

Embryonic electronics

Daniel Mange ^a, Moshe Sipper ^{a,*}, Pierre Marchal ^b

^a *Logic Systems Laboratory, Swiss Federal Institute of Technology, IN-Ecublens, CH-1015 Lausanne, Switzerland*

^b *Centre Suisse d'Electronique et de Microtechnique SA, CH-2007 Neuchâtel, Switzerland*

Received 6 August 1998; received in revised form 5 May 1999; accepted 5 May 1999

Abstract

Within the general domain of bio-inspired computing, a particular trend over the past few years has been that of constructing actual hardware devices that are inspired by nature. This paper describes one such project—*Embryonics* (embryonic electronics)—inspired in particular by the process of embryogenesis. Our ultimate objective is the construction of large-scale integrated circuits, exhibiting the properties of self-repair (healing) and self-replication, found until now only in living beings. We present the silicon-based artificial cell, followed by a description of mechanisms operating at the cellular level: cellular differentiation, cellular division, regeneration, and replication. We then present the cell's composition as an ensemble of lower-level elements, known as 'molecules'. As electronic chips grow evermore complex, the need for repair capabilities will become increasingly crucial. The Embryonics approach represents one possible way of confronting this pivotal problem. © 1999 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Embryonics; Ontogeny; Self-repair; Self-replication

1. Introduction

In recent years we are witness to a growing interest among engineers in nature's workings, giving birth to novel engineering methodologies inspired by biological processes. As examples one can cite the domains of artificial neural networks and evolutionary algorithms. Living beings are complex systems, exhibiting a range of desirable characteristics, such as evolution, adaptation, and

fault tolerance, which have proved difficult to realize using traditional engineering methodologies. This paper describes research whose inspiration is drawn from the process of ontogeny, and in particular from the embryonic process, by which multicellular organisms are formed. Over the past 5 years we and our colleagues have been developing the *Embryonics* (embryonic electronics) project, the ultimate objective of which is the construction of large-scale integrated circuits, exhibiting the properties of self-repair (healing) and self-replication, found until now only in living beings. Specifically, the ability to self-repair would allow an electronic 'organism' to recover from

* Corresponding author. Tel.: +41-21-693-2658; fax: +41-21-693-3705.

E-mail address: moshe.sipper@epfl.ch (M. Sipper)

minor faults, while the ability to self-replicate would allow such an organism to recover from a major fault—by creating a novel, faultless clone. Such systems will be more robust than current-day ones, able to function within complex dynamic environments, which cannot be fully specified in advance and, moreover, may change in time.

In a previous paper we discussed the general idea of ontogenetic hardware, describing a number of examples of ontogenetic systems developed in our laboratory (Sipper et al., 1997). Our aim in this paper is to expose the fundamental principles of the Embryonics project, focusing in particular on the biological inspiration. We present the silicon-based artificial cell, dubbed *biodule* (biological module), followed by a description of mechanisms operating at the cellular level: cellular differentiation, cellular division, regeneration, and replication (Section 2). We then present the cell's composition as an ensemble of lower-level elements, or 'molecules' (Section 3). We end in Section 4 with some concluding remarks about the application of the Embryonics approach within future electronic processors. This exposition is aimed at readers without an engineering background, toward which end we have eschewed most technical details relating to the actual electronic implementation (these can be found in the recent book by Mange and Tomassini (1998), as well as in several other publications (Marchal et al., 1994; Mange et al., 1997, 1998; Sipper et al., 1997, 1999; Tempesti et al., 1997).

Before proceeding, we wish to emphasize that our project is one of *bio-inspiration* and not one of *bio-mimicking*: the work presented herein is not a true reflection of biology, nor is it intended to be; rather, the biology is a source of inspiration for developing novel computing systems, which resemble (albeit abstractly) their biological counterparts. We are dealing here with a 'life-as-it-could-be' approach, as opposed to a 'life-as-we-know-it' one (Langton, 1992); the reader should bear this in mind when encountering biological terms that have taken on new meanings (at least with respect to their strict biological definitions).

2. The cellular level

2.1. *The structure and functioning of the cell*

There are four basic cell activities: multiplication, movement, change in character, and signaling (Wolpert, 1991, p. 7; Wolpert et al., 1998). In a silicon substrate, we demonstrate that it is possible to create artificial cells, exhibiting three of the above properties: cellular division (multiplication), during which a mother cell gives rise to two daughter cells, cellular differentiation, which enables a cell to specialize as a function of its environment, and cellular signaling, where a cell transmits and receives messages from its neighboring cells.

The three key cell structures are the cell membrane, the cytoplasm, and the nucleus (Wolpert, 1991, p. 7; Wolpert et al., 1998). We have developed an artificial cell, the *biodule*, essentially comprising the above three structures: (1) a plastic box constitutes the external membrane, ensuring the cell's material encasement and realizing all the electronic functions necessary for communication with neighboring cells; (2) a processor responsible for interpreting the genome constitutes the artificial ribosome (which, in living beings, operates in the cytoplasm); and (3) a random access memory (RAM) acts as the cell's nucleus, containing a copy of the entire genetic makeup, encoded as a linear sequence of genes (Fig. 1). In contrast to living cells, the material structure (plastic box, silicon substrate) and the energy supply (electricity) are given a priori and are therefore not implemented by internal cellular mechanisms (i.e. the metabolism is given in our case). This is due in part to current limitations imposed by the underlying hardware (configurable processors); with the advent of nanotechnology, it might be possible in the future to build systems with a true metabolism.

2.2. *Cellular differentiation*

Cells must accomplish two different (and difficult) tasks: first, they need to know where they are, i.e. acquire positional information, and second, they must use this information appropri-

ately (Wolpert, 1991, p. 37; Wolpert et al., 1998). Following an example given by Wolpert (1991), we demonstrate the process whereby each cell computes its coordinates, followed by their use in cellular differentiation, by using the Swiss flag as an example (Fig. 2a). This example is of course highly simplified with respect to biological reality, however, it provides a convenient explanatory vehicle. Our artificial organism is bi-dimensional, consisting of 25 cells, each of which can be either red or white.

A cell is characterized by its color and by a pair of horizontal and vertical coordinates X , Y , respectively; it may be connected to any one of its immediate neighbors to the north, south, east, and west (no diagonal connections). Note that connectivity is completely local.

Assuming arbitrarily that the mother cell is placed at the origin of the coordinate system with $X, Y = 1, 1$, the position of each cell is then defined as follows: the horizontal coordinate X

equals $WX + 1$, where WX is the X value of the cell's west neighbor, and, similarly, the vertical coordinate Y equals $SY + 1$, where SY is the Y value of the cell's south neighbor (Fig. 2b).

A possible representation of our artificial 'Swiss-flag' organism is in the form of a so-called *truth table*, or *lookup table* (Fig. 2c); this is an ordered, linear description, containing the color of each of the 25 cells. Using a binary code to represent X and Y entails three bits per coordinate. We can thus employ six auxiliary binary variables $X0, X1, X2$ and $Y0, Y1, Y2$ to represent X and Y , respectively. It can be shown that the truth table of Fig. 2c can be transformed into an equivalent form, known as a *binary decision tree* (Mange, 1992), as shown in Fig. 2d. Such a tree is composed of two types of elements: diamond-shaped *test elements*, analogous to *regulatory genes*, and square-shaped *output elements*, analogous to *structural genes*.

In most living beings the four-letter DNA string (or, to be more precise, its RNA equivalent) is sequentially executed by a chemical processor, the ribosome. Motivated by this biological mechanism, we have implemented a program that resides in each cell, first computing its coordinates, followed by an extraction of the appropriate gene which determines the cell's color. The horizontal coordinate X is thus computed by referring to the west neighbor's horizontal coordinate ($X = WX + 1$), while the cell's Y coordinate is similarly computed by referring to the south neighbor's vertical coordinate ($Y = SY + 1$). The sub-programs involved, called *Xcoord* and *Ycoord*, are shown in Fig. 2e, along with the binary decision tree of Fig. 2d, which is the sub-program called *Org_gene*. The ensemble of these three sub-programs *Xcoord*, *Ycoord*, and *Org_gene* constitutes the final genome *Org_genome* of our artificial organism.

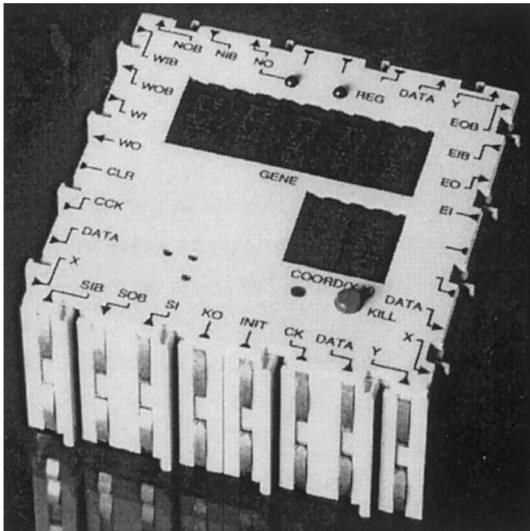


Fig. 1. An artificial cell: the biodule. A processor responsible for interpreting the genome constitutes the artificial ribosome, along with a random access memory (RAM) acting as the cell's nucleus, containing a copy of the entire genetic makeup, i.e. the genome. Displayed on the top cover are the cell's coordinates, as well as the specific gene within the genome that determines its functionality—these are acquired during cellular differentiation. The KILL button is used to induce the self-repair (regeneration) mechanism.

2.3. Cellular division

Cellular division involves the continual multiplications of a mother cell, ultimately resulting in the full-blown individual. In some cases, such as with the small nematode *Caenorhabditis elegans* (Gilbert, 1991, p. 265), the pattern of cleavages is

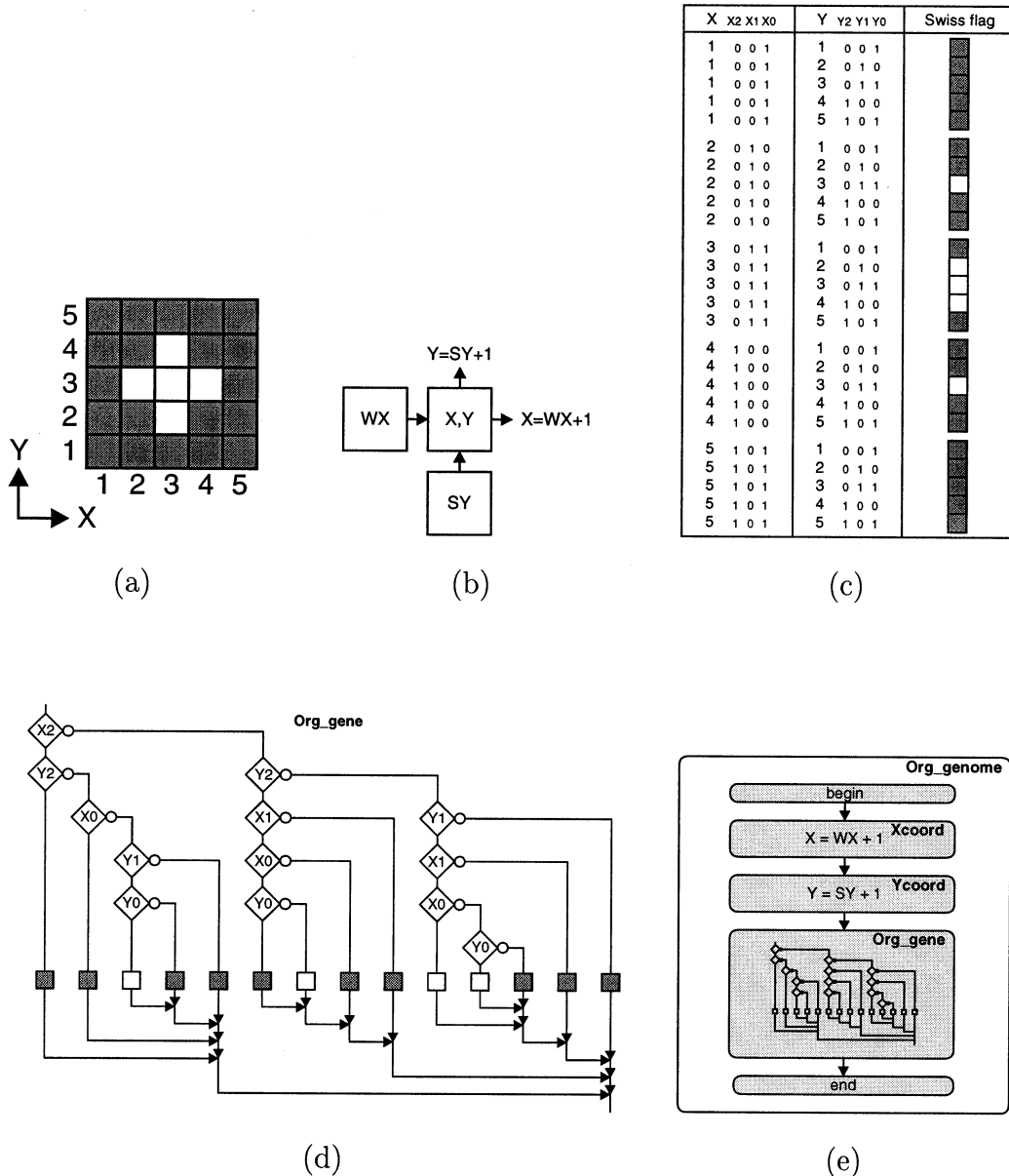


Fig. 2. Cellular differentiation. (a) The artificial ‘Swiss-flag’ organism is bi-dimensional, consisting of 25 cells, each of which can be either red (represented in the figure by dark gray) or white. A cell is characterized by its color and by a pair of coordinates X, Y . (b) Computing the position (coordinates) of a cell within the organism by referring to the coordinates of the neighboring cells to the west and to the south. (c) A complete listing (lookup table) of each cell’s color. Coordinates are given both in decimal form as well as in the equivalent binary representation, consisting of three bits. (d) Representing the lookup table as a binary decision tree. Such a tree is composed of two types of elements: diamond-shaped *test elements*, analogous to *regulatory genes*, and square-shaped *output elements*, analogous to *structural genes*. (e) The complete genome of the Swiss-flag organism, *Org_genome*, consists of an ensemble of three sub-programs: *Xcoord*, *Ycoord*, and *Org_gene*. *Xcoord* and *Ycoord* compute the cell’s X and Y coordinates, respectively, and *Org_gene* is the binary decision tree of (d).

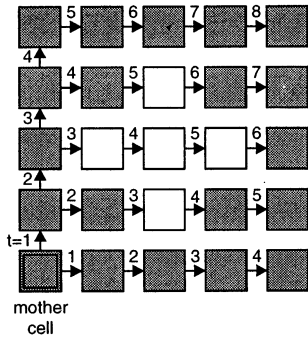


Fig. 3. Cellular division proceeds in discrete time steps (denoted by t), beginning at $t = 0$ with a mother cell, the *zygote*, arbitrarily placed at coordinates $X, Y = 1, 1$ (bottom left), containing the sole copy of the genome. At time $t = 1$, this genome is copied to the two neighboring cells (the daughter cells) to the north and to the east. This process continues until the bi-dimensional space is completely programmed, i.e. each cell contains a copy of the genome. In our case this occurs at $t = 8$, the time at which the farthest cell receives its copy of the genome.

known in its entirety. Indeed, the worm’s development can be described as a (deterministic) tree of divisions through time (Wolpert, 1991, pp. 51–52).

We derive our inspiration directly from the process of cellular division in living beings. This proceeds, in our case, in discrete time steps (denoted by t), beginning at $t = 0$ with a mother cell, the *zygote*, arbitrarily placed at coordinates $X, Y = 1, 1$, containing the sole copy of the genome (Fig. 3). At time $t = 1$, this genome is copied to the two neighboring cells (the daughter cells) to the north and to the east. This process continues until the bi-dimensional space is completely programmed, i.e. each cell contains a copy of the genome. In our example this occurs at $t = 8$, the time at which the farthest cell receives its copy of the genome. As with embryonic development in living beings, each mother cell produces at most two daughter cells. The temporal process of Fig. 3 can therefore be described by a sort of binary decision tree, referred to by biologists as a *lineage* (e.g. Gilbert, 1991, p. 265). We note that contrary to biological cellular division, where the mother cell disappears during the process, in our case the ensemble of cells composing the bi-dimensional space is conserved. Moreover,

a cell’s lineage, i.e. its temporal position in the sequence of divisions, has no role in its differentiation. Only the spatial positioning matters, i.e. the cell’s coordinates within the cellular space, determined during the differentiation phase.

2.4. Regeneration and replication

Many aspects of regeneration seem related to embryonic regulation and can be considered in terms of replacing those positional values of cells that have been lost (Wolpert, 1991, pp. 171–179; Wolpert et al., 1998). This mechanism can be imitated by proceeding as follows: identifying the cell to be eliminated (by pressing the KILL button of the biodule, see Fig. 1), the entire column to which this cell belongs is thereby considered ‘dead’ and is inactivated (the $X = 2$ column in Fig. 4a). All columns to the right of the dead one are then displaced one position to the right. The original Swiss-flag organism has now been regenerated with a ‘scar’ in the previous $X = 2$ column. It is clear that the process of regeneration requires

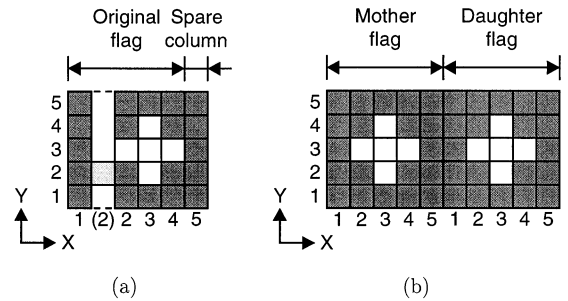


Fig. 4. (a) Regeneration. Pressing the KILL button of the biodule (Fig. 1) identifies the cell to be eliminated (the $X, Y = 2, 2$ cell above). The entire column to which this cell belongs is thereby considered ‘dead’ and is inactivated (the $X = 2$ column in the figure). All columns to the right of the dead one are then displaced one position to the right. The original Swiss-flag organism has now been regenerated with a ‘scar’ in the previous $X = 2$ column. (b) Replication. In analogy to hydra, the Swiss-flag organism can self-replicate, provided the following two conditions are satisfied: (i) there is a sufficient number of reserve cells (in our case at least 25, partitioned into five columns to the right of the organism); and (ii) the coordinate computation produces a cycle. In the figure, repetition of the horizontal coordinates pattern, that is, $X = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, produces a copy, a Swiss-flag ‘daughter’ of the original Swiss-flag ‘mother.’

as many reserve columns to the right as there are dead cells needing repair (a single reserve column is shown in Fig. 4a). Note that the detection of a malfunctioning cell is done manually, by having an outside observer press the KILL button. This shall be rectified ahead by introducing automatic fault detection at the molecular level.

Hydra is a small (about 0.5 cm long), glove-shaped animal with tentacles for catching prey at one end and a sticky foot at the other end (Wolpert, 1991; Wolpert et al., 1998). Its regenerative and regulative powers are due to its normal mode of reproduction, which is by budding: since a whole new hydra is remodeled and patterned from tissue in the adult body column, budding is essentially the same as regeneration (Wolpert, 1991, p. 176; Wolpert et al., 1998).

In analogy to hydra, our artificial organism can self-replicate, provided the following two conditions are satisfied (Fig. 4b): (i) there is a sufficient number of reserve cells (in our example at least 25, partitioned into five columns to the right of the organism); and (ii) the coordinate computation produces a cycle (in our example $X = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1 \dots$). As the same pattern of coordinates produces an identical pattern of genes, replication is achieved if the genome program *Org_genome*, associated with the cellular space, produces several occurrences of the original pattern of coordinates. In our example the repetition of the horizontal coordinates pattern, that is, $X = 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$, produces a copy, a Swiss-flag ‘daughter’ of the original Swiss-flag ‘mother.’ Given a sufficiently large cellular space, replication of an organism can occur several times, in both the horizontal and vertical directions.

2.5. Structural, regulatory, and homeotic genes

In our artificial organism, the genome is described in the form of a sequential program, composed of a series of instructions. In order to reduce the number of inter-cellular connections, this program is copied from one cell to another in a serial manner, that is bit by bit. The genome of Fig. 2e can thus be regarded as a string of bits, within which three types of genes can be iden-

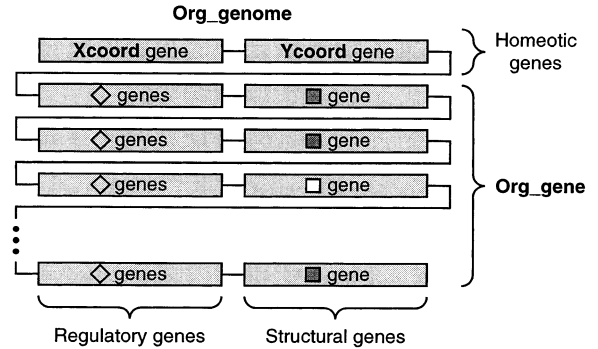


Fig. 5. Structural, regulatory, and homeotic genes. Representing the genome of Fig. 2e as a sequence of bits, one can identify three types of genes: (1) *homeotic genes* (Xcoord and Ycoord), that are responsible for computing the spatial position, i.e. the cell's *X* and *Y* coordinates; (2) *regulatory genes*, that control the expression of structural genes, in accordance with the cell's coordinates; and (3) *structural genes*, that produce the operational functions of the organism (the cell's color).

tified, which we have named in accordance with their natural counterparts (Fig. 5): (1) *homeotic genes* (Xcoord and Ycoord), that are responsible for computing the spatial position, i.e. the cell's *X* and *Y* coordinates; (2) *regulatory genes*, that control the expression of structural genes, in accordance with the cell's coordinates; and (3) *structural genes*, that produce the operational functions of the organism (in our example, the color of the cell). The existence of these three types of genes results from purely logical considerations that arise in the design of such a multicellular system. It is interesting to observe a posteriori such a relationship with molecular biology.

The Swiss-flag example is highly simplified with respect to biological reality, its main purpose being pedagogical. One simplification has to do with the Swiss-flag's genome being *descriptive* rather than *generative*—it delineates in a one-to-one manner the precise makeup of the organism (i.e. the color specification of each cell). A natural genome usually does not contain a description of the animal to which it will give rise, but rather contains a generative program for making it. In other applications (e.g. the biowatch described in Sipper et al. (1997, 1999)) we do use a generative

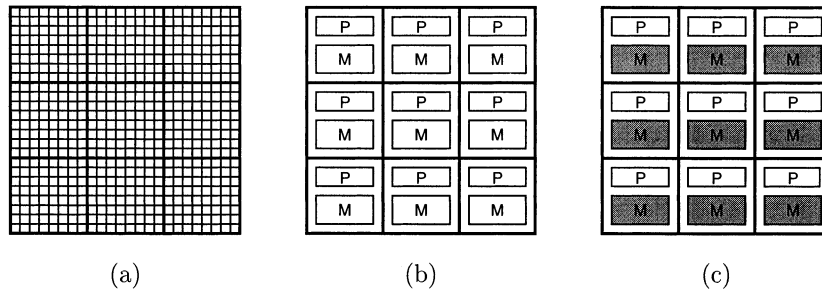


Fig. 6. The molecular composition of the artificial cell. Each cell is composed of yet finer elements, referred to as molecules. The system as a whole now possesses a three-part genome, corresponding to the three phases it undergoes: (a) loading the first part of the genome sets the boundaries between cells, i.e. fixes the dimensions of the molecular array; (b) loading the second part of the genome programs the system at the molecular level to act as a processor (P) with memory (M) that implements the cellular-level decoding; (3) and, finally, loading the third part of the genome into the random access memory of each cell programs the system at the cellular level to carry out its intended function (this is represented in the figure by shaded memory).

genome, which comprises a program to be executed rather than a precise cellular description, i.e. it is a recipe rather than a blueprint.

3. The molecular level

In the cellular system described above the detection of a malfunctioning cell is done manually, requiring that an outside observer press the KILL button, i.e. pinpoint the fault. To overcome this shortcoming, as well as to attain a property known as universal construction,¹ we have recently added a molecular level to the extant cellular level. In this new addition to the Embryonics family each cell is composed of yet finer elements, referred to as molecules. The system as a whole now possesses a three-part genome, corresponding to the three phases it undergoes (Fig. 6): (1) loading the first part of the genome sets the boundaries between cells, i.e. fixes the dimensions of the molecular array; (2) loading the second part of the genome programs the system at the molecular level to act as a processor with memory that implements the cellular-level decoding; and finally (3) loading the third part of the genome into the random access memory of each cell pro-

grams the system at the cellular level to carry out its intended function (e.g. the Swiss flag described herein or the biowatch described in Sipper et al. (1997, 1999)). The presence of a completely programmable molecular ‘tissue’ allows the implementation of a cell of any dimension (subject, of course, to the availability of sufficient hardware resources), thus enabling universal construction.

A major advantage of this novel two-level system is the ability to *automatically* detect a malfunctioning cell, and repair it, provided there is a sufficient number of reserve molecules (within the cell). When these latter are exhausted, and only in this case, the cell is incapable of repair at the molecular level—it thereupon (automatically) generates a KILL signal that brings about the regenerative process at the cellular level (as described in the previous section). The automatic detection of faults is based on so-called BIST (Built-In Self Test) techniques (Mange and Tomassini, 1998; McCluskey, 1986; Tempesti et al., 1997).

4. Concluding remarks

Observing the evolution of microprocessor technology over the past 30 years one can make an educated guess as to the direction it will take in the future: microprocessors will become more complex, evincing an ever-increasing number of

¹ A universal constructor can essentially construct any machine whose description is given by an artificial genome. This term was introduced by von Neumann in his seminal work on self-replication (von Neumann, 1966).

basic elements. Since the overall chip size remains roughly the same (due to packaging constraints) this means that the size of the elemental unit will continually shrink. One of the prime objectives of chip producers is to maintain power consumption at an acceptable level. Thus, the growing number of elemental units in future processors will need to consume less and less power (per unit); the downside is that such low-power chips are more faulty. We thus hold that the future will see the self-repair issue moving to the center stage of chip design.

We hope one day to see the coming of Embryonics-based chips, able to self-repair both at a coarse-grained (cellular) level, as well as at a fine-grained (molecular) level. These will be truly healing machines.

Acknowledgements

We are grateful to several people with whom we have been collaborating over the years, including Dominik Madon, Eduardo Sanchez, André Stauffer, and Gianluca Tempesti, from the Logic Systems Laboratory at the Swiss Federal Institute of Technology, Lausanne, Switzerland, and Serge Durand and Christian Pigué, from the Centre Suisse d'Electronique et de Microtechnique SA, Neuchâtel, Switzerland. We also wish to thank David B. Fogel and the anonymous referee for helpful comments. This work was supported in part by grants 20-42270.94 and 2000-049349.96 from the Swiss National Science Foundation, by a grant from the Leenaards Foundation, and by a grant from the Consorzio Ferrara Ricerche.

References

- Gilbert, S.F., 1991. *Developmental Biology*, 3rd edn. Sinauer Associates, Boston, MA.
- Langton, C.G., 1992. Preface. In: Langton, C.G., Taylor, C., Farmer, J.D., Rasmussen, S. (Eds.), *Artificial Life II. SFI Studies in the Sciences of Complexity*, vol. X. Addison-Wesley, Redwood City, CA, pp. xiii–xviii.
- Mange, D., 1992. *Microprogrammed Systems: an Introduction to Firmware Theory*. Chapman and Hall, London.
- Mange, D., Tomassini, M. (Eds.), 1998. *Bio-Inspired Computing Machines: Toward Novel Computational Architectures*. Presses Polytechniques et Universitaires Romandes, Lausanne, Switzerland.
- Mange, D., Madon, D., Stauffer, A., Tempesti, G., 1997. Von Neumann revisited: a Turing machine with self-repair and self-reproduction properties. *Robotics and Autonomous Syst.* 22 (1), 35–58.
- Mange, D., Sanchez, E., Stauffer, A., Tempesti, G., Marchal, P., Pigué, C., 1998. Embryonics: a new methodology for designing field-programmable gate arrays with self-repair and self-replicating properties. *IEEE Trans. VLSI Syst.* 6 (3), 387–399.
- Marchal, P., Pigué, C., Mange, D., Stauffer, A., Durand, S., 1994. Embryological development on silicon. In: Brooks, R.A., Maes, P. (Eds.), *Artificial Life IV*. MIT Press, Cambridge, MA, pp. 365–370.
- McCluskey, E.J., 1986. *Logic Design Principles with Emphasis on Testable Semicustom Circuits*. Prentice-Hall, Englewood Cliffs, NJ.
- Sipper, M., Mange, D., Stauffer, A., 1997. Ontogenetic hardware. *BioSystems* 44 (3), 193–207.
- Sipper, M., Mange, D., Sanchez, E., 1999. Quo vadis evolvable hardware? *Commun. ACM* 42 (4), 50–56.
- Tempesti, G., Mange, D., Stauffer, A., 1997. A robust multiplexer-based FPGA inspired by biological systems. *J. Syst. Architecture* 43 (10), 719–733.
- von Neumann, J., 1966. Edited and completed by A.W. Burks. *Theory of Self-Reproducing Automata*. University of Illinois Press, Illinois.
- Wolpert, L., 1991. *The Triumph of the Embryo*. Oxford University Press, New York.
- Wolpert, L., Beddington, R., Brockes, J., Jessell, T., Lawrence, P., Meyerowitz, E., 1998. *Principles of Development*. Oxford University Press, Oxford, UK.