EVOLUTIONARY DESIGN OF SMART ANALOG CIRCUITS

ADRIAN STOICA* **RICARDO ZEBULUM***

DIDIER KEYMEULEN* **YILI JIN** [?]
[?]California Institute of Technology
Pasadena CA 911254

TAHER DAUD*
*Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91101

ABSTRACT

This paper discusses the use of Evolvable Hardware (EHW) in automatic synthesis of electronic circuits for computational intelligence (CI) hardware. EHW refers to HW design and self-reconfiguration using evolutionary/genetic mechanisms. Evolutionary experiments in simulations and with a Field Programmable Transistor Array (FPTA) chip in-the-loop demonstrate automatic synthesis of electronic circuits. Unconventional circuits (e.g. implementing CI mechanisms) for which there are no textbook design guidelines are particularly appealing to EHW. To illustrate this situation, one demonstrates here the evolution of circuits implementing parametrical connectives for fuzzy logics.

1. INTRODUCTION

The application of evolution-inspired formalisms to hardware design and self-configuration leads to the concept of evolvable hardware (EHW). In the narrow sense, EHW refers to self-reconfiguration of electronic hardware by evolutionary/genetic reconfiguration mechanisms. In a broader sense, EHW refers to various forms of hardware, from sensors and antennas to complete evolvable space systems that could adapt to changing environments and, moreover, increase their performance during their operational lifetime.

The paper starts with an overview of the main concepts of EHW. It then describes an effort toward building evolution-oriented devices, introducing the Field Programmable Transistor Array architecture, which has been used as the experimental platform for evolutionary experiments. The platform is quite flexible and supports implementation of both analog and digital circuits. While previous works [1,2] illustrated the implementation of several conventional building blocks for electronic circuits, such as amplifiers, filters and gaussian neuron, this paper illustrates the automatic design of the rather more unconventional circuits for combinatorial fuzzy logics.

The paper is organized as follows: Section 2 presents the components of an evolvable hardware system. Section 3 surveys some important evolutionary experiments and applications of evolvable hardware. Section 4 presents an evolution-oriented architecture

based on the concept of a Field Programmable Transistor Array. Section 5 illustrates how the FPTA can be used to evolve reconfigurable circuits for combinatorial fuzzy logic. Circuits implementing parametric triangular norms are evolved in software and in hardware directly on the chip.

2. EVOLUTIONARY SYNTHESIS OF ELECTRONICS

The main idea of evolutionary/genetic algorithms is inspired by the principle of natural selection. In nature the fittest individuals survive and reproduce passing along their genetic material to their offspring, who will inherit the characteristics that made the parents successful. Similarly, the evolution of artificial systems is based on a population of competing designs, the best ones (i.e. the ones that come closer to meeting the design specifications) being selected for further investigation. The offspring of this elite, in which pairs of parents were randomly selected for "mating", combine genetic material from two parents and may also suffer genetic "mutations. The offspring are the new generation of competing designs. This process of trial-and-error parallel search can last many generations, and can be constructed with many choices on how to implement reproduction, selection, etc.

The concept of evolvable hardware was born partially inspired by search/optimization/adaptation mechanisms and partially by the availability of reconfigurable devices such as Field Programmable Gate Arrays (FPGAs) [3]. Circuits can be evolved reconfiguring programmable devices (which is called *intrinsic* EHW) or evolving software models – descriptions of the electronic HW (referred to as *extrinsic* EHW).

Figure 1 illustrates the main steps of evolutionary design for electronic circuits.

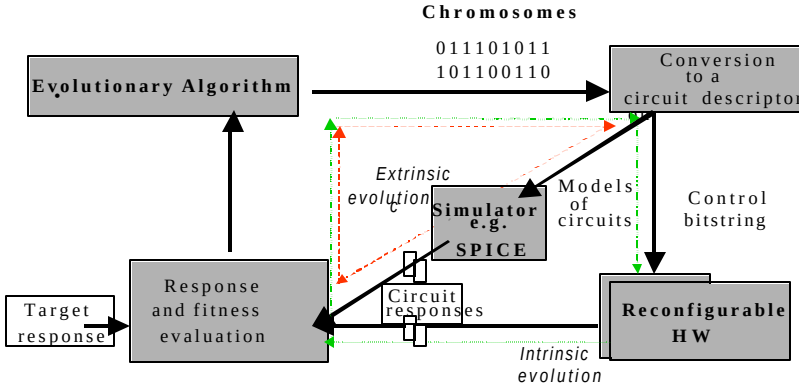


Figure 1 Evolutionary synthesis of electronic circuits

Each candidate circuit design is associated with a "genetic code" or chromosome. The simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encode a circuit. The first step of evolutionary synthesis is to generate a random population of chromosomes. In the case of extrinsic evolution, the chromosomes are then converted into a model that gets simulated (e.g. by a circuit simulator such as SPICE) and produces responses that are compared against specifications. In intrinsic evolution the chromosomes are converted into control bitstrings, which are downloaded to program the

reconfigurable device. The configuration bitstring determines the functionality of the cells of the programmable device and the interconnection pattern between cells. Circuit responses are compared against specifications of a target response and individuals are ranked based on how close they come to satisfying it.

3. EVOLUTIONARY EXPERIMENTS

A variety of circuits have been synthesized through evolution. For example, Koza used Genetic Programming (GP) to grow an "embryonic" circuit to one that satisfies desired requirements [2]. This technique was used to evolve a variety of circuits, ranging from filters to controllers. Evolution in hardware (intrinsic) was demonstrated by Thompson [3], who used an FPGA as the programmable device, and a Genetic Algorithm (GA) as the evolutionary mechanism. Evolutions of analog circuits reported in [2] were performed in simulations, without concern for a physical implementation, but rather as a proof-of-concept to show that evolution can lead to designs that compete with human designs, or even exceed them in performance.

4. THE FIELD PROGRAMMABLE TRANSISTOR ARRAY (FPTA)

The efforts toward hardware evolution have been limited to simple circuits. In particular, for analog circuits, this limitation comes from a lack of appropriate reconfigurable analog devices to support the search, which precludes searches directly in hardware and requires evolving in software on hardware device models. Such models require evaluation with circuit simulators such as SPICE; which need to solve differential equations and, for anything beyond simple circuits, they require too much time for practical searches of millions of circuit solutions. A hardware implementation may offer a substantial advantage in circuit evaluation time.

The idea of a field programmable transistor array was introduced in [1]. The FPTA is a concept design for hardware reconfigurable at transistor level. As both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA appears as a versatile platform for the synthesis of both analog and digital (and mixed-signal) circuits. The architecture is cellular, and has similarities with other cellular architectures as encountered in FPGAs (e.g. Xilinx X6200 family) or cellular neural networks. One key distinguishing characteristic relates to the definition of the elementary cell. The architecture is largely a "sea of transistors" with interconnections implemented by other transistors acting as signal passing devices (gray-level switches). Figure 2 illustrates an FPTA cell consisting of 8 transistors and 24 programmable switches.

The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011…", where by convention one can assign 1 to a switch turned ON and 0 to a switch turned OFF. Programming the switches ON and OFF defines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation (for low frequency circuits).

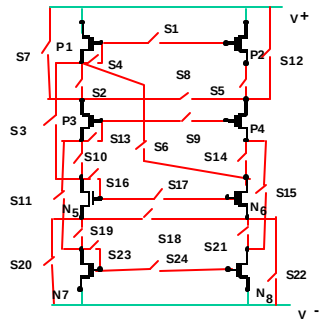


Figure 2. Module of the Field Programmable Transistor Array

5. EVOLVING RECONFIGURABLE CIRCUITS FOR FUZZY LOGICS

This section illustrates the evolutionary design of infinitesimal multi-valued logic circuits, more precisely circuits for fuzzy logics. The objective is to determine circuit implementations for conjunctions and disjunctions for fuzzy logics. In such logics, conjunction and disjunction are usually interpreted by a *T-norm* and by its dual *T-conorm* (*S-norm*) respectively. A function **T**: [0,1] x [0,1] => [0,1] is called a triangular norm (T-norm for short) if it satisfies the following conditions:

??associativity (**T**(x,**T**(y,z)) = **T**(**T**(x,y),z)),
??commutativity (**T**(x,y) = **T**(y,x)),
??monotonicity (**T**(x,y) = **T**(x,z), whenever y= z), and
??boundary condition (**T**(x,1) = x).

A function **S**: [0,1] x [0,1] => [0,1] is called a triangular conorm (T-conorm or S-norm for short) if it satisfies the conditions of associativity, commutativity, monotonicity, and the boundary condition **S**(x,0) = x. **S** and **T** are corresponding (or pairs) if they comply with De Morgan's laws [4, 5]. Frank's parametric T-norms and T-conorms (also refered to as fundamental T-norms/conorms in [4]) were the selected choice for modeling the logical connectives. The family of Frank T-norms is given by

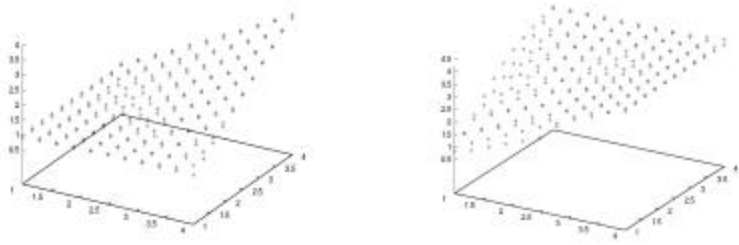
$$T_s(x,y) = \begin{cases} MIN(x,y) & \text{if } (s = 0) \\ x \cdot y & \text{if } (s = 1) \\ \log_s\left[1 + \dfrac{(s^x - 1).(s^y - 1)}{s - 1}\right] & \text{if } (0 < s < ?), s ? 1 \\ MAX(1, x + y) & \text{if } (s = ?) \end{cases} \quad (1)$$

The family of Frank T-conorms is given by

$$S_s(x,y) = \begin{cases} MAX(x,y) & \text{if } (s = 0) \\ x + y - x.y & \text{if } (s = 1) \\ 1 - \log_s \left[1 + \dfrac{(s^{1-x} - 1).(s^{1-y} - 1)}{s - 1} \right] & \text{if } ((0 < s < ?), \; s ? 1) \\ MIN(1, x + y) & \text{if } (s = ?) \end{cases} \quad (2)$$

Electronic circuits implementing the above equations can be used in implementations of fuzzy logic computations or in implementing fuzzy ST neurons. The following preliminary results illustrate the possibility of evolving circuits that implement **T** and **S**. The circuits were powered at 5V and the signal excursion was chosen between 1V (for logical level "0") and 4V (for logical level "1"). Intermediary values were in linear correspondence, i.e. 2.5V corresponds to logic level 0.5. etc. The experiments were performed both in software (SPICE simulations) and in hardware using 2 FPTA cells. The experiments used a population size of 128 individuals, were performed for 400 generations (with uniform crossover, 70% crossover rate, 4% mutation rate, tournament selection) and took around 15 minutes using 16 processors when evolving in simulations. Each switch in the FPTA cell has a control bit associated with it in a direct mapping. Hence there are 24 bits in the chromosome describing one cell. Four bits encoded the cells' interconnections. Therefore a two cell experiment used 52 bits (24*2+4).

Evolutionary experiments have been performed for different values of the parameter *s* (0, 1, 100). Due to space reasons, we show here the results for s = 100. Figures 3.a shows the response of circuits targeting the implementation of fundamental T-norms and Figure 3b shows the response of the circuit implementing the fundamental S-norm. The output is mapped on the vertical axis; values on axis are in Volts. The circuit for T-norm is shown mapped on two FPTA cells in Figure 4



(A) (B)

Figure 3 Response of circuit implementing the fundamental T-norm (A) and fundamental S-norm (B) for s=100 (?). Target characteristic shown with (+).

The approximation error achieved in these experiments ranged from 3.6% to a maximum of 9% MAPE (Mean Absolute Percent Error) in software and to a peak of 11.6% in hardware. Several approaches can be devised to improve these results in the future, such as transistor size optimization, use of more cells and allowing more

flexibility in the selection of the points where the inputs are applied, and where the output is collected.

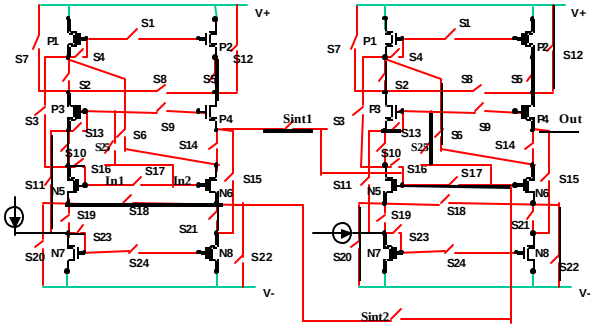


Figure 4 Evolved circuit implementing the fundamental T-norm for s=100 (with the response in Figure 3.a).

6. CONCLUSIONS

This paper presented an overview of Evolvable Hardware and described the concept of the FPTA as a result of an effort toward building evolution-oriented devices. It has been demonstrated how electronic circuits can be automatically synthesized to produce a desired functionality. We illustrated the aspects of using evolvable hardware for the design of unconventional circuits such as combinatorial circuits for fuzzy logics.

ACKNOWLEDGEMENTS

The work described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration.

REFERENCES

[1] A. Stoica, D. Keymeulen, R. Tawel, C. Lazaro and Wei-te Li. “Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digital CMOS circuits.” Proc. of 1st NASA/DoD Workshop on Evolvable Hardware, pp.77-84, Pasadena, CA, July , 1999.

[2] J. Koza, F.H. Bennett, D. Andre, and M.A Keane, “Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming”, *Proc. of Genetic Programming Conference*, Stanford, CA, 1996, 28-31

[3] A. Thompson, “An evolved circuit, intrinsic in silicon, entwined in physics”. In *Int. Conf. on Evolvable Systems*. Springer-Verlag LNCS 1259, 1997, 390-405

[4] D. Butnariu,. and E. P. Klement, , “Triangular norm-based measures and games with fuzzy coalitions”, Kluwer Academics, 1993

[5] A. Stoica,., “Synaptic and somatic operators for fuzzy neurons: which T-norms to choose?” In Proc. of 1996 Biennial Conference of the North American Fuzzy Information Processing Society - NAFIPS, Berkeley, CA, June 19-22, 55-58