

A New Paradigm for Developing Digital Systems Based on a Multi-Cellular Organization

Daniel MANGE, Serge DURAND, Eduardo SANCHEZ, André STAUFFER, Gianluca TEMPESTI
The Swiss Federal Institute of Technology - CH 1015 LAUSANNE (Switzerland)
Phone: (+41 21) 693 26 39 Fax: (+41 21) 693 37 05
e-mail: mange@di.epfl.ch

Pierre MARCHAL, Christian PIGUET
Swiss Center for Electronic Technology - CH 2000 NEUCHÂTEL (Switzerland)
Phone: (+41 38) 205 662 Fax: (+41 38) 205 630
e-mail: marchal@csemne.ch

Abstract

Embryological electronics or "Embryonics" is a new paradigm for developing digital systems of any complexity, endowed of universal computation, self-repair and self-reproduction capabilities. Embryonics, which can also be defined as a quasi-biological development of artificial systems, is based on the natural mechanism of development of living multi-cellular beings. Starting with a single mother cell containing the description of the organism in the form of a genome, the final organism is achieved through a succession of cell divisions, occurring with the differentiation of each cell, i.e. a specialization dependent essentially on the physical position of the cell (i.e. on its coordinates) in the given space. Applying this technique to the design of digital neural networks could lead to the development of artificial organisms simulating both the biological development and the final architecture of natural neural networks.

1. Artificial neural networks and artificial embryology

Engineering and natural sciences, traditionally proceeding along separate tracks, have drawn closer in these last decades. Engineers have been allured by certain natural processes; two recent attempts in this new direction will be discussed hereafter with the hope to combine them in a common methodology:

- *artificial neural networks* are based on a rough copy of the way the brain and its associated neural system are supposed to function; these networks have a structure completely different from that of the traditional computer (Von Neumann architecture) and surpass its performance for non-numerical applications such as symbol (handwriting) or shape (drawings, pictures) recognition;
- *artificial embryology*, which is the subject of this paper, is based on the natural mechanism of development of living multi-cellular beings; starting from a single mother cell, the zygote, containing the complete description of the organism in the form of a genome, the final organism is achieved through a succession of cell divisions, occurring concurrently with a differentiation of each cell, i.e. a specialization dependent essentially on the physical position of the cell (i.e. on its coordinates) in the given space.

Even if the results reported in this paper are beyond the present possibilities of silicon technology, the important point is the following: we propose a complete methodology for designing very complex integrated circuits with living creature properties, in particular self-repair and self-reproduction properties. Developing artificial neural networks with such a methodology could provide digital organisms simulating both the biological development (embryology) and the final architecture of natural neural networks.

2. Embryonics

Embryonics, i.e. the quasi-biological development of multi-cellular automata, is based on a *general hypothesis*, which describes the environment in which the development occurs, and on *three*

characteristics, which roughly approximate the biological mechanism of cellular development [Mange 94].

2.1 The general hypothesis: the environment

The general hypothesis describes the selected environment: (a) the automaton deals exclusively with the flow of information; the physical material (usually a silicon substrate) and the energy (power supply) are given a priori; (b) the physical space is two-dimensional and as large as desired; (c) the physical space is *homogeneous*, that is comprised by identical cells, all of which have the same internal architecture and the same connections with their neighbors; only the *state* of a cell (the combination of the values in its memories) can distinguish it from its neighbors; (d) the reproduction is asexual: the daughter automaton is identical to the mother automaton.

2.2 The three characteristics

The first characteristic is that of *multi-cellular organization*. The proposed automaton meets all the general hypotheses described before. The cell is essentially a random access memory (RAM) controlled by a small processor: a part of the state of this RAM, called the *gene*, completely defines the permanent operation of the cell. The value of the gene depends only on the position of the cell in the whole automaton. Fig. 1 illustrates the example of a simple artificial organism realized with nine cells, each cell being defined by a gene, from 1 to 9.

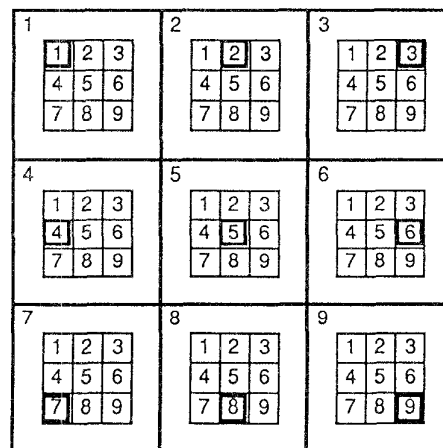


Fig. 1. A 9-cell organism with a genome in every cell.

The second characteristic is that of *cellular differentiation*. Each cell holds the complete description of the organism, i.e. the genome (Fig. 1) and computes its gene by extracting from the genome the

function which characterizes it. The gene of a cell depends on its coordinates, i.e. on its place within the organism.

The third characteristic is that of *cellular division*. At the beginning (time t0) only one cell, the mother cell, holds the genome of the organism (Fig. 2). After a first division (time t1), two adjacent cells (to the north and to the east in the diagram of Figure 2) hold a copy of the information of the mother cell. These two cells will then be themselves copied into their own neighbors (time t2), and so on until the plan has been fully implemented. In the proposed example, the farthest cell is duplicated at time t4.

3. Example: a cipher machine

In order to illustrate the different steps of our methodology, we have chosen a very simple example, the basic element of a cipher machine. In a first step (§ 3.1), we will show that the cipher function, like any other boolean function, can be represented by a binary decision tree. In a second step (§ 3.2), we will suggest that this binary decision tree can be realized by a new kind of field-programmable gate array (FPGA) satisfying the three basic characteristics discussed above: multi-cellular organization, cellular differentiation and cellular division. In the conclusion (§ 3.3 and 3.4), we will illustrate the self-reproduction and self-repair capabilities of this novel architecture.

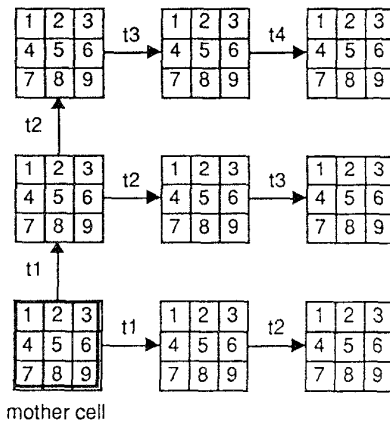


Fig. 2. Copying the genome from the mother cell into daughter cells.

3.1 Binary decision tree

Let C (Central), L (Left) and R (Right) be binary states of three successive elements of the cipher machine, a 1-dimensional cellular automaton described in [Wolfram 86](pp. 247 - 293). The future state of C is defined by the boolean equation:

$$C+ = C' L' R + C' L R' + C L'$$

This equation may be represented by a *binary decision tree* (Fig. 3) in which each diamond is a *test element*, performing the test of an input variable, and each rectangle is an *output element* producing an output state (0 or 1).

It is well known that any boolean function can be represented by a binary decision tree; this mode of representation and the related methods are described for example in [Akers 78], [Bryant 92] and [Mange 92].

We observe now that the tree in Figure 3 has the following characteristics: two sub-trees (ST1 and ST2) are trivial (their leaves are identical) and can be replaced by the logic constants "0" (for ST1) and "1" (for ST2). In order to make the final implementation easier, we transform slightly the binary decision tree of Figure 3 and we obtain the *normalized ordered binary decision tree* of Figure 4a which is characterized by the following constraints:

- each test element (diamond) is located on a rectangular array, at the intersection of a row and a column;
- each test element can only be connected to its nearest neighbours.

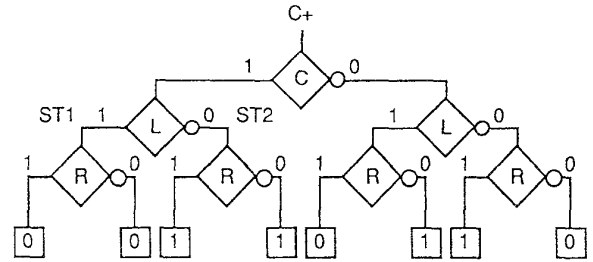


Fig. 3. Binary decision tree for the cipher machine.

3.2 A new field-programmable gate array for multi-cellular organization

In order to implement in a piece of hardware any normalized ordered binary decision tree satisfying the general hypothesis of § 2.1 (2-dimensional homogeneous array of operators and connections) and the first characteristic of § 2.2 (multi-cellular organization), we have developed a prototype of a new field-programmable gate array based on a fine-grained cell built around a one variable multiplexer and called MUXTREE.

The detailed description of MUXTREE cell is beyond the scope of this paper and can be found in [Marchal 94]. It will be shown here how this array of multiplexers is programmed in order to calculate the gene of each cell and therefore to deduce the genome of the whole system.

We start from the normalized ordered binary decision tree of Figure 4a and observe that a rectangular array of $3 \times 2 = 6$ multiplexers (test elements) is necessary to implement the cipher function. In our simple example, programming this array is very easy:

- each output of a test element which is a logic constant (0, 1) keeps its value in Figure 4b;
- each output of a test element heading South is marked "SOUTH" in Figure 4b, each output of a test element heading South-East is marked "EAST" and each output heading South-West is marked "WEST" in Figure 4b;
- each output of an unused test element is marked "Ø" (don't care condition) in Figure 4b.

If logic constants (0, 1) and cardinal points (SOUTH, EAST, WEST) are coded from 0 to 4 according to Figure 4c, we obtain a 2-digit word, i.e. a *gene*, for each test element.

Moreover, if we introduce discrete coordinates (x: horizontal coordinate, y: vertical coordinate), it is possible to address each cell of the array by a word x,y.

The second characteristic of § 2.2 (cellular differentiation) necessitates a copy of the whole genome in each cell's RAM, together with the right coordinates at the right place. A possible implementation is illustrated in Figure 5:

- each cell of the MUXTREE array is equipped with a RAM; each RAM contains the whole genome of the cipher function, i.e. the 6 genes of Figure 4c;
- each RAM is addressed by a word x,y, where x is the horizontal coordinate and y the vertical coordinate; the increment of the coordinates x and y is realized by simple adders (marked by "+1" in Fig. 5); in this particular example, x is a boolean variable and the adders for this coordinate are modulo 2 adders.

For normal operation, a rectangular array of 2 columns and 3 rows, i.e. 6 MUXTREE cells (with a multiplexer, programmable connections, RAM and coordinate adders) is sufficient. More cells are necessary in case of self-repairing or self-reproduction processes.

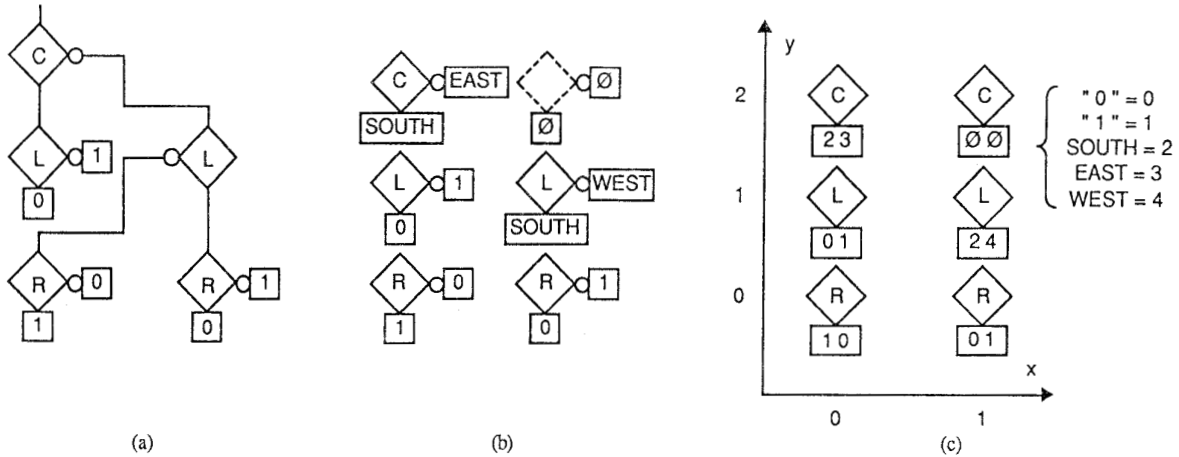


Fig. 4. A new field-programmable array for multi-cellular organization.
 (a) Normalized ordered binary decision tree.
 (b) Programming the connections of the multiplexer array.
 (c) Coding the genes of the 6 cells.

3.3 Self-reproduction mode

Duplicating a given function is straightforward. Suppose we have two spare columns. With the original border condition ($x = 0$ as in Fig. 5) and assuming a boolean horizontal coordinate x , the chain of modulo 2 adders ("1" in Fig. 5) will produce the following values for the horizontal coordinate x , read from left to right: 0, 1, 0, 1. Combined with the vertical coordinate y which is unchanged, the pattern of the horizontal coordinate x produces two copies of the original array of Figure 4c and, therefore, two copies of the same cipher function.

As our cipher element is part of a 1-dimensional cellular automaton, self-reproduction makes it possible to automatically generate several copies identical to the basic element, and to get finally an automaton of any length, the only limitation being the number of MUXTREE cells at our disposal.

3.4 Self-repair mode

Suppose we have a faulty cell in the first column of the cipher element (Figure 6) and we have at our disposal one spare column at the right of the array. A built-in self-test (BIST), as described in

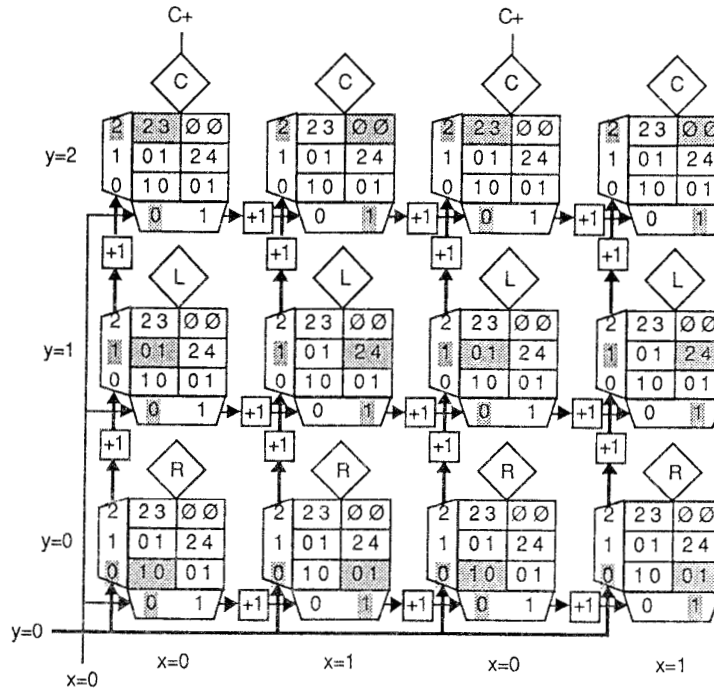


Fig. 5. Final cellular array for the normal operation (with the 2 first columns) and self-reproduction mode (with 4 columns).

[McCluskey 86], allows us to detect such a fault. It is possible to reprogram the cipher function in the cellular array simply by changing the border conditions. Replacing the border condition $x = 0$ by $x = 1$ results in a shift of the whole pattern by one column to the right (thanks to the chain of modulo 2 adders) and thus allows us to obtain a new and correct version of the cipher function.

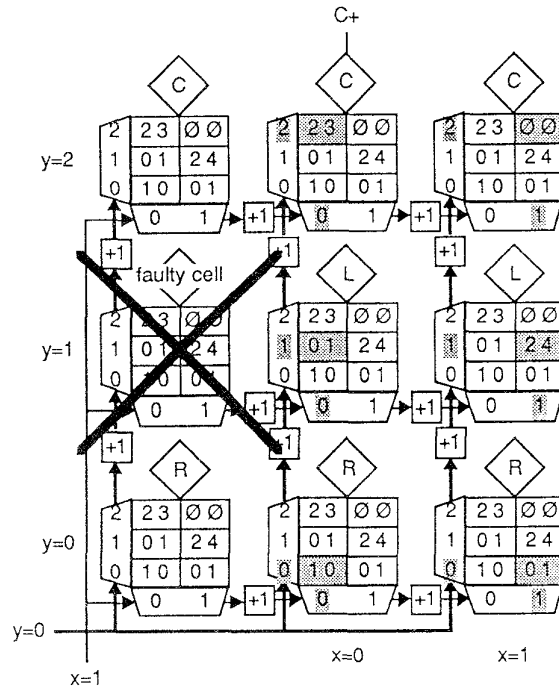


Fig. 6. Self-repair mode.

4. Conclusion

Using standard FPGA and RAM circuits, we have implemented a first prototype of our MUXTREE cell, which we have called "Biodule" [Durand 94]. With a couple of these biodules, we have successfully verified the self-reproduction and self-repair processes illustrated in Figures 5 and 6.

Our network of MUXTREE cells is capable of universal calculation and can therefore realize any universal Turing machine. Thus, our automaton can implement, within the limits imposed by the available material, any digital realization of an artificial neural network.

The ease of reprogramming by genome rewriting renders our automaton particularly well suited for the realization of evolvable circuits and, notably, of artificial neural networks hitherto implemented using conventional FPGA circuits [Gustin 92] [Ferruci 94] [Gschwind 94].

The capability of self-reproduction is particularly interesting for the implementation of neural networks composed of identical elements, such as those realized with massively parallel architectures [Distante 91] or systolic networks [Ienne 93].

Within the framework of the development of very-large scale integrated circuits, we believe that the marriage between artificial neural networks and quasi-biological cellular networks is a worthwhile research goal.

References

- [Akers 78] S.B.Akers, "Binary Decision Diagrams", *IEEE Transactions on Computers*, Vol. C-27, No 6, pp. 509-516, June 1978.
- [Bryant 92] R.E.Bryant, "Symbolic Boolean Manipulation with Ordered Binary-Decision Diagrams", *ACM Computing Surveys*, Vol. 24, No 3, pp. 293-318, September 1992.
- [Durand 94] S.Durand, A.Stauffer, D.Mange, "Biodule: An Introduction to Digital Biology", Technical Report, Logic Systems Laboratory, EPFL, Lausanne, 1994.
- [Distante 91] F.Distante, M.Sami, R.Stefanelli, G.Storti-Gajani, "Mapping Neural Nets onto a Massively Parallel Architecture: A Defect-Tolerance Solution", *Proc. of the IEEE*, Vol. 79, No 4, pp. 444 - 460, April 1991.
- [Ferruci 94] A.Ferruci, M.Martin, T.Geocarlis, M.Schlag, P.K.Chan, "A Field-Programmable Gate Array Implementation of a Self-Adapting and Scalable Connectionist Network", *Proc. of the ACM International Workshop on Field-Programmable Gate Arrays*, Berkeley, February 1994.
- [Gschwind 94] M.Gschwind, V.Salapura, O.Maischberger, "Space Efficient Neural Net Implementation", *Proc. of the ACM International Workshop on Field-Programmable Gate Arrays*, Berkeley, February 1994.
- [Gustin 92] V.Gustin, "Artificial Neural Network Realization with Programmable Logic Circuit", *Microprocessing and Microprogramming*, Vol. 35, pp. 187 - 192, 1992.
- [Ienne 93] P.Ienne, M.A.Viredaz, "GENES IV: A Bit-Serial Processing Element for a Multi-Model Neural Network Accelerator", *Proc. of the International Conference on Application Specific Array Processors*, Venezia, pp. 345 - 356, October 1993.
- [McCluskey 86] E.J.McCluskey, "Logic Design Principles with Emphasis on Testable Semicustom Circuits", Prentice-Hall, Englewood Cliffs, 1986.
- [Mange 92] D.Mange, "Microprogrammed Systems: An Introduction to Firmware Theory", Chapman & Hall, London, 1992.
- [Mange 94] D.Mange, A.Stauffer, "Introduction to Embryonics: Towards New Self-repairing and Self-reproducing Hardware Based on Biological-like Properties", *Artificial Life and Virtual Reality*, John Wiley, 1994.
- [Marchal 94] P.Marchal, A.Stauffer, "Binary Decision Diagram Oriented FPGAs", *Proc. of the ACM International Workshop on Field-Programmable Gate Arrays*, Berkeley, February 1994.
- [Wolfram 86] S. Wolfram, "Theory and Applications of Cellular Automata", World Scientific, Singapore, 1986.