardware. Here, at the Logic Systems Laboratory of the Swiss Federal Institute of Technology in Lausanne (EPFL), we have utilized 5,700 Xilinx Spartan™ FPGAs, in multiple configurations, to build bio-inspired computing machines that exploit three essential biological models:

- **Phylogenesis** – the history of the evolution of the species
- **Ontogenesis** – the development of an individual as directed by his genetic code
- **Epigenesis** – the development of an individual through learning processes (nervous system, immune system), influenced both by genetic code (the innate) and environment (the acquired).

Although we have individually and jointly investigated all three models, we have concentrated on the ontogenetic model through the Embryonics (embryonic electronics) Project. This project studies the development of multi-cellular organisms for the purpose of obtaining in digital hardware some of the features of biological organisms, notably growth and fault tolerance.

Our work has attracted a flattering amount of interest in the most varied and sometimes unexpected milieus. Among the most unexpected sources of funding and support came from Mrs. Jacqueline Reuge. Mrs. Reuge decided to fund the construction of the BioWall to display the principles of embryonics within a museum built to honor the memory of her late husband.

Her generous support has allowed us to maintain our tradition of verifying our theoretical concepts in hardware. Without Mrs. Reuge's support, we could not have constructed a computing machine of such magnificent proportions.

We named this machine BioWall because of its biological inspiration, as well as its size (5.3m x 0.6m x 0.5m = 3.68m^3, or 130 cubic feet). The main purpose of creating the BioWall is to demonstrate the features of our embryonics systems to the public through a visual and tactile interaction (Figure 1).

Figure 1 – The BioWall reacts to touch.

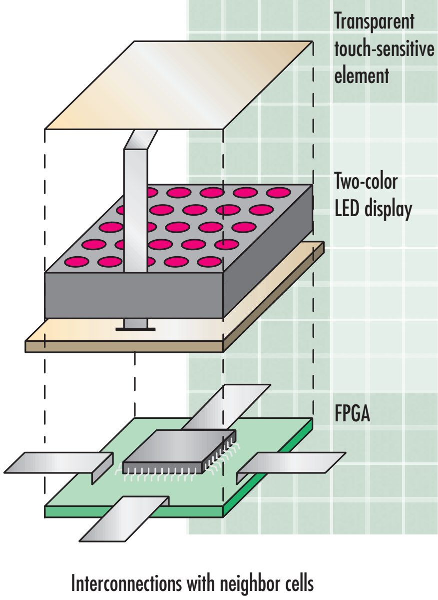


Figure 2 – The BioWall's basic building block

Bio-Inspired Machines

We implemented – for the first time in actual hardware – an organism endowed with all of the features of an embryonics machine, as it has often been defined in the literature. One of the functions of our organism is the BioWatch, which counts hours, minutes, and seconds. It demonstrates the growth and self-repair capabilities of our systems.

The implementation of the BioWatch would have been sufficient to justify the effort that went into the construction of our embryonic BioWall. However, in developing our machine, we quickly realized that the capabilities of such a platform were not limited to a single application. In fact, it is an ideal platform to prototype many different kinds of two-dimensional cellular systems, which are systems comprising arrays of small, locally connected elements.

For example, "cellular automata" (CA) are very common environments in bio-inspired research. The BioWall is ideally suited to the implementation of CAs, but it is by no means limited to it. We have only begun to explore the possibilities of the BioWall as a research tool.

Xilinx Behind the Scenes

We built the BioWall to demonstrate an embryonic machine. The structure of such machines is hierarchical: Organisms (application-specific systems) are created by the parallel operation of a number of cells (small processors). Each cell is implemented as an array of molecules (programmable logic elements).

To implement this kind of machine, the BioWall is structured as a two-dimensional tissue comprising units (each unit corresponds to a molecule). As shown in Figure 2, each unit consists of:

- An input element (a touch-sensitive membrane)
- An output element (an array of 64 two-color LEDs)
- A programmable computing element (a Xilinx Spartan XCS10XL FPGA).

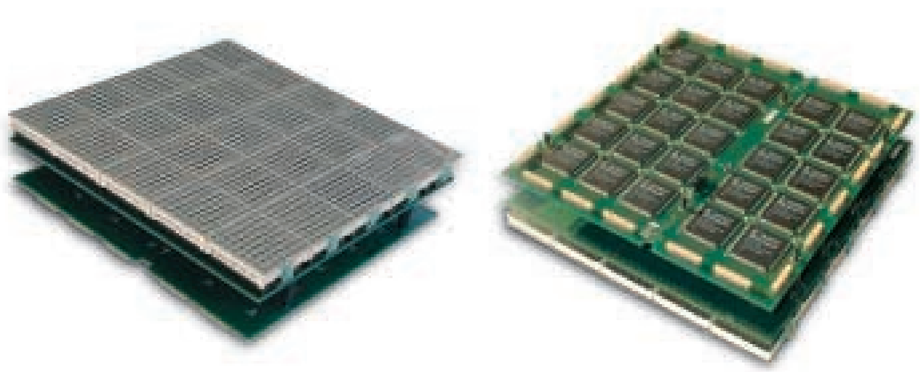


Figure 3 – Front and back view of 25 FPGAs assembled on a BioWall panel

The circuits are mounted on double boards. The logic board hosts 25 FPGAs, and the display board holds the displays and membranes (Figure 3). The two boards are rigidly bound together and connected by a bus to allow two-way communication between the logic and the display (a dedicated circuit on the logic board automatically distributes the signals to and from the displays).

On the logic board, the Spartan devices are placed in a regular two-dimensional grid. A subset of the pins of each FPGA (approximately 20 per side) are used to make a direct pin-to-pin connection between each circuit and its four cardinal neighbors. The pins of the FPGAs placed along the edges of the board are brought to a set of connectors to allow the pin-to-pin association to continue across boards (Figure 4), thus creating perfectly uniform surfaces of FPGAs spanning as many boards as required.

The remaining pins are connected to a centralized circuit that handles the distribution of the global signals (the clocks, resets, and FPGA configurations) arriving from the outside.

We have built 228 such boards (not including spare materials) for a total of 5,700 units. The architecture of the boards implies that they can be seamlessly connected with each other to form a uniform surface of any shape and size. Throughout the development phase and lifetime of the machine, we have so far constructed several independent machines:

- A 3,200-unit machine (Figure 5) displayed at the Villa Reuge museum
- A 2,000-unit machine kept in our laboratory to develop and test new applications

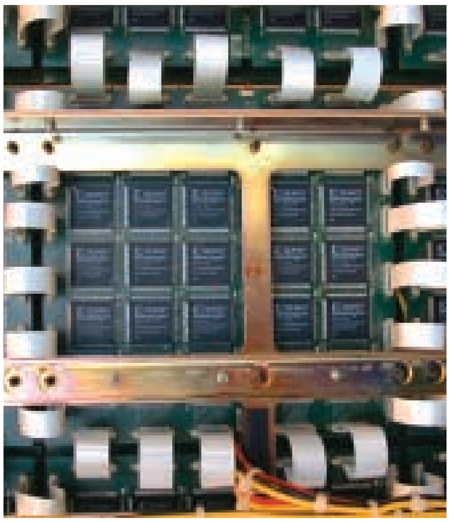


Figure 4 – Connectors allow pin-to-pin association to join multiple boards.

- A 150-unit machine embedded together with the necessary control logic to charge applications into a suitcase for portability
- A 4,000-unit machine that will be on display at the Telecom ’03 conference in Geneva in October 2003.

This tissue of 5,700 FPGAs represents an impressive amount of computational power, coupled with I/O interfaces (comprising the membranes and LED arrays) that allow for large-scale visual and tactile interaction. The advantage of this solution is the size of the display, which enables an immediate interaction with applications normally limited to software simulation on a computer screen. Furthermore, the computing power and programmability of the Xilinx FPGAs enables the prototyping of new bio-inspired systems.

In the current version of the BioWall, the Xilinx FPGAs can only be programmed with the same configuration, which limits the functionality of the units to the 10,000 equivalent logic gates of Spartan devices. The considerable delays inherent in propagating a global signal over distances measured in meters seriously limit the clock speed. Considering the role of the BioWall as a demonstration tool, we have not tried to push the clock to its limits, as the current frequency of 1 MHz is more than adequate if coupled with the massive parallelism of the machine.

Besides the I/O capabilities of the membranes and LED displays, a set of modules placed on the borders of the machine allows the tissue to be interfaced with standard logic either via a PC or directly with user-defined modules. The modules allow access only to the borders of the array, but, if necessary, signal propagation logic can be programmed in the FPGAs.

The software tools developed for the BioWall are rudimentary but complete. A simple interface on a PC allows users to define a set of files to configure the tissue. Four kinds of files are currently defined: the configuration file for the Xilinx FPGAs, and three different formats used to send user-defined data on the input pins at the borders of the tissue (used, for example, to provide an initial configuration for a cellular automation). The values on the output pins at the borders of the tissue can be read by the PC and either stored on disk or used as required.

Applications

The cellular structure of the BioWall is well suited to the implementation of all sorts of bio-inspired applications. The BioWall can exploit the versatility inherent in its programmable logic and in its architecture to implement hardware inspired by all the three models of biological inspiration: phylogenesis, ontogenesis, and epigenesis.

BioWatch

To illustrate the implementation of the BioWatch application on the BioWall, we will introduce a slightly simplified example. Whereas the complete BioWatch is an organism capable of counting hours, min-

Figure 5 – The BioWall on display in the Villa Reuge Museum

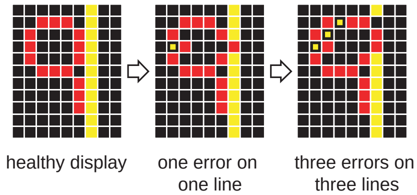


Figure 7a – Molecular self-repair of the counter application

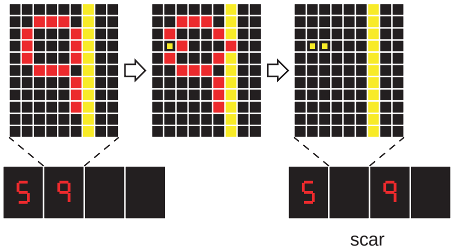


Figure 7b – Cellular self-repair of the counter application

utes, and seconds, the "counter application" we describe here only counts seconds. Otherwise, the principles of operation of the two machines are identical.

The counter counts seconds, from 00 to 59. From left to right, the display shows tens of seconds (from 0 to 5), units of seconds (0 to 9), and a spare zone, which remains inactive during normal operation (Figure 6a). The counter is divided into four cells: two active (indicating tens and units, respectively) and two spare. Each unit of the BioWall is a molecule of the

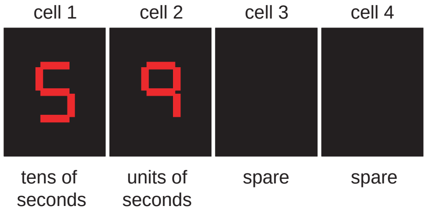


Figure 6a – Counter implemented on BioWall

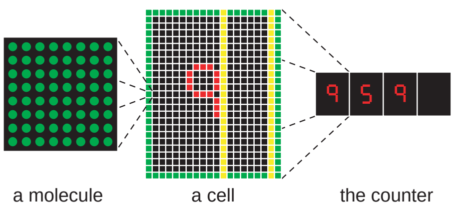


Figure 6b – Hierarchical structure of the counter application

embryonics hierarchy. A cell is then a mosaic of (20 x 25) 500 molecules (Figure 6b), and contains two repair columns for a total of (2 x 25) 50 molecules.

You have control over the "life" of each molecule. A stuck-at-fault can be inserted in any molecule simply by pressing on the corresponding unit's membrane. The fault detection mechanism included in the embryonics molecular layer (embedded into the programmable logic of the Spartan FPGAs) automatically detects the error and activates the molecular self-repair mechanism. A "dead" molecule is instantly replaced by the neighbor immediately to its right, and so on, until the nearest repair column (Figure 7a).

The limits of this kind of self-repair imply that only a single molecule per line, between two repair columns, can be killed. If this constraint is respected, the cell survives any amount of faults, although the figure displayed is distorted. Each cell can thus tolerate up to two faults per line (one fault between each pair of repair columns), equaling (2 x 25) 50 faults in total.

If the above rule is not respected, and several faults are inserted on the same line of the same cell between two repair columns, the molecules can no longer repair themselves and the cell dies. However, the death of a cell does not imply the death of the organism. It is instantly replaced by a spare cell to its right (Figure7b), while the dead cell is switched off and becomes a scar.

Throughout this self-repair process, the counter continues to work without fault. The tissue remembers its state and recovers the correct time after repair. Moreover, we are currently implementing an "unkill" mechanism to address the issue of transient faults. If a sufficient number of faults are removed (by pressing the membranes of dead molecules), this mechanism will automatically re-activate a dead cell, which will recover its functionality (and its state) within the organism.

The self-repair capabilities of the embryonics machines are based on a general principle of life – cell differentiation. Each organism is a collection of cells, each containing a full copy of the genetic program, the genome. This structure makes the whole organism extremely robust, because each cell contains the complete plan and can therefore replace any other defective cell.

Nevertheless, like all artificial and natural organisms, the death of a sufficiently large number of cells cannot be repaired, causing the death of the organism. The advantage of the controlled environment in which the machine operates is that the death of the organism causes a general reset of the system, the obliteration of all injected faults, and the "birth" of a new, perfectly functioning machine.

The complete implementation of the BioWatch on the BioWall uses eight cells of (20 x 20) 400 molecules each, with two spare columns of molecules in each cell. Six of the eight cells are active during normal operation, while two are spares, ready to replace a dead cell. All the theories of the Embryonics Project have been tested and verified in hardware through this implementation.

Self-Replicating Loops

Initiated by von Neumann more than 50 years ago, the study of self-replicating computing machines has produced a plethora of results. Much of this work is motivated by a desire to understand the fundamental information processing principles and algorithms involved in self-replication independently of their physical manifestation. The construction of artificial self-replicating machines can have diverse applications ranging from nanotechnology to space exploration to reconfigurable computing tissues.

To render the self-replication process more interactive and visible, we implemented self-replicating loops on the BioWall, initially of size 2 x 2 and then of variable size (Figure 8). In this implementation, every unit of the BioWall is one cell of the CA. Pressing on the membrane of a unit belonging to a loop causes the unit to replicate in one of four cardinal directions (Figure 9).

Turing Neural Networks

In 1948, Alan Turing wrote a little-known report entitled "Intelligent Machinery." Turing never had great interest in publicizing his ideas, so the paper went unpublished until 1968, 14 years after his death.

Few people know that Turing's "Intelligent Machinery" paper contains a fascinating investigation of different connectionist models that would today be called neural networks. In describing randomly connected networks of artificial neurons, Turing wrote one of the first manifests of the field of artificial intelligence (although he did not use this term).

Recently, we implemented Turing's

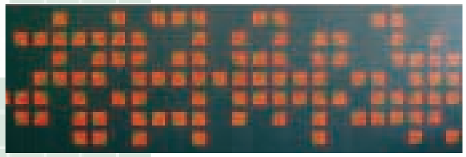


Figure 8 – Self-replicating loops on the BioWall

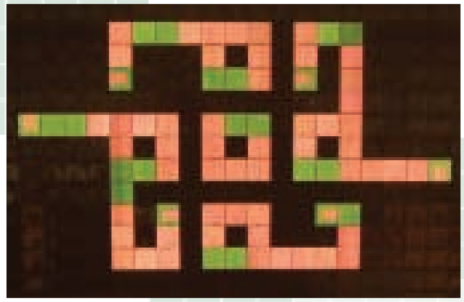


Figure 9 – Self-replication of a set of loops

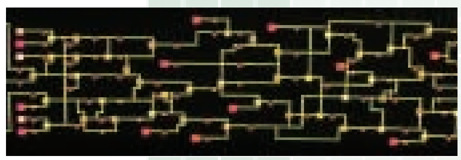


Figure 10 – Turing neural networks on the BioWall

Figure 11 – Two-dimensional Firefly

neural networks on the BioWall's reconfigurable tissue (Figure 10). Each of the 3,200 units of the machine can be interactively configured by choosing one out of five possible functions:

1. Empty cell
2. Neuron
3. Connection
4. Synapse
5. Input cell.

You can discover and affect the behavior of this "unorganized" machine by opening and closing synapses, "organizing" the machine and modifying the network's inputs. All modifications occur by simply pressing on the touch-sensitive membranes. This application is first and foremost a demonstration of Turing's neural networks on reconfigurable hardware (and to the best of our knowledge, the first one). However, it also exemplifies the fusion of the ontogenetic and epigenetic models in a single artificial tissue.

Firefly

In 1997 the Logic Systems Laboratory presented an evolving hardware system called Firefly, based on a cellular programming approach, in which parallel cellular machines evolve to solve computational tasks. The computational task studied – and successfully solved – is known as "synchronization": Given any initial configuration, the nonuniform CA must reach, within M time steps and using only local connections, a final configuration in which all cells oscillate synchronously between all 0s and all 1s on successive time steps.

The novelty of Firefly is that it operates with no reference to an external device, such as a computer that carries out genetic operators. Thus, the Firefly demonstrates online autonomous evolution.

The original Firefly machine was able to find a solution for a one-dimensional CA. Subsequently, we have been able to evolve, on the BioWall's 3,200 FPGAs, a CA that solves the synchronization task in two dimensions.

The implementation on the BioWall (Figure 11) consists of a two-state, nonuniform CA, in which each cell (FPGA) may contain a different rule. The cells' rule tables are encoded as a bit-string, known as the genome. This genome has a length of 25 = 32 bits for our two-dimensional CA (the binary CA has a neighborhood of 5).

Rather than employ a population of evolving CAs, our algorithm evolves a single, nonuniform CA the size of the entire BioWall (one cell of the CA in each unit of the BioWall, or 3,200 cells) whose rules are initialized at random. Initial configurations are then randomly generated and for each configuration the CA is run for M time steps.

Each cell's fitness is accumulated over C initial configurations: a single run's score is 1 if the cell is in the correct state after M + 4 iterations, and 0 otherwise. The local fitness score for the synchronization task is assigned to each cell by considering the last four time steps (M + 1 to M + 4). If the sequence of states over these steps is precisely 0-1-0-1, the cell's fitness score is 1; otherwise this score is 0.

After every C configuration, the rules are evolved through crossover and mutation. This evolutionary process is performed in a completely local manner; that is, genetic operators are applied only between directly connected cells.

Unlike standard genetic algorithms, where a population of independent problem solutions globally evolves, our approach involves a grid of rules that co-evolves locally. The CA implemented on the BioWall performs computations in a completely local manner, each cell having access only to its immediate neighbors' states. In addition, the evolutionary process is also completely local, because the application of genetic operators as well as the fitness assignment occurs locally.

Using the above-described cellular programming approach on the BioWall, we have shown that a nonuniform CA of radius 1 can be evolved to successfully solve the synchronization task. In addition, after having found a set of successful rules, our machine allows the state of each CA cell to be changed by pressing on its membrane. You can then observe how the machine re-synchronizes the 3,200 cells.

DNA Sequence Comparison

The comparison and alignment of characters taken from a finite alphabet is a fundamental task in many applications, ranging from full-text search to computational biology. In particular, string comparison is a critical issue in the field of molecular biology.

In fact, both DNA fragments and proteins can be represented as sequences of characters (taken from alphabets of four and 20 symbols, respectively). Sequence similarities provide useful information about the functional, structural, and evolutionary relationships between the corresponding molecules.

Biological sequences may differ because of local substitutions, insertions, and deletions of one or more characters. The complexity of string comparison comes from the large number of possible combinations of these three basic mutations.

The similarity between two strings can

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	
G	0	1	1	1	1	1	1	1	1	1		
G	0	1	1	1	1	1	1	1	2			
A	0	1	2	2	2	2	2	2				
T	0	1	2	2	3	3	3					
C	0	1	2	2	3	3						
G	0	1	2	2	3							
A	0	1	2	3								

Figure 12 – The Needleman-Wunsch algorithms in two dimensions

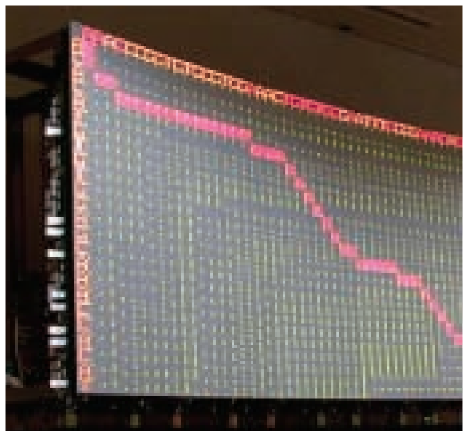


Figure 13 – The Needleman-Wunsch algorithm implemented on the BioWall

be evaluated either in terms of edit distance or similarity score. The edit distance is a measure of the minimum number of edit operations (mutations) required to make the two strings equal to each other.

The similarity score is a measure of the maximum number of residual matches between the two strings. Both metrics can be evaluated in polynomial time by means of dynamic programming techniques.

The key algorithm for evaluating the similarity between two strings of length N and M was developed by Needleman and Wunsch and takes O (N x M) steps to complete execution. The two-dimensional structure of the algorithm (Figure 12) makes it suitable for a parallel implementation on a systolic array.

In particular, hardware parallelism can be exploited to perform string comparison in O (N + M) steps. In this experiment, we present a parallel implementation of the Needleman-Wunsch algorithm on the BioWall (Figure 13).

The BioWall cannot compete in performance with existing parallel implementations of the Needleman-Wunsch algorithm, because it suffers from the typical performance limitations of a large prototyping platform. Nevertheless, the implementation of the Needleman-Wunsch algorithm on the BioWall is a significant design experiment in the field of reconfigurable computing, because of the peculiarities of the target architecture.

Conclusion

The current configuration of the BioWall is a mosaic of more than 3,000 transparent electronic modules. Each module enables visitors to communicate with the surface simply by touching it with their fingers. The BioWall calculates its new status and indicates it immediately on an electronic display. The usefulness of this approach has been demonstrated through a number of experiments.

In fact, the applications presented here are just a small sample of the capabilities of the BioWall – capabilities that we are still discovering. The cellular structure of the machine makes it an ideal platform for prototyping bio-inspired systems.

The size and structure of the BioWall impose a certain number of limitations, such as clock speed. But its complete programmability provides outstanding versatility, and the visual and interactive components of the system are invaluable tools both for the dissemination of ideas as well as the verification of research concepts often limited to software simulations.

Some of the other bio-inspired applications we have implemented or plan to implement on our machine include L-systems, ant simulations, predator-prey environments, other kinds of CAs, and more conventional artificial neural networks.

We invite you to come and "play" with the machine at one of the events at which it will be on public display, or even at our laboratory.

Finally, we are extremely interested in putting our machine at the disposal of other research groups interested in hardware realizations of their ideas and concepts. For more information, please visit us at *lslwww.epfl.ch/biowall/*.

[Daniel Mange, André Stauffer, Fabien Vannel, André Badertscher, and Enrico Petraglio also contributed to this article.]

Photos: © André Badertscher, Alain Herzog, EPFL.