

# Toward Self-Repairing and Self-Replicating Hardware: The Embryonics Approach

Daniel Mange, Moshe Sipper, André Stauffer, and Gianluca Tempesti  
Logic Systems Laboratory, Swiss Federal Institute of Technology  
CH-1015 Lausanne, Switzerland  
daniel.mange@epfl.ch

## Abstract

*The growth and operation of all living beings are directed by the interpretation, in each of their cells, of a chemical program, the DNA string or genome. This process is the source of inspiration for the Embryonics (embryonic electronics) project, whose final objective is the design of highly robust integrated circuits, endowed with properties usually associated with the living world: self-repair (cicatrization) and self-replication. The Embryonics architecture is based on four hierarchical levels of organization: 1) The basic primitive of our system is the molecule, a multiplexer-based element of a novel programmable circuit. 2) A finite set of molecules make up a cell, essentially a small processor with associated memory. 3) A finite set of cells make up an organism, an application-specific multiprocessor system. The organism can itself replicate, giving rise to a population of identical organisms. In the conclusion, we describe our ongoing research efforts to meet three challenges: a scientific challenge, that of implementing the original specifications formulated by John von Neumann; a technical challenge, that of realizing very robust integrated circuits; and a biological challenge, that of attempting to show that the genomes of artificial and natural organisms share common properties.*

## 1 Introduction<sup>1</sup>

### 1.1 Embryonics = embryonic electronics

A human being consists of approximately 60 trillion ( $60 \times 10^{12}$ ) cells. At each instant, in each of these 60 trillion cells, the *genome*, a ribbon of 2 billion characters, is decoded to produce the proteins needed for the survival of

the organism. This genome contains the ensemble of the genetic inheritance of the individual and, at the same time, the instructions for both the construction and the operation of the organism. The parallel execution of 60 trillion genomes in as many cells occurs ceaselessly from the conception to the death of the individual. Faults are rare and, in the majority of cases, successfully detected and repaired. This process is remarkable for its complexity and its precision. Moreover, it relies on completely discrete information: the structure of DNA (the chemical substrate of the genome) is a sequence of four bases, usually designated with the letters A (adenine), C (cytosine), G (guanine), and T (thymine).

Our *Embryonics* project (for *embryonic electronics*) is inspired by the basic processes of molecular biology and by the embryonic development of living beings [1][27]. By adopting certain features of cellular organization, and by transposing them to the two-dimensional world of integrated circuits on silicon, we will show that properties unique to the living world, such as *self-replication* and *self-repair*, can also be applied to artificial objects (integrated circuits).

### 1.2 Objectives

Our final objective is the development of very large scale integrated (VLSI) circuits capable of self-repair and self-replication. Self-repair allows partial reconstruction in case of a minor fault, while self-replication allows complete reconstruction of the original device in case of a major fault. These two properties are particularly desirable for complex artificial systems requiring improved reliability in short-, medium-, or long-term applications.

#### 1. Short-term applications [2]:

- applications that require very high levels of reliability, such as avionics or medical electronics;
- applications designed for hostile environments, such as space, where the increased radiation levels reduce the reliability of components;
- applications which exploit the latest technological advances, and notably the drastic device shrinking, low power supply levels, and increasing operating

<sup>1</sup> Based on the following paper: D. Mange, M. Sipper, A. Stauffer, G. Tempesti. "Toward Robust Integrated Circuits: The Embryonics Approach". *Proceedings of the IEEE*, Vol. 88, No. 4, April 2000, to appear.

speeds, which accompany the technological evolution to deeper submicrometer levels and significantly reduce the noise margins and increase the soft-error rates [17].

2. Medium-term applications, where our aim is to develop very complex integrated circuits capable of online self-repair, dispensing with the systematic detection of faults at fabrication, impossible for systems consisting of millions of logic gates [18].
3. Long-term applications, executed on systems built with imperfect components: this is von Neumann's historical idea [4], the basis of all present projects aimed at the realization of complex integrated circuits at the atomic scale (nanotechnologies) [19][20][21][22].

Self-replication, or "cloning", can be justified independently of self-repair:

- to replicate, within a field-programmable gate array (FPGA), functionally equivalent systems [23];
- to produce the massive quantity of future integrated circuits, implemented using nanotechnologies [14];
- to finally accomplish John von Neumann's unachieved dream, that is, the realization of a self-replicating automaton endowed with the properties of universal computation and construction [4].

These emerging needs require the development of a new design paradigm that supports efficient online VLSI testing and self-repair solutions. Inspired by the architecture of living beings, we will show how to implement online testing, self-repair, and self-replication using both hardware and software redundancy. The programmable degree of robustness of our systems, a function of an overhead itself programmable, is one of the major original features of the Embryonics project.

## 2 A bird's eye view of Embryonics

### 2.1 Biological inspiration

The majority of living beings, with the exception of unicellular organisms such as viruses and bacteria, share three fundamental features.

1. *Multicellular organization* divides the organism into a finite number of *cells*, each realizing a unique function (neuron, muscle, intestine, etc.). The same organism can contain multiple cells of the same kind.
2. *Cellular division* is the process whereby each cell (beginning with the first cell or *zygote*) generates one or two daughter cells. During this division, all of the genetic material of the mother cell, the *genome*, is copied into the daughter cell(s).
3. *Cellular differentiation* defines the role of each cell of the organism, that is, its particular function (neuron, muscle, intestine, etc.). This specialization of the cell

is obtained through the expression of part of the genome, consisting of one or more *genes*, and depends essentially on the physical position of the cell in the organism.

A consequence of these three features is that each cell is "universal", since it contains the whole of the organism's genetic material, the genome. Should a minor (wound) or major (loss of an organ) trauma occur, living organisms are thus potentially capable of self-repair (cicatriziation) or self-replication (cloning or budding) [1].

The two properties of self-repair and self-replication based on a multicellular tissue are unique to the living world. The main goal of the Embryonics project is the implementation of the above three features of living organisms in an integrated circuit in silicon, in order to obtain the properties of self-repair and self-replication.

Our artificial organism will ultimately be divided into cells, themselves decomposed into molecules, a structure which determines the plan of this section: Subsection 2.2 describes the three fundamental features of the organism (multicellular organization, cellular differentiation, and cellular division), while in Subsection 2.3 we demonstrate that the organism, thanks to these three features, exhibits the two sought properties (self-replication and self-repair). Subsection 2.4 describes the cells and presents their essential features (multimolecular organization, molecular configuration, and molecular error detection), while Subsection 2.5 shows that the two sought properties (self-replication and self-repair) apply both at the cellular level as well as at the organismic level. Subsection 2.6 underscores the four organizational levels of the hierarchy of the Embryonics project, which are, from the bottom up, the molecule, the cell, the organism, and finally the population of organisms.

### 2.2 The organism's features: multicellular organization, cellular differentiation, and cellular division

The environment in which our quasi-biological development occurs is imposed by the structure of electronic circuits, and consists of a finite (but arbitrarily large) two-dimensional surface of silicon. This surface is divided into rows and columns, whose intersections define the cells. Since such cells (small processors and their memory) have an identical physical structure (i.e., an identical set of logic operators and of connections), the cellular array is homogeneous. As the program in each cell (our artificial genome) is identical, only the state of a cell (i.e., the contents of its registers) can differentiate it from its neighbors.

In this section, we first show how to implement in our artificial organisms the three fundamental features of

multicellular organization, cellular differentiation, and cellular division, by using a generic and abstract six-cell example. *Multicellular organization* divides the artificial organism (ORG) into a finite number of cells (Fig. 1). Each cell (CELL) realizes a unique function, defined by a sub-program called the *gene* of the cell and selected as a function of the values of both the horizontal (X) and the vertical (Y) coordinates (in Fig. 1, the genes are labeled A to F for coordinates  $X, Y=1, 1$  to  $X, Y=3, 2$ ). Our final artificial genome will be divided into three main parts: the *operative genome* (OG), the *ribosomic genome* (RG), and the *polymerase genome* (PG). Let us call operative genome (OG) a program containing all the genes of an artificial organism, where each gene (A to F) is a sub-program characterized by a set of instructions and by the cell's position (coordinates  $X, Y=1, 1$  to  $X, Y=3, 2$ ). Fig. 1 is then a graphical representation of organism ORG's operative genome.

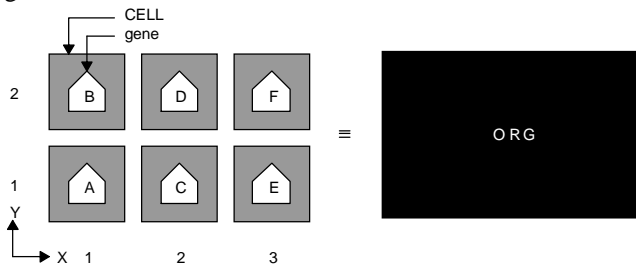


Fig. 1. Multicellular organization of a 6-cell organism ORG.

Let then each cell contain the entire operative genome OG (Fig. 2a): depending on its position in the array, i.e., its place within the organism, each cell can then interpret the operative genome and extract and execute the gene which defines its function. In summary, storing the whole operative genome in each cell makes the cell universal: given the proper coordinates, it can execute any one of the genes of the operative genome and thus implement *cellular differentiation*. In our artificial organism, any cell  $CELL[X, Y]$  continuously computes its coordinate X by incrementing the coordinate WX of its neighbor immediately to the west (Fig. 2b). Likewise, it continuously computes its coordinate Y by incrementing the coordinate SY of its neighbor immediately to the south. Taking into consideration these computations, Fig. 3 shows the final operative genome OG of the organism ORG.

At startup, the first cell or *zygote* (Fig. 4), arbitrarily defined as having the coordinates  $X, Y=1, 1$ , holds the one and only copy of the operative genome OG. After time  $t_1$ , the genome of the zygote (*mother* cell) is copied into the neighboring (*daughter*) cells to the east ( $CELL[2, 1]$ ) and to the north ( $CELL[1, 2]$ ). This process of *cellular division* continues until the six cells of the organism ORG are completely programmed (in our example, the farthest cell is programmed after time  $t_3$ ).

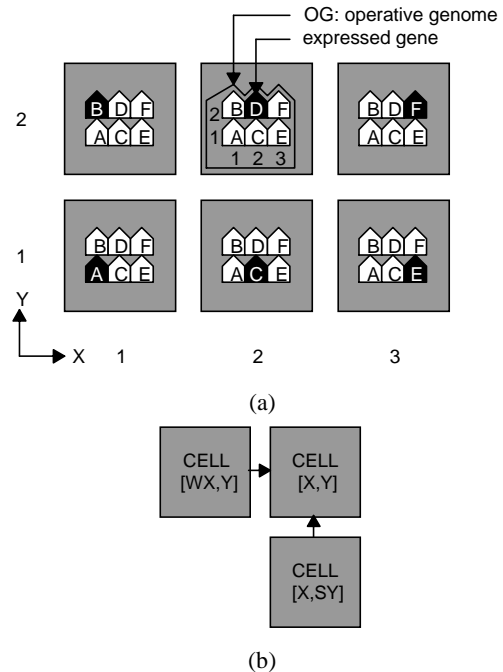


Fig. 2. Cellular differentiation. (a) Global organization. (b) A cell  $CELL[X, Y]$  with its west neighbor  $CELL[WX, Y]$  and its south neighbor  $CELL[X, SY]$ ;  $X=WX+1$ ;  $Y=SY+1$ .

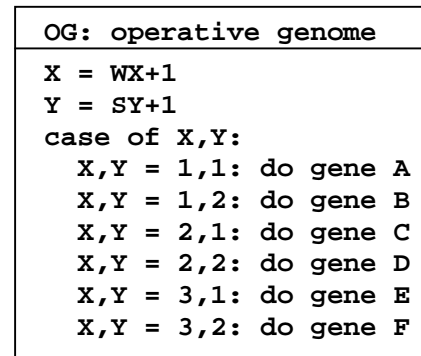


Fig. 3. The operative genome OG of the organism ORG.

### 2.3 The organism's properties: organismic self-replication and organismic self-repair

The *self-replication* or *cloning of the organism*, i.e., the production of an exact copy of the original, rests on two assumptions.

1. There exists a sufficient number of spare cells in the array (at least six in the example of Fig. 5) to contain the additional organism.
2. The calculation of the coordinates produces a cycle ( $X=1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$  and  $Y=1 \rightarrow 2 \rightarrow 1 \dots$  in Fig. 5, implying  $X=(WX+1) \text{ modulo } 3$  and  $Y=(SY+1) \text{ modulo } 2$ ).

As the same pattern of coordinates produces the same pattern of genes, self-replication can be easily

accomplished if the program of the operative genome OG, associated with the homogeneous array of cells, produces several occurrences of the basic pattern of coordinates. In our example (Fig. 5), the repetition of the vertical coordinate pattern ( $Y=1 \rightarrow 2 \rightarrow 1 \rightarrow 2$ ) in a sufficiently large array of cells produces one copy, the *daughter organism*, of the original *mother organism*. Given a sufficiently large space, the self-replication process can be repeated for any number of specimens in the X and/or the Y axes.

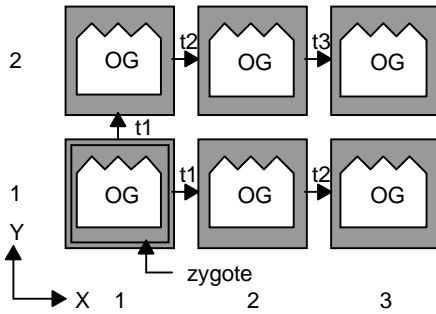


Fig. 4. Cellular division; OG: operative genome;  $t1 \dots t3$ : three cellular divisions.

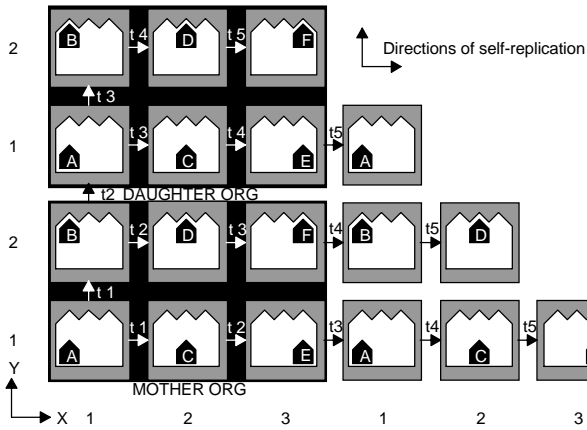


Fig. 5. Self-replication of a 6-cell organism ORG in a limited homogeneous array of 6x4 cells (situation at time  $t5$  after 5 cellular divisions); MOTHER ORG = mother organism; DAUGHTER ORG = daughter organism.

In order to implement the *self-repair of the organism*, we decided to use spare cells to the right of the original organism (Fig. 6). The existence of a fault is detected by a KILL signal which is calculated in each cell by a built-in self-test mechanism realized at the molecular level (see Subsection 2.5 below). The state  $KILL=1$  identifies the faulty cell, and the entire column to which the faulty cell belongs is considered faulty, and is deactivated (column  $X=2$  in Fig. 6). All the functions (X coordinate and gene) of the cells to the right of the column  $X=1$  are shifted by one column to the right. Obviously, this process requires as many spare columns to the right of the array as there are faulty cells or columns to repair (two spare columns,

tolerating two successive faulty cells, in the example of Fig. 6). It also implies that the cell needs to be able to bypass the faulty column and to divert to the right all the required signals (such as the operative genome and the X coordinate, as well as the data busses).

Given a sufficient number of cells, it is obviously possible to combine self-repair in the X direction, and self-replication in both the X and Y directions.

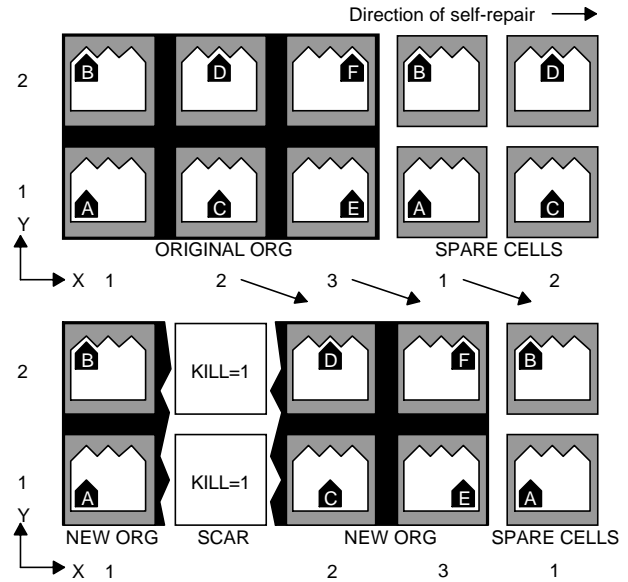


Fig. 6. Organismic self-repair.

## 2.4 The cell's features: multimolecular organization, molecular configuration, and molecular fault detection

In each cell of every living being, the genome is translated sequentially by a chemical processor, the *ribosome*, to create the proteins needed for the organism's survival. The ribosome itself consists of molecules, whose description is an important part of the genome.

As mentioned, in the Embryonics project each cell is a small processor, sequentially executing the instructions of a first part of the artificial genome, the operative genome OG. The need to realize organisms of varying degrees of complexity has led us to design an artificial cell characterized by a flexible architecture, that is, itself configurable. It will therefore be implemented using a new kind of fine-grained, field-programmable gate array (FPGA).

Each element of this FPGA (consisting essentially of a multiplexer associated with a programmable connection network) is then equivalent to a *molecule*, and an appropriate number of these artificial molecules allows us to realize application-specific processors. We will call *multimolecular organization* the use of many molecules to

realize one cell. The configuration string of the FPGA (that is, the information required to assign the logic function of each molecule) constitutes the second part of our artificial genome: the *ribosomic genome* RG. Fig. 7a shows a generic and abstract example of an extremely simple cell (CELL) consisting of six molecules, each defined by a *molecular code* or MOLCODE (a to f). The set of these six MOLCODES constitutes the ribosomic genome RG of the cell.

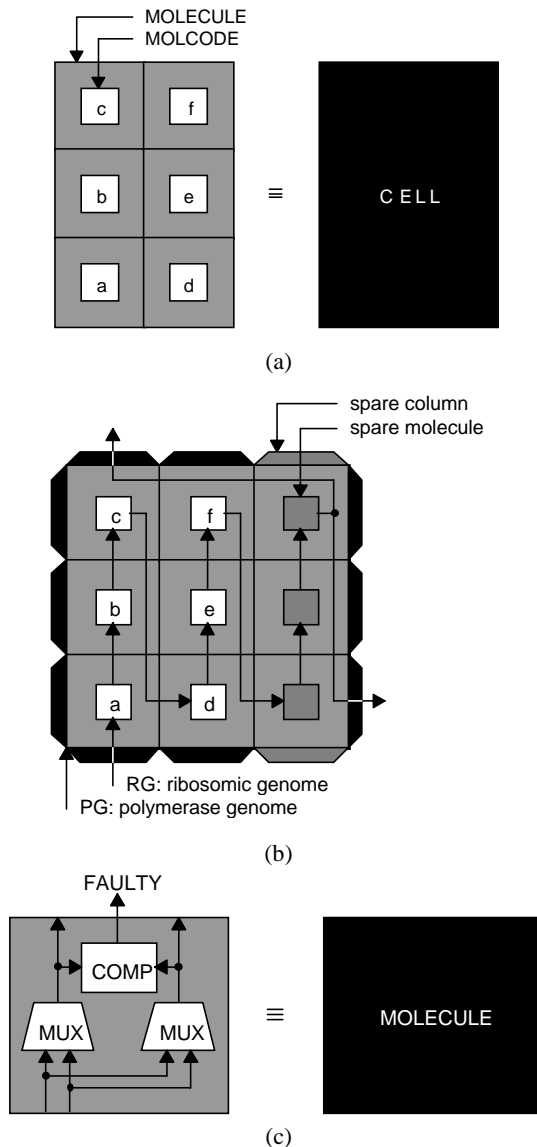


Fig. 7. The cell's features. (a) Multimolecular organization; RG: ribosomic genome: a, b, c, d, e, f. (b) Molecular configuration; PG: polymerase genome: height x width = 3x3; 001 = spare column. (c) Molecular fault detection; MUX: multiplexer; COMP: comparator.

The information contained in the ribosomic genome RG thus defines the logic function of each molecule by assigning a molecular code MOLCODE to it. To obtain a

functional cell, we require two additional pieces of information:

- the *physical position* of each molecule in the cellular space;
- the presence of one or more *spare columns*, composed of *spare molecules*, required for the self-repair described below (Subsection 2.5).

The definition of these pieces of information is the *molecular configuration* (Fig. 7b). Their injection into the FPGA will allow:

- the creation of a border surrounding the molecules of a given cell;
- the insertion of one or more spare columns;
- the definition of the connections between the molecules, required for the propagation of the ribosomic genome RG.

The information needed for the molecular configuration (essentially, the height and width of the cell in number of molecules and the position of the spare columns) makes up the third and last part of our artificial genome: the *polymerase genome* PG (a terminology which will be justified in Subsection 2.5).

Last, it is imperative to be able to automatically detect the presence of faults at the molecular level and to relay this information to the cellular level. Moreover, if we consider that the death of a column of cells is quite expensive in terms of wasted resources, the ability to repair at least some of these faults at the molecular level (that is, without invoking the organismic self-repair mechanism) becomes highly desirable. The biological inspiration for this process derives from the DNA's double helix, the physical support of natural genomes, which provides complete redundancy of the genomic information through the presence of complementary bases in the opposing branches of the helix. By duplicating the material of each molecule (essentially the multiplexer MUX) and by continuously comparing the signals produced by each of the two copies (Fig. 7c), it is possible to detect a faulty molecule and to generate a signal FAULTY=1, realizing the *molecular fault detection* which will make possible cellular self-repair (described below in Subsection 2.5).

## 2.5 The cell's properties: cellular self-replication and cellular self-repair

A consequence of the multimolecular organization and of the molecular configuration of the FPGA (Subsection 2.4 and Fig. 7b) is the ability, for any given cell, to propagate its polymerase genome PG and its ribosomic genome RG in order to automatically configure two daughter cells, architecturally identical to the mother cell, to the east and to the north (Fig. 8), thus implementing *cellular self-replication*.

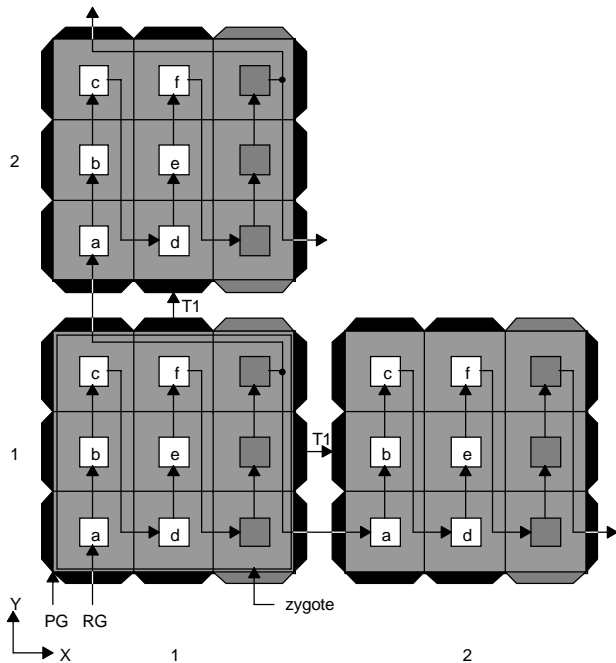


Fig. 8. Cellular self-replication; T1: one instance of cellular self-replication; RG: ribosomic genome; PG: polymerase genome.

Cellular self-replication is a prerequisite for cellular division at the organismic level described above (Subsection 2.2 and Fig. 4), during which the operative genome is copied from the mother cell into the daughter cells. In living systems, a specific molecule, the *polymerase enzyme*, allows cellular replication through the duplication of the genome. It is by analogy to this enzyme that the third part of our artificial genome is called polymerase genome.

The presence of spare columns, defined by the molecular configuration, and the automatic detection of faulty molecules (Subsection 2.4, Fig. 7b and 7c) allow *cellular self-repair*: each faulty molecule is deactivated, isolated from the network, and replaced by a neighboring molecule, which will itself be replaced by a neighbor, and so on until a spare molecule (SM) is reached (Fig. 9a).

The number of faulty molecules handled by the molecular self-repair mechanism is necessarily limited: in the example of Fig. 9a, we tolerate at most one faulty molecule per row. If more than one molecule is faulty in one or more rows (Fig. 9b), molecular self-repair is impossible, in which case a global signal  $KILL=1$  is generated to activate the organismic self-repair described above (Subsection 2.3 and Fig. 6).

## 2.6 The Embryonics landscape

The final architecture of the Embryonics project is based on four hierarchical levels of organization which, described from the bottom up, are the following (Fig. 10).

1. The basic primitive of our system is the *molecule*, the element of our new FPGA, consisting essentially of a multiplexer associated with a programmable connection network. The multiplexer is duplicated to allow the detection of faults. The logic function of each molecule is defined by its molecular code or MOLCODE.
2. A finite set of molecules makes up a *cell*, essentially a processor with the associated memory. In a first programming step of the FPGA, the polymerase genome PG defines the topology of the cell, that is, its width, height, and the presence and positions of columns of spare molecules. In a second step, the ribosomic genome RG defines the logic function of each molecule by assigning its molecular code or MOLCODE.
3. A finite set of cells makes up an *organism*, an application-specific multiprocessor system. In a third and last programming step, the operative genome OG is copied into the memory of each cell to define the particular application executed by the organism (electronic watch, random number generator, and a Turing machine being examples shown by us to date) [29].

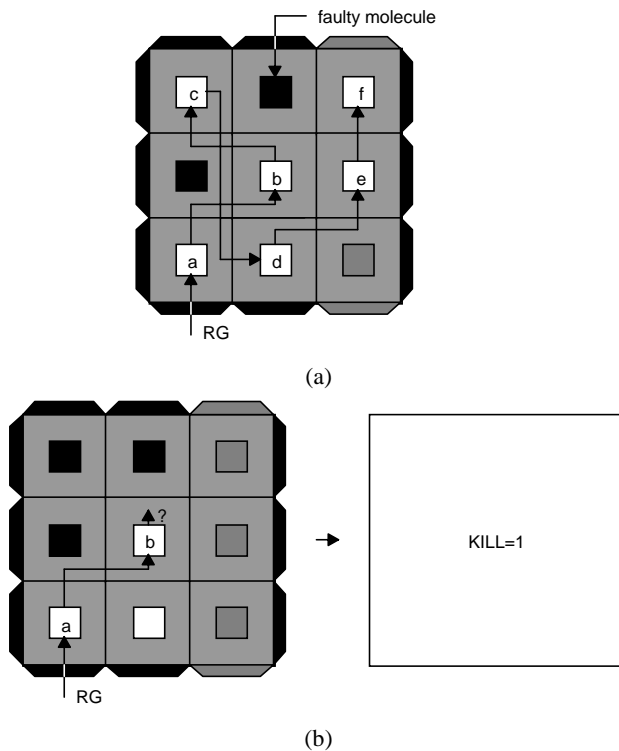


Fig. 9. Cellular self-repair. (a) Possible self-repair (at most one faulty molecule per row). (b) Impossible self-repair (more than one faulty molecule per row):  $KILL=1$  (self-repair at the organismic level).

4. The organism can itself self-replicate, giving rise to a *population* of identical organisms, the highest level of our hierarchy.

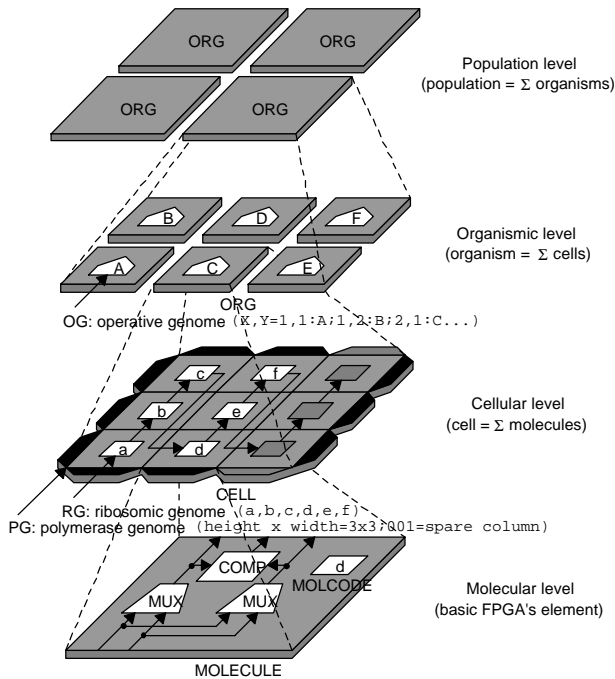


Fig. 10. The Embryonics landscape: a 4-level hierarchy.

### 3 Conclusion

#### 3.1 A scientific challenge: von Neumann revisited

The early history of the theory of self-replicating machines is basically the history of John von Neumann's thinking on the matter [4][8]. Von Neumann's automaton is a homogeneous two-dimensional array of elements, each element being a finite state machine with 29 states. In his historic work, von Neumann showed that a possible configuration (a set of elements in a given state) of his automaton can implement a *universal constructor* able to build onto the array any computing machine described in a dedicated part of the universal constructor, the *tape*. Self-replication is then a special case of construction, occurring when the universal constructor itself is described on the tape. Moreover, von Neumann demonstrated that his automaton is endowed with two major properties: *construction universality*, the capability of describing on the tape and building onto the array a machine of any dimension, and *computation universality*, the capability of describing and building a universal Turing machine.

One should be reminded that, in biology, the *cell* is the smallest part of the living being containing the complete blueprint of the being, the genome. On the basis of this definition, von Neumann's automaton can be considered as

a *unicellular organism*, since it contains a single copy of the genome, i.e., the description stored on the tape. Each element of the automaton is thus a part of the cell, i.e., a *molecule*. Von Neumann's automaton, therefore, is a *molecular automaton*, and self-replication is a very complex process due to the interactions of hundreds of thousands of molecules.

Arbib [9] was the first to suggest a truly "cellular" automaton, in which every cell contains a complete copy of the genome, and a hierarchical organization, where each cell is itself composed of smaller regular elements, the "molecules". The Embryonics project is therefore the first actual implementation of Arbib's concept, as each of its elements contains a copy of the genome. Each element of our automaton is thus a cell in the biological sense, and our automaton is truly a *multicellular automaton*.

The verification of the property of *universal computation*, that is, the design of a universal Turing machine on our multicellular array, is one of the major ongoing projects in our laboratory [28]. The property of *universal construction* raises issues of a different nature, since it requires (always according to von Neumann) that our cells be able to implement organisms of any dimension. This challenge is met, as shown in Fig. 10, by decomposing a cell into molecules and tailoring the structure of cells to the requirements of a given application.

In conclusion, the original specifications of the historical automaton of von Neumann will be entirely satisfied after the implementation of a universal Turing machine on a multicellular array, and after the realization of the corresponding cells on our FPGA composed of molecules. The dream of von Neumann will then become a reality, with the additional properties of self-repair and real-time operation; moreover, we envisage the possibility of *evolving* the genome using genetic algorithms.

#### 3.2 A technical challenge: toward new robust integrated circuits

Keeping in mind that our final objective is the development of very large scale integrated (VLSI) circuits capable of self-repair and self-replication, as a first step, which is the subject of this paper, we have shown that a hierarchical organization based on four levels (molecule, cell, organism, population of organisms) allows us to confront the complexity of real systems. The realization of demonstration modules at the cellular level and at the molecular level [29] demonstrates that our approach can satisfy the requirements of highly diverse artificial organisms and attain the two sought-after properties of self-repair and self-replication.

The programmable robustness of our system depends on a redundancy (spare molecules and cells) which is itself programmable. This feature is one of the main original

contributions of the Embryonics project. It becomes thus possible to program (or re-program) a greater number of spare molecules and spare cells for operation in hostile environments (e.g., space exploration). A detailed mathematical analysis of the reliability of our systems is currently under way at the University of York [24][25].

The multiplexer-based molecule is the main hardware prototype we realized in order to test the validity of our approach. However, the size of the demonstration module used to implement a single molecule prevents us from realizing systems which require more than a few logic elements. In the long term, we hope to overcome this difficulty through the realization of a dedicated VLSI circuit which will contain a large number of elements; in the short term, however, such a solution is not yet within our reach. To obtain a larger number of programmable elements, we investigated the possibility of exploiting a system based on an array of Xilinx FPGAs mounted on a single printed circuit board and configured so as to implement an array of molecules [6]. Such a system, while far from affording the same density as a VLSI chip, would nevertheless allow us to obtain a 4x4 array, i.e., 16 molecules.

To the best of our knowledge, there exist today few projects, industrial or academic, which aim at integrating the properties of self-repair and/or of self-replication on FPGAs. In the framework of the Embryonics project, a fine-grained FPGA based on a demultiplexer [3][10] and a coarse-grained FPGA based on a binary decision machine [11] have been developed at the Centre Suisse d'Électronique et de Microtechnique in Neuchâtel (Switzerland). A fine-grained FPGA based on a multiplexer is also under development at the University of York (United Kingdom) [12]. Industrial projects dealing with self-repairing FPGAs (without self-replication) are also underway at NEC (Japan) [7] and at Altera (U.S.A.) [15].

In our laboratory, the next major step in the Embryonics project is the design of the BioWatch 2001, a complex machine which we hope to present on the occasion of a major scientific and cultural event which will take place in the year 2001 in Switzerland. The function of the machine will be that of a self-replicating and self-repairing watch, implemented with macroscopic versions of our artificial cells and molecules (Fig. 11).

A far-future technical application of the Embryonics project is in the domain of nanotechnology [14]. The concept of a self-replicating machine, or "assembler", capable of arranging "the very atoms" was first introduced by Drexler as a possible solution to the problem of the increasing miniaturization of VLSI circuits: as manufacturing technology advances beyond conventional lithography, some new, accurate, and low-cost approach to the fabrication of VLSI circuits is required, and self-

replicating assemblers could be a remarkably powerful tool for this kind of application.

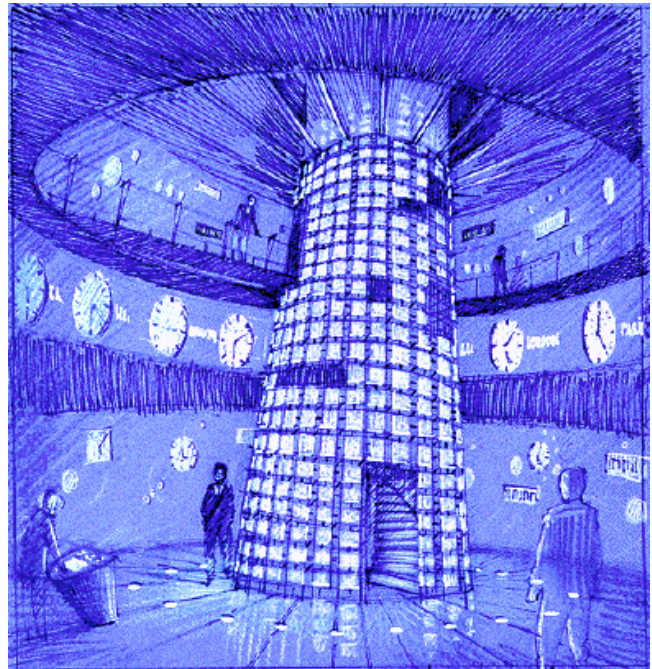


Fig. 11. An artist's rendition of a possible realization of the BioWatch 2001 [Art by Anne Renaud].

### 3.3 A biological challenge: artificial and natural genomes

In our Embryonics project, the design process for a multicellular automaton requires the following stages (Fig. 10).

1. The original specifications are mapped onto a homogeneous array of cells (binary decision machines with their associated read/write memory). The software (a microprogram) and the hardware (the architecture of the cell) are tailored according to the needs of the specific application (Turing machine, electronic watch, random number generator, etc.). In biological terms, this microprogram can be seen as the *operative genome* OG, or, in other words, the *operative part* of the final artificial genome. In the example of a specialized Turing machine [3][5], the parenthesis checker, the operative part of the genome consists of (Fig. 12): *coordinate genes* (Xcoord, Ycoord, localconfig, Initcond), which handle the computation of the coordinates and the calculation of the initial conditions, similar to the *homeboxes* or *HOX genes* recently found to define the general architecture of living beings; *switch genes* (G and SY0), used to express the functional genes according to the cell's position in the organism (that is,



according to the value of the cell's coordinates); and *functional genes* (Headgene, Tape gene), which effect the operative functions of our artificial organism (i.e., calculating the head and tape states), analogous to the genes which constitute the coding part of the natural genome.

2. The hardware of the cell is implemented with a homogeneous array of molecules. Spare columns are introduced in order to improve the global reliability. With our artificial cell, being analogous to the ribosome of a natural cell, the string of the molecular codes MOLCODES can be considered as the *ribosomic genome* RG or the *ribosomic part* of the final genome.
3. The dimensions of the final molecular array, as well as the frequency of the spare columns, define the string of data required by the finite state machine, the space divider, which creates the boundaries between cells. As this information will allow to create all the daughter cells starting from the first mother cell, it can be considered as equivalent to the *polymerase genome* PG or the *polymerase part* of the genome.

With respect to this design process, the programming of the molecular array takes place in reverse order.

1. The polymerase genome is injected in order to set the boundaries between cells.

2. The ribosomic genome is injected in order to configure the molecular FPGA and to fix the final architecture of the cell.
3. The operative genome is stored within the read/write memory of each cell in order for it to execute the specifications.

The existence of these different categories of genes is the consequence of purely logical needs deriving from the conception of our multicellular automaton.

One of the most promising domains of molecular biology, *genomics*, is the research of a syntax of the genome, that is, rules dictating the ordering of different parts of the genome, the genes [13]. One can imagine the artificial and the natural genomes sharing common, invariant properties. Should this indeed be the case, the Embryonics project could contribute to biology itself [16][26].

### Acknowledgments

This work was supported in part by the Swiss National Foundation under grant 21-54113.98, by the Consorzio Ferrara Recherche, Università di Ferrara, Ferrara, Italy, and by the Leenaards Foundation, Lausanne, Switzerland.

We are grateful to Anne Renaud for her rendition of a possible realization of the BioWatch 2001.

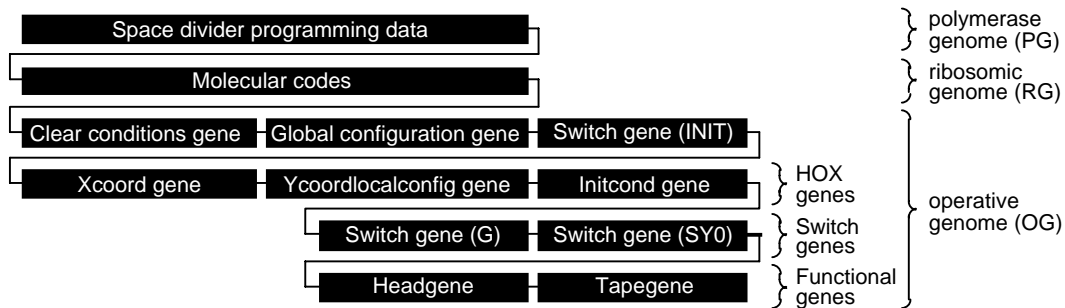


Fig. 12. The artificial genome of the parenthesis checker.

### References

[1] L. Wolpert. *The Triumph of the Embryo*. Oxford University Press, New York, 1991.

[2] M. Nicolaidis. "Future Trends in Online Testing: a New VLSI Design Paradigm?". *IEEE Design and Test of Computers*, Vol. 15, No. 4, 1998, p. 15.

[3] D. Mange, M. Tomassini, eds. *Bio-inspired Computing Machines: Towards Novel Computational Architectures*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland, 1998.

[4] J. von Neumann. *The Theory of Self-Reproducing Automata*. A. W. Burks, ed. University of Illinois Press, Urbana, IL, 1966.

[5] D. Mange, D. Madon, A. Stauffer, G. Tempesti. "Von Neumann Revisited: A Turing Machine with Self-Repair and Self-Reproduction Properties". *Robotics and Autonomous Systems*, Vol. 22, No. 1, 1997, pp. 35-58.

[6] G. Tempesti. *A Self-Repairing Multiplexer-Based FPGA Inspired by Biological Processes*. Ph.D. Thesis No. 1827, EPFL, Lausanne, 1998.

[7] A. Shibayama, H. Igura, M. Mizuno, M. Yamashina. "An Autonomous Reconfigurable Cell Array for Fault-Tolerant LSIs". In: *Proc. 44<sup>th</sup> IEEE International Solid-State Circuits Conference*, San Francisco, California, February 1997, pp. 230-231 and 462.

[8] M. Sipper. "Fifty Years of Research on Self-Replication: an Overview". *Artificial Life*, Vol. 4, No. 3, 1998, pp. 237-257.

- [9] M. A. Arbib. *Theories of Abstract Automata*. Prentice-Hall, New Jersey, 1969.
- [10] P. Marchal, C. Piguët, D. Mange, A. Stauffer, S. Durand. "Embryological Development on Silicon". *Artificial Life IV*, MIT Press, 1994, pp. 365-370.
- [11] P. Nussbaum, B. Girau, A. Tisserand. "Field Programmable Processor Arrays". In M. Sipper, D. Mange, A. Perez-Urbe, eds., *Evolvable Systems: From Biology to Hardware*, Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Berlin, 1998, pp. 311-322.
- [12] C. Ortega, A. Tyrrell, "MUXTREE revisited: Embryonics as a Reconfiguration Strategy in Fault-Tolerant Processor Arrays". In M. Sipper, D. Mange, A. Perez-Urbe, eds., *Evolvable Systems: From Biology to Hardware*, Lecture Notes in Computer Science, Vol. 1478, Springer-Verlag, Berlin, 1998, pp. 206-217.
- [13] D. Duboule. "The Evolution of Genomics". *Science*, Vol. 278, 24 Oct. 1997, p. 555.
- [14] R. C. Merkle. "Making Smaller, Faster, Cheaper Computers". *Proceedings of the IEEE*, Vol. 86, No. 11, November 1998, pp. 2384-2386.
- [15] C. F. Lane, S. T. Reddy, B. I. Wang. *Means and apparatus to minimize the effect of silicon processing defects in programmable logic devices*. Altera Corporation. U.S. Patent 5592102. Filed Oct. 19, 1995.
- [16] M. Barbieri. "The Organic Codes: The Basic Mechanism of Macroevolution". *Rivista di Biologia / Biology Forum* 91, 1998, pp. 481-514.
- [17] K. Roy, ed. "A D&T Roundtable: Online Test". *IEEE Design & Test of Computers*, Vol. 16, No. 1, January-March 1999, pp. 80-86.
- [18] Y. Zorian. "Testing the Monster Chip". *IEEE Spectrum*, Vol. 36, No. 7, July 1999, pp. 54-60.
- [19] G. D. Watkins. "Novel Electronic Circuitry", Predictive Paper Reprint. *Proceedings of the IEEE*, Vol. 86, No. 11, November 1998, p. 2383.
- [20] P. Kuekes. "Molecular Manufacturing: Beyond Moore's Law". Invited Talk. *Proc. Field-Programmable Custom Computing Machines (FCCM'99)*, Napa, CA, April 1999.
- [21] J. R. Heath, P. J. Kuekes, G. S. Snider, R. S. Williams. "A Defect-Tolerant Computer Architecture: Opportunities for Nanotechnology". *Science*, Vol. 280, No. 5370, 12 June 1998, pp. 1716-1721.
- [22] R. F. Service. "Organic Molecule Rewires Chip Design". *Science*, Vol. 285, No. 5426, 16 July 1999, pp. 313-315.
- [23] S. R. Park, W. Burleson. "Configuration Cloning: Exploiting Regularity in Dynamic DSP Architectures". *Proc. ACM/SIGDA International Symposium on Field Programmable Gate Arrays (FPGA'99)*, Monterey, CA, February 1999, pp. 81-89.
- [24] C. Ortega, A. Tyrrell. "Reliability Analysis in Self-Repairing Embryonic Systems". *Proc. of the First NASA/DOD Workshop on Evolvable Hardware*, Pasadena, CA, July 1999, pp. 120-128.
- [25] C. Ortega, A. Tyrrell. "Self-Repairing Multicellular Hardware: A Reliability Analysis". In D. Floreano, J.-D. Nicoud, F. Mondada, eds., *Advances in Artificial Life*, Lecture Notes in Artificial Intelligence, Vol. 1674, Springer-Verlag, Berlin, 1999, pp. 442-446.
- [26] R. Gordon. *The Hierarchical Genome and Differentiation Waves*. World Scientific & Imperial College Press, Singapore and London, 1999.
- [27] D. Mange, M. Sipper, P. Marchal. "Embryonic electronics". *BioSystems*, Vol. 51, No. 3, 1999, pp. 145-152.
- [28] H. F. Restrepo, D. Mange, M. Sipper. "A Self-Replicating Universal Turing Machine: From von Neumann's Dream to New Embryonic Circuits". *Proc. 7th International Conference on Artificial Life*, Portland, OR, August 2000, to appear.
- [29] D. Mange, M. Sipper, A. Stauffer, G. Tempesti. "Toward Robust Integrated Circuits: The Embryonics Approach". *Proceedings of the IEEE*, Vol. 88, No. 4, April 2000, to appear.