Half-day Colloquium Evolutionary Hardware Systems, IEE, London, 2 June 1999, pp.1/1-1/4.

EMBRYONICS: MULTI-CELLULAR AND MULTI-MOLECULAR DIGITAL SYSTEMS

Gianluca Tempesti†, Daniel Mange, André Stauffer
Logic Systems Laboratory
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland

1 Introduction

Drawing inspiration from biological systems for the design of robust digital circuits is far from being a novel concept: as far back as the 1950s, John von Neumann [8], one of the founding fathers of the field of computer architecture, began investigating the biological concept of self-replication in order to design robust machines (the idea being, of course, that at least one of the replicas would keep operating after the "death" of the original machine).

The design of machines capable of executing very complex tasks and at the same time capable of self-replication, however, proved to be too complex for the technology of the 1950s, and von Neumann's project never went beyond a theoretical study. Indeed, the realization of such machines is, arguably, too complicated even by today's standards, at least if von Neumann's approach were to be used. The *Embryonics* project [3, 6, 7] is an attempt at realizing machines which fulfill von Neumann's ambition to design robust self-replicating machines, but exploit a different approach so as to become feasible given today's technology.

The main feature of our Embryonic systems is the idea of structuring a digital machine (a digital circuit dedicated to the execution of an arbitrarily complex task, equivalent to a biological organism) as an array of identical processors (the digital equivalent of a biological cell), capable of altering their size and structure to fit the requirements of the task at hand. This adaptability is achieved by implementing the cells themselves using an array of small, programmable digital elements (a field-programmable gate array, or FPGA [2], whose elements are analogous to biological molecules).

2 The Cellular Level

To define our approach to partitioning the task to be executed into smaller fragments, we turned to the biological process of *ontogenesis* [5], the mechanism which determines the development of an organism from a single mother cell (the *zygote*) to a full-blown adult. The zygote divides, each offspring containing a copy of the genome (*cellular division*). This process continues (each new cell divides, creating new offspring, and so on), and each newly formed cell acquires a given functionality (i.e., liver cell, epidermal cell, etc.) depending on its surroundings, i.e., its position in relation to its neighbors (*cellular differentiation*).

The limitations of current technology do not allow us to realize this same process to the development of our artificial organisms. However, it is possible to design a system which follows the same basic approach. Our solution consists of implementing our organisms as two-dimensional arrays of identical processing elements, our artificial cells, each storing the same program, but executing different subsets of this program according to their position within the array. The program thus becomes the artificial equivalent of the biological genome, whose presence in each cell is the basis for the mechanism of ontogenesis, and the execution of different subsets of this program in each processor is the equivalent of cellular differentiation.

† Corresponding author. Phone: +41-21-6932676. Fax: +41-21-6933705. E-mail: tempesti@di.epfl.ch.

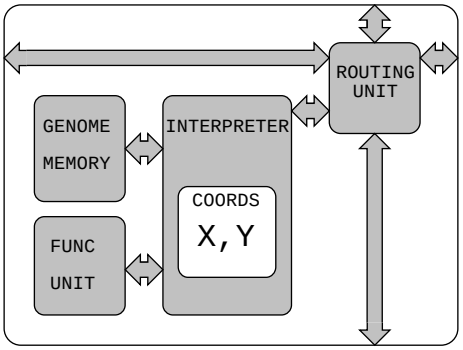

Figure 1: Structure of an electronic cell.

The architecture of our artificial cells follows directly from this approach (Fig. 1). Each cell must contain a memory to store the genome and an [X,Y] coordinate system to allow the cell to find its position within the array and thus its function. In addition, it requires an interpreter to read and execute the genome, a set of connections handled by a routing unit to exchange information with its neighbors, and a functional unit to execute the given task.

This approach, not very efficient from the standpoint of conventional circuit design (storing a complete copy of the genome program in each processor is redundant), allows us to easily exploit some of the most interesting features of biological systems. For example, the presence of the genome in each cell enormously simplifies the process of self-repair: if a hardware fault should "kill" one or more of our processors, a simple recomputation of the coordinates of the array (Fig. 2) allows the "dead" processors to be replaced (provided, of course, that a set of "spare" cells is available). Moreover, since the function of a cell depends on its coordinates, by allowing our coordinates to cycle (Fig. 3) we can obtain multiple copies of an organism with a single copy of the program, and thus achieve the self-replication of our machine (provided, of course, that enough processors are available).

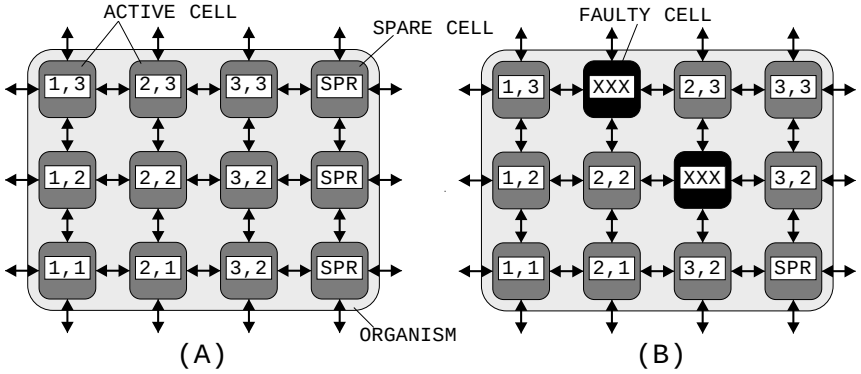

Figure 2: The electronic organism with no faults (A) and after reconfiguration (B).

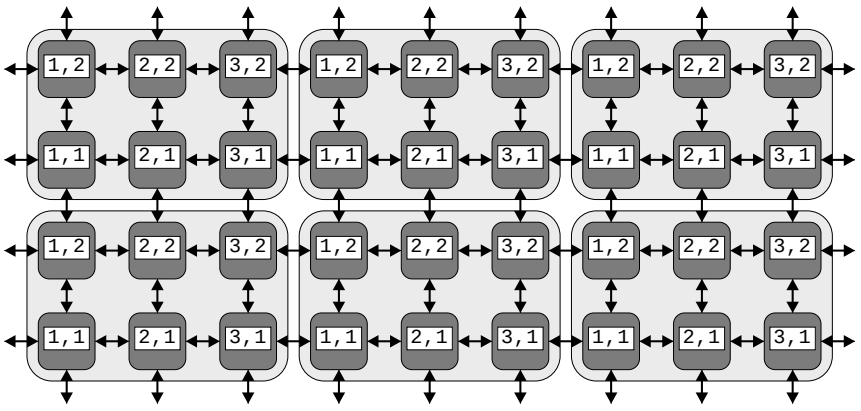

Figure 3: Multiple copies of the organism through coordinate cycling.

3 The Molecular Level

In order for our organisms to efficiently execute a given task, the architecture of our cells (e.g., their functional unit or the size of their memory) must vary depending on their intended functionality (as the structure of, for example, a liver cell is adapted to its function). We thus decided to realize our cells using a custom reconfigurable logic circuit (FPGA [7]),whose elements can then be seen as the artificial equivalent of *molecules*. . This FPGA (called *MuxTree*, for *tree of multiplexers*) has a fairly conventional architecture: its elements contain a tiny functional unit (essentially a two-input multiplexer and a flip-flop), a set of connections (to propagate signals across the array), and a configuration memory, implemented as a shift register (all registers in the array are connected together to form a single long shift register, through which the configuration bitstream propagates). To this architecture, we then added two rather non-conventional features: self-replication and self-repair.

Self-replication at the molecular level (not to be confused with self-replication at the cellular level) realizes the automatic replication of the configuration of a block of molecules throughout the array. It allows us to create, starting from the configuration of a single cell, a two-dimensional array of such cells. It is implemented (Fig. 4) by a tiny state machine (the *membrane*) placed between the FPGA elements. This membrane can partition the FPGA into blocks of molecules of arbitrary size (each block will then contain one cell) and "guide" the propagation of the configuration. In addition, it can define a set of columns in the array as containing spare molecules, required for the second original feature of our FPGA: self-repair.

Self-repair at the molecular level (again not to be confused with self-repair at the cellular level) consists of two separate subsystems. The first deals with self-test, that is, with the detection of the occurrence of a fault in the element, and is implemented using a mixture of more or less conventional self-test techniques [1]. The second is concerned with reconfiguration [4], that is, with re-arranging the array's configuration and connections to avoid the faulty element, exploiting the spare molecules defined by the membrane (Fig. 5).

4 Conclusion

In conclusion, Embryonics draws inspiration from the biological process of ontogenesis to develop digital systems based on three levels of complexity: the organism (an application-specific digital system), the cell (a simple processor), and the molecule (the element of a field-programmable gate array). Our approach includes robustness on all three levels: the organism can self-replicate, and thus achieve robustness by multiplication, our cells can differentiate, and thus achieve robustness through the recomputation of the coordinates, while our molecules can be reprogrammed, and thus achieve robustness through reconfiguration.

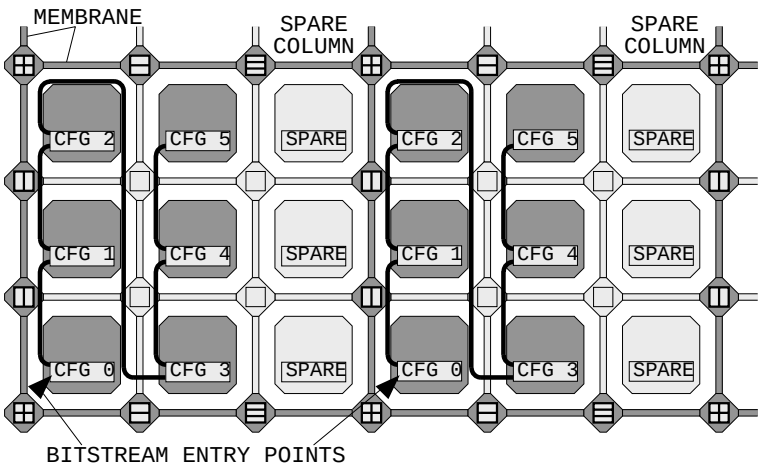

Figure 4: Self-replication of the configuration bitstream.

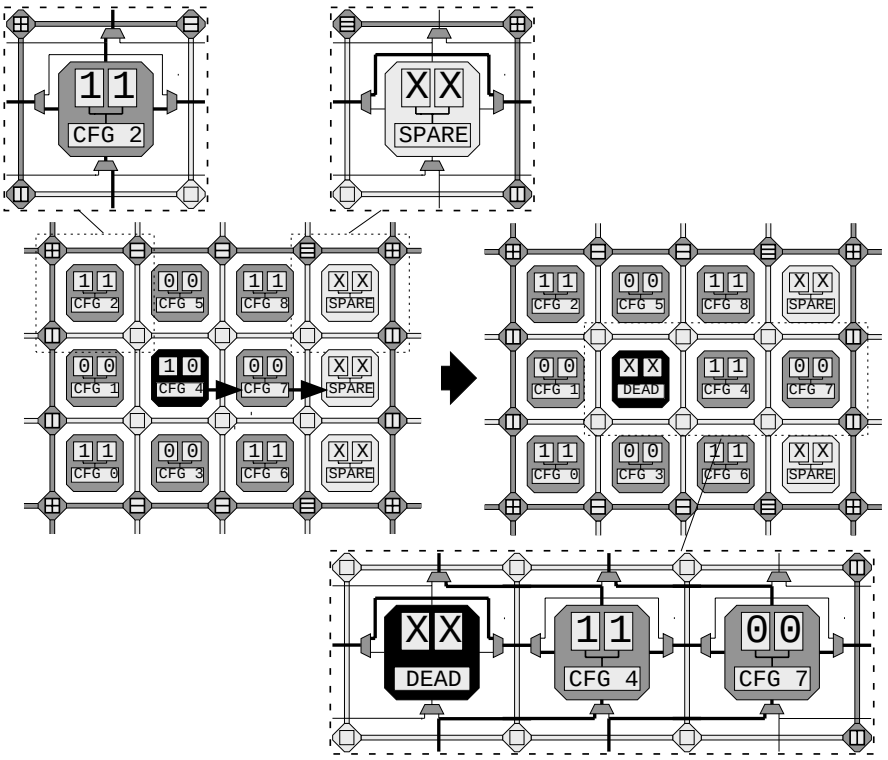

Figure 5: A fault is detected and the array is reconfigured to avoid the faulty element.

Similarly, the phenomenon of self-replication is present across all three levels: the molecules support, in hardware (thorough the cellular membrane), the self-replication of the cells, while the cells support, in software (through the cycling of the coordinates), the self-replication of the organism. The result of this process is the development of a very robust massively parallel computing platform, very much in accordance with the spirit of von Neumann's original research.

References

[1] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, New York, 1990.

[2] Stephen D. Brown, Robert J. Francis, Jonathan Rose, Zvonko G. Vranesic, *Field-programmable gate arrays*, Kluwer Academic Publishers, Boston, 1992.

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

[4] R. Negrini, M. G. Sami, and R. Stefanelli, *Fault Tolerance Through Reconfiguration in VLSI and WSI Arrays*, The MIT Press, Cambridge, MA, 1989.

[5] E. Sanchez, D. Mange, M. Sipper, M. Tomassini, A. Perez-Uribe, A. Stauffer. "Phylogeny, Ontogeny, and Epigenesis: Three Sources of Biological Inspiration for Softening Hardware". In T. Higuchi, M. Iwata, W. Liu, eds., *Proc. 1st Int. Conference on Evolvable Systems: From Biology to Hardware (ICES96)*, Lecture Notes in Computer Science, vol. 1259, Springer-Verlag, Berlin, 1997, pp. 35-54.

[6] G. Tempesti, D. Mange, A. Stauffer, "Self-Replicating and Self-Repairing Multicellular Automata". *Artificial Life* Vol. 4, No 3 Summer 1998.

[7] G. Tempesti. *A Self-Repairing Multiplexer-Based FPGA Inspired by Biological Processes. Ph.D*. Thesis, Swiss Federal Institute of Technology, Lausanne, 1998.

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