# Self-replicating and Self-repairing Field-Programmable Processor Arrays (FPPAs) with Universal Computation

**Daniel MANGE, André STAUFFER and 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**

**Content Areas: artificial life**
Tracking Number: 000

## Abstract

The Embryonics project (for embryonic electronics), inspired by the basic process of molecular biology, adopts certain features of cellular organization and tranposes them to the 2-dimensional world of integrated circuits on silicon. Properties unique to the living world, such as self-replication and self-repair, are thus applied to artificial objects. Our final objective is the development of a wafer scale integrated circuit capable of self-replication and self-repair. These two biological-like properties seem particularly desirable for artificial systems meant for hostile (nuclear plants) or inaccessible (space) environments. Self-replication allows the complete reconstruction of the original device in case of a major fault, while self-repair allows a partial reconstruction in case of a minor fault.

In this paper, we shall demonstrate that a new kind of multicellular automaton, a field-programmable processor array (FPPA), is able to verify the property of universal computation as defined by von Neumann in his historic work. In a companion paper, it will be shown that this automaton can also verify the property of universal construction.

# Self-replicating and Self-repairing Field-Programmable Processor Arrays (FPPAs) with Universal Computation

**Content Areas: artificial life**
Tracking Number: 000


## Abstract

The Embryonics project (for embryonic electronics), inspired by the basic process of molecular biology [Wolpert, 1993], adopts certain features of cellular organization and tranposes them to the 2-dimensional world of integrated circuits on silicon. Properties unique to the living world, such as self-replication and self-repair, are thus applied to artificial objects. Our final objective is the development of a wafer scale integrated circuit capable of self-replication and self-repair. These two biological-like properties seem particularly desirable for artificial systems meant for hostile (nuclear plants) or inaccessible (space) environments. Self-replication allows the complete reconstruction of the original device in case of a major fault, while self-repair allows a partial reconstruction in case of a minor fault.
In this paper, we shall demonstrate that a new kind of multicellular automaton, a field-programmable processor array (FPPA), is able to verify the property of universal computation as defined by von Neumann in his historic work. In a companion paper [Mange, 1997a], it will be shown that this automaton can also verify the property of universal construction.


## 1 Historical survey

### 1.1 Von Neumann's self-replicating molecular automaton

Von Neumann's self-replicating automaton [Neumann, 1966], as well as the machines described in this article, is based on the following general hypotheses:

(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 2-dimensional and as large as desired;
(c) the physical space is *homogeneous*, that is comprised by identical *elements*, all of which have the same internal architecture and the same connections with their neighbors; only the *state* of an element (the combination of the values in its memories) can distinguish it from its neighbors;
(d) replication is considered a special case of growth: this process involves the creation of an *identical* organism by duplicating the genetic material of a mother entity onto a daughter one, thereby creating an exact clone.

To avoid conflicts with biological definitions, we do not use the term "cell" to indicate the parts of a cellular automaton, opting rather for the term "element"; in fact, in biological terms, a "cell" can be defined as the smallest part of a living being which carries the complete blueprint of the being, that is the being's *genome*.

In his historic work, von Neumann showed that a special *configuration* (a set of elements in a given state) of his automaton can implement a *universal constructor* able to replicate any machine described on a tape, and satisfying the following properties:

(a) *universal computation*, which makes it possible to replicate a universal Turing machine (qualitative property);
(b) *universal construction*, which makes it possible to replicate a machine of any dimensions (quantitative property).

The dimentions of von Neumann's automaton are substantial (in the order of 200,000 elements); it has thus never been physically implemented.

According to the biological definition of a cell, it can be stated that von Neumann's automaton is a unicellular organism: its unique genome is composed of the description of the universal constructor and the machine to be replicated, which are both described on the tape. We shall say that von Neumann's automaton is a *molecular automaton*, each element being a part of a cell, i.e., a *molecule*.

### 1.2 Langton's self-replicating loop

In order to construct a self-replicating automaton simpler than this of von Neumann, Langton [Langton, 1984] adopted more liberal criteria. He dropped the condition that the self-replicating unit must be capable of universal construction and computation. The size of Langton's loop is perfectly reasonable, since it requires 94 elements, thus allowing complete simulation. There is no universal construction nor calculation: the loop does

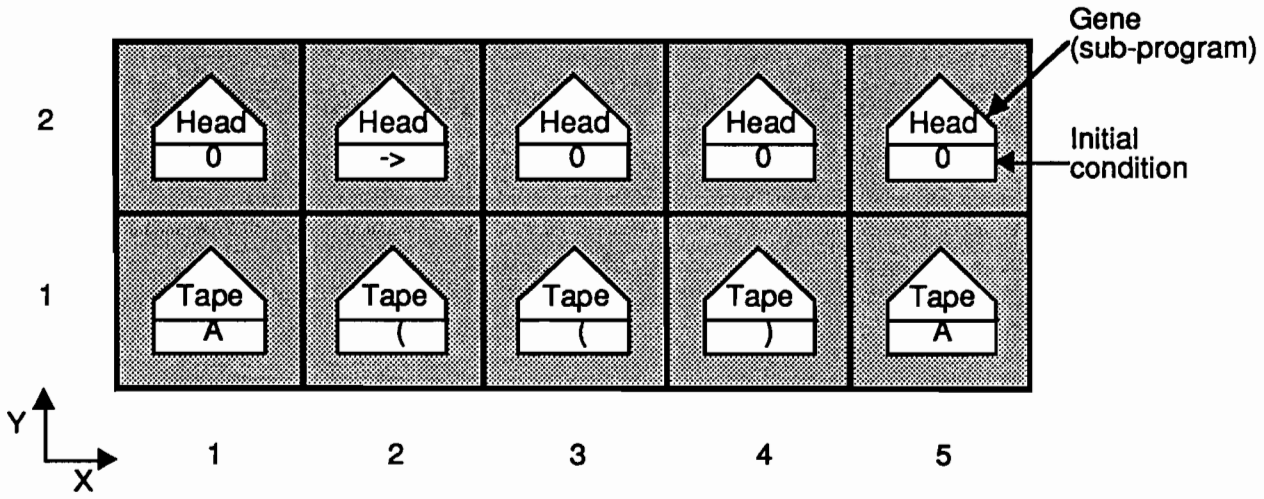


Fig. 1. Multicellular organization of a specialized Turing machine, a parenthesis checker.

nothing but reproduce itself; Langton's self-replicating loop represents a special case of von Neumann's self-replication of a universal constructor; the loop is a non-universal constructor, capable of building, on the basis of its genome, a single type of machine: itself.

Again referring to biological definitions, we can observe that Langton's self-replicating loop is a unicellular organism, i.e., a molecular automaton.

## 2 The foundations of Embryonics (embryonic electronics)

Our final objective is the realization of a computing machine offering at the same time the original properties of von Neumann's self-replicating automaton (universal computation and construction), and the simplicity of Langton's loop (reasonable size, allowing not only a complete software simulation, but also a physical realization through existing digital integrated circuits). This objective can be approached by introducing a new architecture, a *multicellular automaton*, roughly derived from the structure of multicellular living beings, and based on the following three features: multicellular organization, cellular differentiation, and cellular division. The proposed automaton meets all the general hypotheses described above for von Neumann's automaton. The element is a true "cell", according to the biological definition: it contains a random access memory (RAM), which stores the microprogram of the complete genome. This microprogram is executed by a small processor, a binary decision machine analogous to the ribosome of the living cell. The microprogram itself is decomposed in sub-programs which are equivalent to the different parts of the genome, i.e., the genes.

In this paper, we shall demonstrate that such an architecture makes it possible to verify the property of universal computation. In a companion paper [Mange, 1997a], it will be shown that this architecture can also verify the property of universal construction. In both cases, the two following properties, which are lacking in von Neumann's design, will also be verified:

(a) the self-replicating automaton is self-repairable;
(b) the self-replicating automaton is able to react on its environment in real time.

### 2.1 First feature: multicellular organization

The first feature is that of *multicellular organization*: the artificial organism is divided into a finite number of cells (Fig. 1), where each cell realizes a unique function, described by a sub-program called the *gene* of the cell. The same organism can contain multiple cells of the same kind (in the same way as living being can contain a large number of cells with the same function: nervous cells, skin cells, liver cells, etc.).

In order to demonstrate the property of universal computation, we will confine ourselves to a simple example of a 2-dimensional artificial organism (Fig. 1): a *specialized Turing machine*, a *parenthesis checker* [Minsky, 1967], implemented with ten cells and featuring two distinct genes, the *tape* gene and the *head* gene. Each cell is associated with some initial condition; in our example the head cells are distinguished by the initial values "0" and "→", the tape cells by "A", "(", and ")" values.

### 2.2 Second feature: cellular differentiation

Let us call *genome* the set of all the genes of an artificial organism, where each gene is a sub-program characterized by a set of instructions, by an initial condition, and by a position (its coordinates X,Y). Fig. 1 then shows the genome of our Turing machine, with the corresponding horizontal (X) and vertical (Y) coordinates. Let then each cell contain the entire genome (Fig. 2): depending on its position in the array, i.e., its place in the organism, each cell can interpret the genome and extract and execute the gene (with its initial condition) which configures it.

In summary, storing the whole genome in each cell makes the cell universal: it can realize any gene of the genome (including the initial condition), given the proper coordinates.

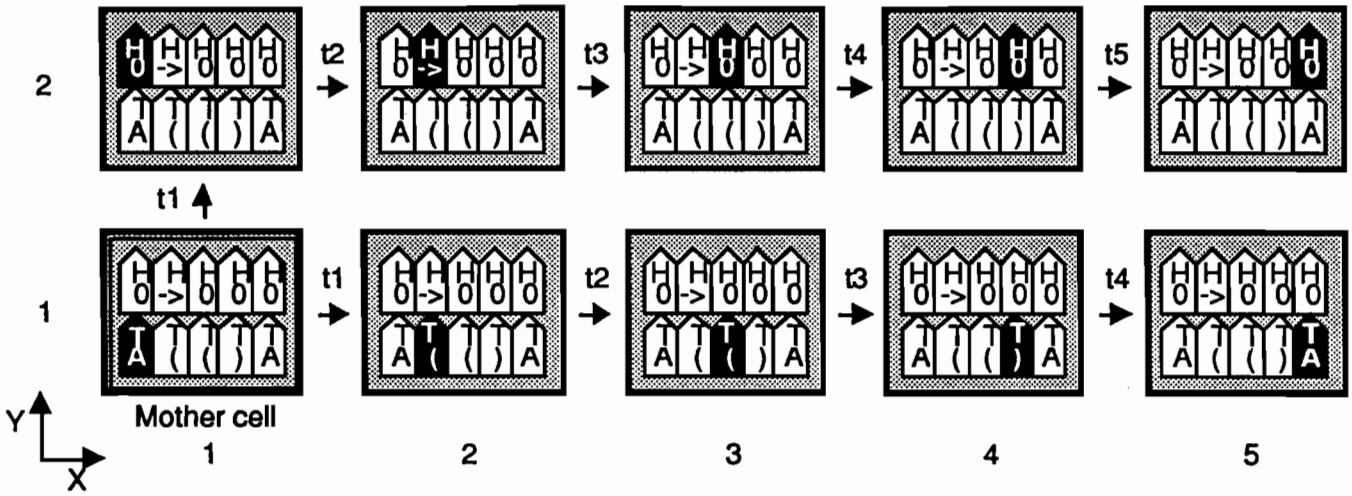


Fig. 2. Cellular differentiation and cellular division; t1 ... t5: five successive divisions.

### 2.3 Third feature: cellular division

At startup, the mother cell or *zygote* (Fig. 2), arbitrarily defined as having the coordinate X,Y=1,1, holds the one and only copy of the genome. At time t1, the genome of the mother cell is copied into the two neighboring (daughter) cells to the North and to the East. The process then continues until the 2-dimensional space is completely programmed. In our example, the furthest cell is programmed at time t5.

## 3 A field-programmable processor array (FPPA) with self-repair and self-replication capabilities

In all living beings, the string of characters which makes up the DNA is executed sequentially by a chemical processor, the *ribosome*. Drawing inspiration from this biological mechanism, we will use a microprogram to compute first the coordinates of the artificial organism, then the initial conditions of each cell, the tape gene and the head gene, and finally the complete genome. The detailed calculation of this microprogram is described elsewhere [Mange, 1997b]; its software implementation requires basically two kinds of instructions: a *test instruction* (if VAR else LABEL), and an *assignment instruction* (do X=DATA).

Each cell is implemented as an element of a new kind of coarse-grained programmable logic network or *field-programmable processor array* (FPPA), which is realized on a field-programmable gate array (FPGA) circuit [Mange, 1997b]; the FPPA element consists basically of a *binary decision machine*, executing the above mentioned instructions, a random access memory, storing the microprogram of the genome, and a number of programmable connections linking the cell to its four immediate neighbors (in the North, East, South and West directions).

### 3.1 Self-repair

In order to demonstrate self-repair, we have decided to add two spare cells in each row, to the right of the original Turing machine, all identified by the same horizontal coordinate (X=6 in Fig. 3). The spare cells may be used not only for self-repair, but also for the example of a Turing machine necessitating a growth of the tape of any, but non infinite, length.

The existence of a fault is detected by a KILL signal which is calculated in each FPPA element by a built-in self-test realized at the FPGA level (see the companion paper [Mange, 1997a]). 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=3 in Fig. 3). All the functions of the FPPA elements at the right of the column X=2 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 columns to repair (two spare columns in the example of Fig. 3). It also implies that the FPPA element has the capability of bypassing the faulty cell and shifting to the right all or part of the original cellular array.

### 3.2 Self-replication

The self-replication of an artificial organism rests on two hypotheses:

(a) there exists a sufficient number of spare cells (unused cells at the upper side of the array, at least ten for our example);
(b) the calculation of the coordinates produces a cycle (Y=1→2→1 in Fig. 4).

As the same pattern of coordinates produces the same pattern of genes (with the initial conditions), self-replication can be easily accomplished if the microprogram of the genome, associated to the homogeneous network of cells, produces several occurences of the basic pattern of coordinates (Y=1→2 in Fig. 1). In our example, the repetition of the vertical coordinate pattern, i.e., the production of the pattern Y=1→2→1→2 (Fig. 4),

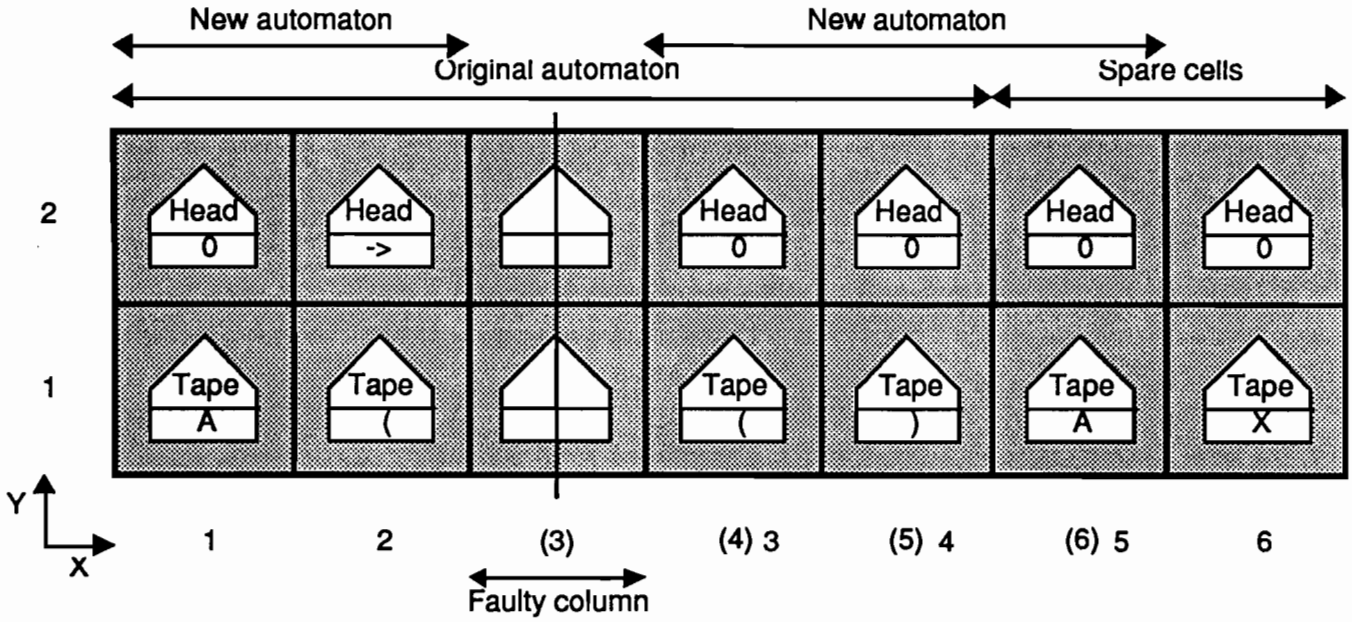


Fig. 3. Self-repair of a 10-cell parenthesis checker in a 14-cell array.

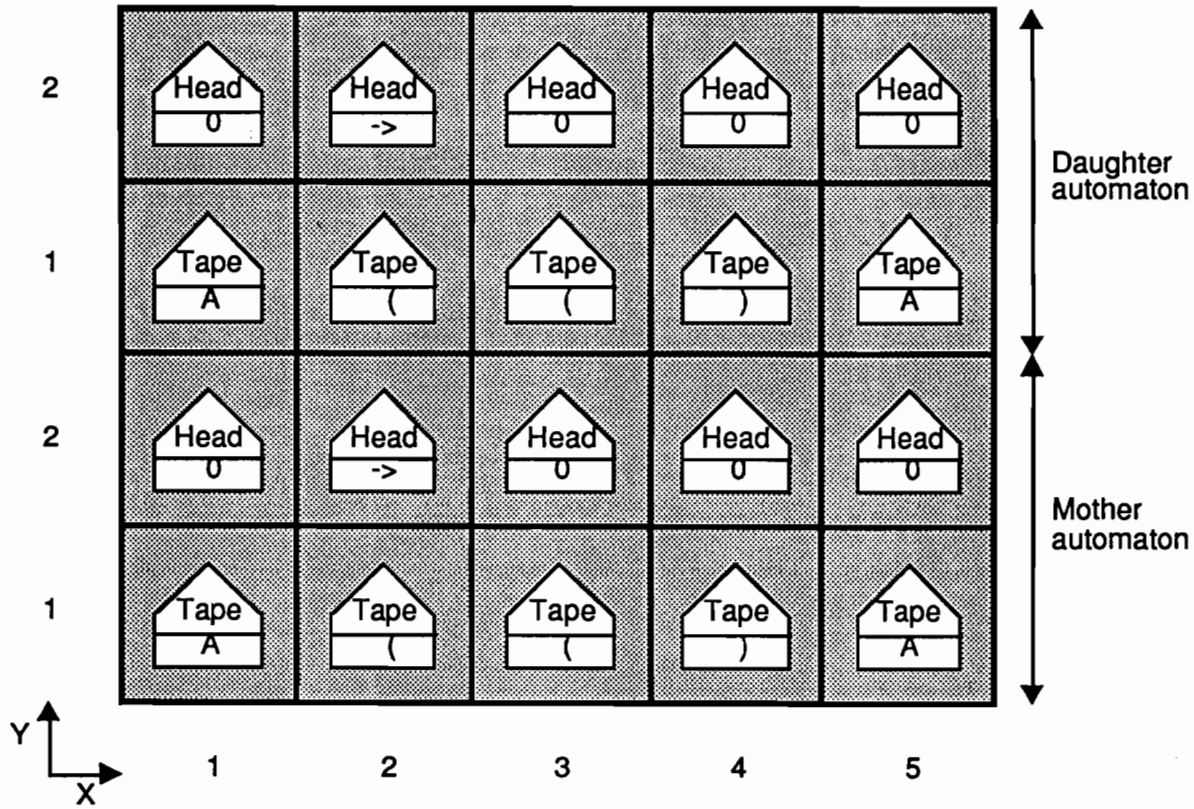


Fig. 4. Self-replication of a 10-cell parenthesis checker in a 20-cell array.

produces one copy, the *daughter automaton*, of the original or *mother automaton*. Given a sufficiently large space, the self-replication process can be repeated for any number of specimens in the Y axis (remember that X axis is reserved for self-repair and/or for a possible growth of the Turing machine).

With a sufficient number of cells, it is obviously possible to combine self-repair (or growth) toward the X direction and self-replication toward the Y direction.

## 4 Conclusions

three features of the Embryonics project: muticellular organization, cellular differentiation, and cellular division. The MICROTREE cell, itself realized with a commercial FPGA and a RAM, was finally embedded into a demonstration module called BIODULE 601 and we showed that an array of BIODULES 601 is capable of self-repairand self-replication [Mange, 1997b].

The property of *universal computation*, that is, the possibility of realizing, repairing, and replicating a universal Turing machine, can be verified with the MICROTREE cell subject to the following limitations:

(a) the setting of the initial conditions is limited by the finite dimensions of the X coordinate register (X=1...16); however, von Neumann and his successors assumed that the initial configuration was given a priori, and thus this limitation has no theoretical significance;

(b) a universal Turing machine, as described by [Minsky, 1967], obviously requires a redesign of the microprogram of the genome. The new design involves a larger number of variables for calculating the head and the tape states and thus a new and more complex architecture of the cellular space and/or of the basic MICROTREE cell as designed for our simple example of the parenthesis checker.

The property of *universal construction* poses problems of a different nature, since it requires (always according to von Neumann) that MICROTREE cells be able to implement artificial organisms of any dimension. The finite dimensions of our cells (memories, registers, etc.)are preventing us from meeting this requirement, a challenge which remains one of our main concerns. We will therefore be led to the design of a new cell with flexible architecture, whose specifications will be part of the genome. This challenge is the subject of a companion paper [Mange, 1997a].

## Acknowledgments

This work was supported in part by grant 20-42270.94 from the Swiss National Science Foundation.

## References

[Langton, 1984] C. G. Langton. Self-Reproduction in Cellular Autamata. *Physica D*, vol. 10, pp. 135-144, 1984.

[Minsky, 1967] M. L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Englewoods Cliffs, 1967.

[Neumann, 1966] J. von Neumann. *The Theory of Self-*

[Mange, 1977b] D. Mange, D. Madon, A. Stauffer and G. Tempesti. Von Neumann Revisited: a Turing Machine with Self-Repair and Self-Reproduction Properties. *Robotics and Autonomous Systems*, to appear.

[Wolpert, 1993] L. Wolpert. *The Triumph of the Embryo*. Oxford University Press, Oxford, 1993.