Embryonics: A Microscopic View of the Molecular Architecture

Daniel Mange, André Stauffer, and Gianluca Tempesti

Logic Systems Laboratory, Swiss Federal Institute of Technology, IN-Ecublens, CH-1015 Lausanne, Switzerland. E-mail: {name.surname}@epfl.ch

Abstract. The ontogenetic development of living beings suggests the design of a new kind of multicellular automaton endowed with novel quasi-biological properties: self-repair and self-replication. In the framework of the Embryonics (embryonic electronics) project, we have developed such an automaton. Its macroscopic architecture is defined by three features: multicellular organization, cellular differentiation, and cellular division which are described in a companion paper [5]. In order to cope with the complexity of real problems, the cell is itself decomposed into an array of smaller elements, the molecules, themselves defined by three features: multimolecular organization, self-test and self-repair, and finally cellular self-replication, which is the basis of the macroscopic process of cellular division. These microscopic properties are illustrated by the example of an up-down counter. Finally, we propose a design methodology based on three successive configurations of the basic molecular tissue, a novel FPGA. These configurations are analogous to the operation of three kinds of genetic information: the polymerase, ribosomic, and operative genomes.

1 Cell's microscopic features

1.1 Objectives and contents

In all living beings, the string of characters which makes up the DNA, i.e., the genome, is executed sequentially by a chemical processor, the *ribosome*. Drawing inspiration from this biological mechanism, we will realize each cell of an artificial organism by means of a small electronic processor, a *binary decision machine*, executing sequentially the instructions of our artificial genome, the operative genome *OG*. In analogy with the ribosome, which is itself decomposed into smaller parts, the molecules, we will embed our artificial cell into an array of programmable logic devices, an FPGA whose basic elements will be considered as our artificial molecules.

While the macroscopic description of the Embryonics project and the corresponding properties (self-repair and self-replication of the artificial organism) are described in a companion paper [5], Section 1 of this paper is dedicated to the microscopic study of the cell; this study relies on three fundamental features: multimolecular organization (the cell is itself decomposed into an array of physically identical elements, the molecules), fault detection within each molecule

M. Sipper et al (Eds.): ICES'98, LNCS 1478, pp. 185-195, 1998.

and self-repair of the cell (through the replacement of the faulty molecules), and cellular self-replication (each group of molecules forming a mother cell is capable of replicating itself to produce a daughter cell and thus bring about the cellular division described at the macroscopic level [5]). This last mechanism is the object of a formal description by an L-system.

Section 2 will finally propose a design methodology based on three successive configurations of the basic molecular tissue, a novel FPGA; these configurations are analogous to the operation of three kinds of genetic information: the *polymerase genome*, dividing the silicon space in order to realize the macroscopic cellular division, the *ribosomic genome*, building the binary decision machine which constitutes the core of the cell, and the *operative genome*, which defines the particular application.

1.2 Multimolecular organization

In order to implement any digital system, in particular the binary decision machine of our artificial cell, into a reconfigurable array, we require a methodology capable of generating, starting from a set of specifications, the configuration of a homogeneous network of elements, the *molecules*, each molecule defined by an identical architecture and a usually distinct state (the *molecule code* or *MOLCODE*).

To meet our requirements, we have selected a particular representation: the *ordered binary decision diagram* (OBDD). This representation, with its well-known intrinsic properties such as canonicity [1], was chosen for two main reasons:

- it is a graphical representation which exploits well the two-dimensional space and immediately suggests a physical realization on silicon;
- its structure leads to a natural decomposition into molecules realizing a logic test (a diamond), easily implemented by a multiplexer.

We will illustrate the handling of ordered binary decision diagrams through a simple example, an artificial cell realizing an up-down counter. Our choice will lead us to define a field-programmable gate array (FPGA) as a homogeneous multicellular array where each molecule contains a programmable multiplexer with one control variable, implementing precisely a logic test.

Let us consider the relization of the aboved-mentioned modulo-4 up-down counter, defined by the following sequences:

- for $M = 0 : Q1, Q0 = 00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ (counting up);
- for $M = 1 : Q1, Q0 = 00 \rightarrow 11 \rightarrow 10 \rightarrow 01 \rightarrow 00$ (counting down).

It can be verified that the two ordered binary decision diagrams $Q1$ and $Q0$ of Figure 1a (where each diamond represents a multiplexer, each square an input boolean value, and each diamond embedded in a square a 1-bit memory, i.e., a flip-flop) correspond to a possible realization of the counter [6](pp. 132-135, 239-240).

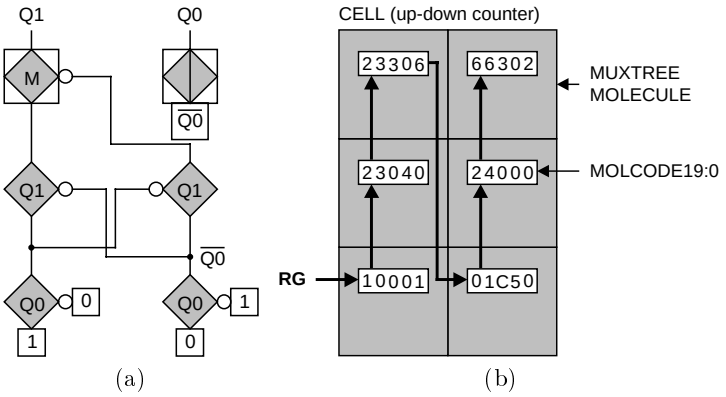


Fig. 1. Modulo-4 up-down counter. (a) Ordered binary decision diagram. (b) Multimolecular implementation of the artificial cell with six MUXTREE molecules; *RG*: ribosomic genome (sum of the $MOLCODE$).

The reconfigurable molecule, henceforth referred as MUXTREE (for multiplexer tree), consists essentially of a programmable multiplexer (with one control variable), a D-type flip-flop, and a switch block allowing all possible connections between two horizontal and two vertical long-distance busses. The behavior of a MUXTREE molecule, described in detail elsewhere [6](pp. 135-143), is completely defined by a molecular code organized as a 20-bit data $MOLCODE19:0$, itself stored in a *configuration register* CREG.

The *multimolecular organization* divides finally the artificial cell, our up-down counter, into a finite number of molecules (six), where each molecule is defined by a unique configuration, its molecular code $MOLCODE19:0$. For clarity's sake, each $MOLCODE$ is represented in Figure 1b by five hexadecimal characters.

Let us call *ribosomic genome* (RG) the string of all the molecular codes of our artificial cell (Figure 1b), where each molecular code is a 20-bit or 5-hexadecimal characters word $MOLCODE19:0$:

$$RG = \sum MOLCODE19:0 = 10001, 23040, 23306, 01C50, 24000, 66302 \qquad (1)$$

In conformance with the definitions of the companion paper [5](Subsection 2.4), the ribosomic genome RG represents the genotype of our cell. It is directly interpreted by the FPGA, that is, by the MUXTREE molecules that act as the equivalent of the ribotype. The phenotype describes the operation of the complete cell. Relation (4) of [5] becomes, in the general case of a cell:

$$\sum MOLECULE[RG] = CELL \qquad (2)$$

and, in the particular case of our reversible counter:

$$\sum MUXTREE[RG] = \sum MUXTREE[10001, 23040, 23306, 01C50, 24000, 66302]$$
$$= up{-}down\ counter$$

1.3 Molecule's self-test and cell's self-repair

We have already described the capability of the complete organism to self-repair by sacrificing a complete column of each faulty cell [5](Subsection 3.2). This operation is analogous to the cicatrization process of living beings, the scar being represented by the column sacrificed by the artificial organism. If the cell is complex, this process is costly: it is thus indispensable to dispose of a second self-repair mechanism, situated at the molecular level. The biological inspiration is again immediate: the DNA's double helix offers a complete redundancy of the genomic information and allows the rectification of any base in an helix by comparison with the complementary base in the opposing helix. The specifications of the molecular self-repair system must include the following features:

- it must operate in real time;
- it must preserve the memorized values, that is, the state of the D-type flip-flop contained in each molecule;
- it must assure the automatic detection of a fault (self-test), its localization, and its repair (self-repair);
- it must involve an acceptable overhead;
- finally, in case of multiple faults (to many faulty molecules), it must generate a global signal $KILL = 1$ which activates the suppression of the cell and starts the self-repair process of the complete organism (see Subsection 3.2 of [5]).

All these constraints forced us to adopt a set of compromises with regard to the fault-detection capabilities of the system. A self-repairing MUXTREE molecule (or MUXTREE SR) can be divided into three parts (Figure 2) [12], [6](pp.249-258):

- the functional part of the molecule (the multiplexer and the internal flip-flop) is tested by space redundancy: the logic is duplicated (M1 and M2) and the outputs of the two elements compared to detect a fault; a third copy of the flip-flop was added to allow self-repair (i.e., to save the flip-flop state);
- the configuration register (CREG) is tested as the configuration is being entered (and thus not on-line); being implemented as a shift register, it can be tested using a special test sequence introduced in all elements in parallel before the actual configuration for the system;
- faults on the connections (and in the switch block SB) can be detected, but cannot be repaired, both because they cannot be localized to a particular connection, and because our self-repair system exploits the connections to

reconfigure the array; in the current system, therefore, we decided not to test the connections directly; this assumption is in accordance with the present state of the art [7].

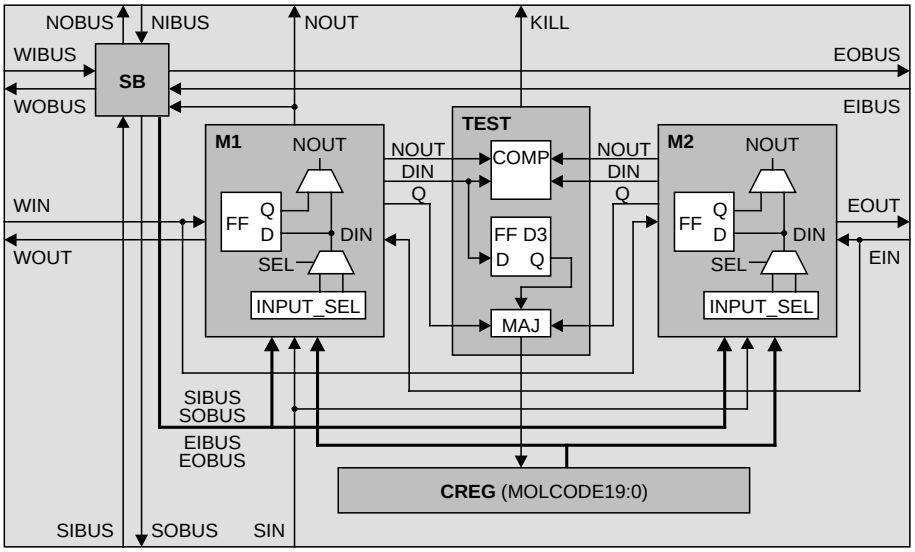


Fig. 2. A self-testing MUXTREE SR molecule using space redundancy.

The self-repair system had to meet the same constraints as those of the self-test system, and in particular the requirement that its additional logic be minimized. Exploiting the fact that the spare columns are distributed and that their frequency is programmable (Subsection 1.4), we limited the reconfiguration of the array to a single molecule per line between two spare columns (Figure 3). This allows us to minimize the amount of logic required for the reconfiguration of the array, while keeping a more than acceptable level of robustness. This mechanism is also in accordance with the present state of the art [9].

It should be added that if the self-repair capabilities of the MUXTREE SR molecular level is exceeded, a global *KILL* signal is generated and the system will attempt to reconfigure at the higher (cellular) level, as mentioned in Subsection 3.2 of [5].

1.4 Cell's self-replication (cellular division): space divider

The macroscopic mechanism of cellular division (Subsection 2.3 and Figure 4 of [5]) implies the construction, from a unique mother cell, of a certain number of daughter cells identical to the mother cell. This construction represents a second

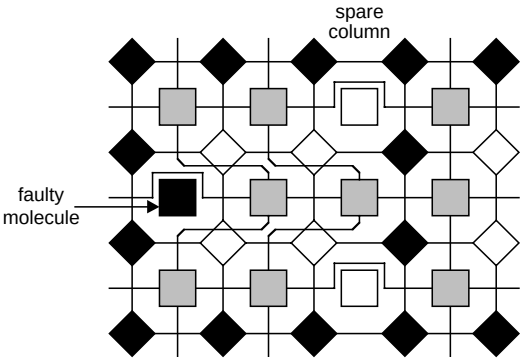


Fig. 3. The self-repair mechanism for an array of MUXTREE SR molecules.

example of self-replication (after that of the organism described in Subsection 3.1 of [5]). Performed by the molecular tissue MUXTREE SR, this self-replication will occur in two steps:

- first, a frontier will define, in the MUXTREE SR tissue, the cell's width, its height, and the number of spare columns;
- then, each molecule in each cell will receive its configuration, that is, its molecular code $MOLCODE$ (the ribosomic genome RG).

The mechanism we have adopted to implement this process is to introduce a very simple *molecular automaton* or *space divider* (Figure 4), capable of creating a set of *boundaries* which partition the FPGA into blocks of molecules, each defining a cell. It then becomes possible to configure the entire array by entering the configuration of a single cell, the ribosomic genome RG, which will automatically be replicated within all the boundaries (cellular division). This molecular automaton is roughly inspired by the self-replicating Langton's loop [4]; its design is described elsewhere [6](pp. 240-249), [10].

The microscopic self-replication of a cell can be described by an L-system [8], [2], [3], [11], in the same manner as the macroscopic cellular division of the organism [5](Subsection 2.3). Our space divider is a two-dimensional molecular automaton and its structuring process is essentially a growth process, starting from the automaton molecule on the lower left-hand side, where the programming data are fed continuously. The representation of the symboles used in the developmental model is given in Figure 5a. The L-system description of the space divider starts with a single letter axiom which corresponds to a left branching apex. The productions applied to the axiom in order to obtain a cellular structure of 3×3 molecules are listed in Figure 5b; they fall into three categories:

- the branching signal propagation productions p1 to p4;
- the simple growth productions p5 to p14;

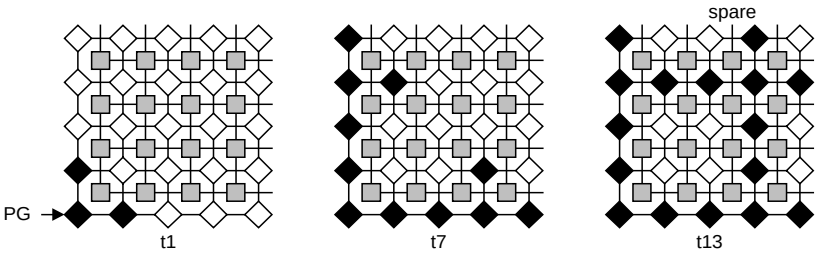


Fig. 4. Space divider: molecular automaton colonizing an array of MUXTREE SR molecules (each molecule of the space divider is a diamond; each MUXTREE SR molecule is a gray square); PG: polymerase genome.

– the branching growth productions p15 to p18.

Thirteen derivation steps of the developmental process are shown in Figure 5c, where the first character of the string to the left of the vertical separator is part of the program that is fed to the space divider: it is applied from the outside and corresponds to the left context of the first character after the vertical separator.

The molecular automaton finally implemented (Figure 5d) defines a division of the cellular space into squares of size 3×3 molecules (Figure 4 where each diamond corresponds to a square in Figure 5d).

A very interesting "bonus" of this system is that it becomes possible to use this automaton to define which columns of the array will be spare columns, used for self-repair (Figure 3). The frequency of these columns, and consequently the robustness of the system, is therefore entirely programmable, rather than hardwired, and can thus be set to meet the requirements of a single application.

Coming back to our original example of a modulo-4 up-down counter (Figure 1b), it should be obvious that the colonizing process of Figure 5d will generate a MUXTREE SR array able to implement our design (the six molecules of the counter) with a spare column to the right (three spare molecules).

The programming data of the space divider are equivalent to the *polymerase genome* (PG) of a living being, as they make possible the cellular division of the organism. Figures 5c and 5d detail the polymerase genome, which is described by a short cycle iib:

$$PG = iib, iib, iib, ... \tag{3}$$

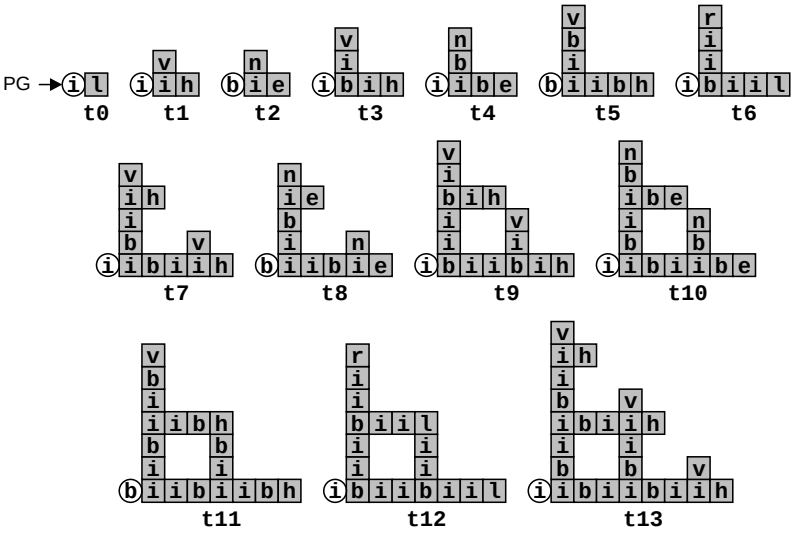
Let us call TISSUE the FPGA consisting of MUXTREE SR molecules, where each molecule includes a copy of the space divider. TISSUE represents then the lowest level hardware primitive in our hierarchy (it cannot be further decomposed). Coming back to the definitions of Subsection 2.4 of [5], the polymerase genome PG represents the genotype of the TISSUE FPGA, itself equivalent to the ribotype. The phenotype is the molecular tissue $\sum MOLECULE$, ready to

```
                          p1:   b<i   -> b      p10:  i<v   -> n
h: horizontal apex        p2:  b(<i   -> b      p11: i(<v   -> n
v: vertical apex          p3:  b[<i   -> b      p12:  i<n   -> iv
e: east growing apex      p4:     b   -> i      p13: i(<n   -> iv
n: north growing apex     p5:   i<h   -> e      p14:  b<n   -> bv
l: left branching apex    p6:  i[<h   -> e      p15:  b<h   -> l
r: right branching apex   p7:   i<e   -> ih     p16:    l   -> i(v)h
i: internode              p8:  i[<e   -> ih     p17:  b<v   -> r
b: branching signal       p9:   b<e   -> bh     p18:    r   -> i[h]v
```

(a) (b)

```
t0:  i|l
t1:  i|i(v)h
t2:  b|i(n)e
t3:  i|b(iv)ih
t4:  i|i(bn)be
t5:  b|i(ibv)ibh
t6:  i|b(iir)iil
t7:  i|i(bii[h]v)bii(v)h
t8:  b|i(ibi[e]n)ibi(n)e
t9:  i|b(iib[ih]iv)iib(iv)ih
t10: i|i(bii[be]bn)bii(bn)be
t11: b|i(ibi[ibh]ibv)ibi(ibv)ibh
t12: i|b(iib[iil]iir)iib(iir)iil
t13: i|i(bii[bii(v)h]bii[h]v)bii(bii[h]v)bii(v)h
     ↓
     PG
```

(c)



(d)

Fig. 5. L-system model of the space divider. (a) The symbol representation. (b) The production set. (c) A sample derivation; *PG*: polymerase genome. (d) The final molecular automaton realizing a space division.

receive its ribosomic genome. Relation (4) of [5] thus becomes, in the general case:

$$TISSUE[PG] = \sum MOLECULE \tag{4}$$

and, in the particular case of the space divider in our MUXTREE SR molecules:

$$TISSUE[iib, iib, iib, ...] = \sum MUXTREE \tag{5}$$

Finally, we will note that the process of molecular development of Figure 5d (the self-replication of the cell) represents a microscopic (that is, molecular level) description of the macroscopic process of cellular division of the organism [5](Figure 4). The construction of a daughter cell requires a total time t (Figure 4 of [5]), decomposed into 12 steps at the molecular level (Figure 5d).

2 Conclusion

2.1 The missing link

In Sections 2 and 3 of the companion paper [5], the macroscopic characteristics and properties of Embryonics have been illustrated by means of an artificial multicellular organism, a StopWatch, made up of four cells (without spares), each a small processor (a binary decision machine) with the associated memory necessary to store the operative genome. In Section 1 of this paper, the microscopic architecture and features of Embryonics' cells have been described through another, far simpler example: an up-down counter comprising six MUXTREE SR molecules (without spares).

The design and the implementation of a specimen of the StopWatch's cell into a regular array of MUXTREE SR molecules is a complex task which is beyond the scope of this paper; as an example, the realization of the basic cell of an even simpler multicellular organism, a modulo-60 counter with only two cells, requires 600 MUXTREE SR molecules (without spares) for both the binary decision machine and the associated memory (a 360-bit shift register) [6](pp. 258-265).

2.2 Design methodology and genome hierarchy

In our Embryonics project, the design of a multicellular automaton requires the following steps:

- the original specifications are mapped into a homogeneous array of cells (binary decision machines with their associated memory); the software (a microprogram) and the hardware (the architecture of the cell) are tailored according to the specific example (Turing machine, electronic watch, random number generator, etc.); in biological terms, this microprogram represents the *operative genome* (OG);

- the cell's hardware is implemented into a homogeneous array of molecules, the MUXTREE SR molecules; spare columns are introduced in order to improve the global reliability; our artificial cell being analogous to the ribosome of a natural cell, the string of the molecule codes can be seen as the *ribosomic genome* (RG);
- the dimensions of the final molecular array, as well as the frequency of the spare columns, define the string of data required by the molecular automaton (the space divider that creates the boundaries between cells); as this information will allow the generation of all the daughter cells starting from the first mother cell, it can be considered as equivalent to the *polymerase genome* (PG).

Given the basic TISSUE FPGA (i.e., the array of MUXTREE SR molecules, with a space divider automaton in each molecule), the corresponding programming has to take place in reverse order:

- the polymerase genome (PG) is injected in order to obtain the boundaries between cells:
$$TISSUE[PG] = \sum MOLECULE \tag{6}$$
- The ribosomic genome (RG) is injected in order to configure the array of MUXTREE SR molecules and obtain the final architecture of each cell:
$$\sum MOLECULE[RG] = CELL \tag{7}$$
- The operative genome (OG) is stored into the memory of each cell in order to make the cell ready to execute the specifications of the whole organism ORG:
$$\sum CELL[OG] = ORG \tag{8}$$

By replacing $CELL$ in expression (8) with the values derived from (2) and (4), we can finally show the sequence of three configurations that transforms the primitive FPGA TISSUE into an operative multicellular organism:

$$\sum((TISSUE[PG])[RG])[OG] = ORG \tag{9}$$

Echoing biology, we have faced complexity by decomposing the organism into cells and then the cells into molecules. This decomposition implies multiple configuration steps: the polymerase genome organizes the space by defining the cells' boundaries, the ribosomic genome defines the architecture of each cell as an array of molecules, and finally the operative genome makes up the program which will be executed by the cellular processors to accomplish the required task. The Latin motto "divide and conquer" maintains its relevance even today.

Acknowledgments

This work was supported by grants 20-42'270.94 and 20-49'375.96 from the Swiss National Science Foundation to which we express our gratitude.

References

1. R. E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.
2. H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.
3. H. Kitano. Morphogenesis for evolvable systems. In E. Sanchez and M. Tomassini, editors, *Towards Evolvable Hardware*, volume 1062 of *Lecture Notes in Computer Science*, pages 99–117. Springer-Verlag, Heidelberg, 1996.
4. C. G. Langton. Self-reproduction in cellular automata. *Physica D*, 10:135–144, 1984.
5. D. Mange, A. Stauffer, and G. Tempesti. Embryonics: A macroscopic view of the cellular architecture. In M. Sipper, D. Mange, and A. Perez, editors, *Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98)*, Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, 1998.
6. D. Mange and M. Tomassini, editors. *Bio-Inspired Computing Machines*. Presses polytechniques et universitaires romandes, Lausanne, 1998.
7. R. Negrini, M. G. Sami, and R. Stefanelli. *Fault Tolerance Through Reconfiguration in VLSI and WSI Arrays*. The MIT Press, Cambridge, 1989.
8. P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag, New York, 1990.
9. A. Shibayama, H. Igura, M. Mizuno, and M. Yamashina. An autonomous reconfigurable cell array for fault-tolerant LSIs. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pages 230–231 and 462, February 1997.
10. A. Stauffer and M. Sipper. L-hardware: Modeling and implementing cellular development using L-systems. In D. Mange and M. Tomassini, editors, *Bio-Inspired Computing Machines*. Presses polytechniques et universitaires romandes, Lausanne, 1998.
11. A. Stauffer and M. Sipper. Modeling cellular development using L-systems. In M. Sipper, D. Mange, and A. Perez, editors, *Proceedings of The Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98)*, Lecture Notes in Computer Science. Springer-Verlag, Heidelberg, 1998.
12. G. Tempesti, D. Mange, and A. Stauffer. A robust multiplexer-based FPGA inspired by biological systems. *Journal of Systems Architecture*, 43(10):719–733, 1997.

This article was processed using the LaTeX macro package with LLNCS style