Computer Science Journal of Moldova, vol.10, no.1(28), 2002

Evolutionary Design of Very Compact Analog Circuit Implementations of Fuzzy Systems

Adrian Stoica Ricardo Zebulum

Abstract

Evolutionary algorithms (EA) offer good promise for automated design of analog circuits as well as for adaptation and automatic reconfiguration of programmable devices. In particular, EA facilitate the design of analog circuits for very specific requirements, such as those related to the implementation of fuzzy operators, or even of complete fuzzy systems. The paper starts with a brief overview of the evolutionary design process and of a family of analog programmable devices that support on-chip evolution, illustrates evolutionary design with examples of evolved analog circuits implementing fuzzy parametric t-norms, and focuses on an evolved circuit with only 7 transistors, which approximates the control surface of a 2-input fuzzy controller, obtaining thus a mapping of a complete fuzzy system in a circuit with only a few transistors. The paper presents evidence that EA can provide very compact solutions for implementation of fuzzy systems, and that programmable analog devices are an efficient and rapid solution for rapid deployment of fuzzy systems.

1 Introduction

Evolutionary algorithms (EA) offer good promise for automated design of analog circuits as well as for adaptation and automatic reconfiguration of programmable devices. A variety of computational analog circuits and filters [1], [2], [3], as well as digital circuits [4] have been synthesized by EA. On the other hand, used with reconfigurable devices in the loop, EA were able to perform the configuration search directly in hardware, e.g. in experiments using Field Programmable Gate Arrays (FPGAs) [5], [6] or Field Programmable Transistor Arrays (FPTAs) [7], leading to device configurations (designs) that satisfied the imposed requirements. More details on current work in evolvable hardware can be found in [8] and [9].

This papers focuses on the application of EA to the design of a specific category of circuits, i.e. circuits used in the implementation of fuzzy systems. It presents both circuit solutions that could be fabricated as Application Specific Integrated Circuits (ASICs) and circuit topologies that can be mapped on programmable devices, such as the FPTAs. The claim is that EA offer the ability of automatic design of non-trivial analog circuits, such as those implementing parametric t-norms, and moreover, the design of very compact circuits that can approximate complete fuzzy systems.

The paper is organized as follows: Section 2 provides a brief overview of the main concepts of evolutionary design of electronic circuits. Section 3 presents a FPTA architecture used as experimental platform for evolutionary experiments and support for the rapid implementation of evolved fuzzy circuits. Section 4 illustrates how the FPTA can be used to evolve reconfigurable circuits implementing parametric triangular norms that model connectives in fuzzy logics. Section 5 presents a very compact evolved circuit that approximates a complete fuzzy system.

2 Evolutionary synthesis of analog circuits

The main idea of evolutionary algorithms is inspired by the principle of natural selection. Biological 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. 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. 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. Or, the chromosomes are transformed into a configuration bitstring downloaded into a programmable 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. Preparation for a new iteration loop involves the generation of a new population of individuals from the pool of the best individuals in the previous generation, with some individuals taken as they were and some modified by genetic operators, such as crossover and mutation. The process is repeated for a number of generations, resulting in increasingly better individuals. The process is usually ended after a given number of generations, or when the closeness to the target response has been reached. In practice, one or several solutions may be found among the individuals of the last generation.

3 Field Programmable Transistor Arrays

Evaluation of a circuit directly on a programmable device may offer a substantial advantage in circuit evaluation time; in certain cases the time for hardware evaluation can be seconds instead of days, as often the case when evaluation is in software. The FPTA is a concept design for hardware reconfigurable at transistor level introduced in [3]. 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, an example of cell being shown in Figure 1. The architecture is largely a "sea of tran-

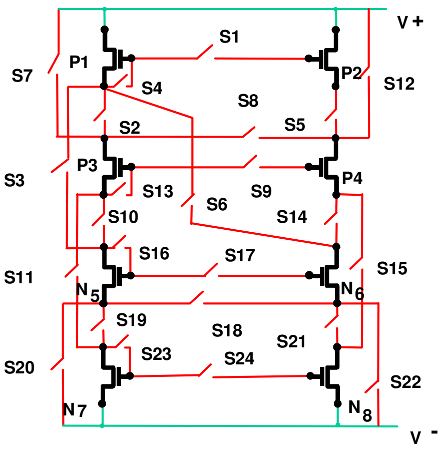


Figure 1. A FPTA cell consisting of 8 transistors and 24 programmable switches.

sistors" with interconnections implemented by other transistors acting as signal passing devices (gray-level switches), and with islands of RC resources in between.

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).

4 Evolving reconfigurable circuits for fuzzy logics

This section illustrates evolved circuits 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. Frank's parametric T-norms and T-conorms (also referred to as fundamental T-norms/conorms in [10]) were selected here for modeling the logical connectives. Their choice is justified by their special properties (see [10] and references therein), and because of the complexity of the equation for which no trivial compact analog circuit solution is apparent. The family of Frank T-norms and the family of Frank T-conorms are respectively given by:

$$T_s(x,y) = \begin{cases} MIN(x,y) & \text{if } (s=0) \\ x.y & \text{if } (s=1) \\ \log_s[1+\frac{(s^x-1)(s^y-1)}{s-1}] & \text{if } 0 < s < \infty, s \neq 1 \\ MAX(0, x+y-1) & \text{if } s=\infty \end{cases}$$

$$S_s(x,y) = \begin{cases} MAX(x,y) & \text{if } (s=0) \\ x+y-xy & \text{if } (s=1) \\ 1-\log_s[1+\frac{(s^{1-x}-1)(s^{1-y}-1)}{s-1}] & \text{if } 0 < s < \infty, s \neq 1 \\ MIN(1, x+y) & \text{if } s=\infty \end{cases}$$

Electronic circuits implementing the above equations can be used in implementations of fuzzy logic computations or in implementing fuzzy S-T neurons.

A series of experiments, presented in detail in [11] illustrate the possibility of evolving circuits that implement T-norms for various values of the parameter 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 hardware on FPTA chips. The software 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. Each switch in the FPTA cell has a control bit associated with it in a direct mapping. Thus there are 24 bits in the chromosome describing one cell. Interconnections

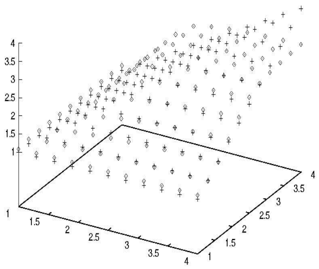


Figure 2. Simulated response of a circuit implementing the fundamental T-norm for s=0 (◇) (Target characteristic shown with (+). x,y axis are for inputs, z (vertical) is the output, T. Axes are in Volts.

experiments were done mostly with 4 bits. Thus a 2 cell experiment would use 52bits (24*2+4).

Figures 2, 3, and 4 show the response of circuits targeting the implementation of fundamental T-norms for s=0, s=1, and s=100 respectively. The diamond symbol (◇) marks points of simulated/measured response of evolved circuit, while the cross symbol (+) marks the points of an ideal/target response for the given inputs. The output (T) is mapped on the vertical axis; values on axis are in Volts. The circuit for T-norm with s=100 is shown mapped on two FPTA cells in Figure 5. From these graphs, it can be verified that the errors are observed mainly at the extremities of the domain.

More details on these results, including circuits for corresponding S-norm, are presented in [11].

The approximation error on FPTA architecture ranges from 3.6% to a maximum of 9% MAPE (Mean Absolute Percent Error) in software and to a peak of 11.6% in hardware.

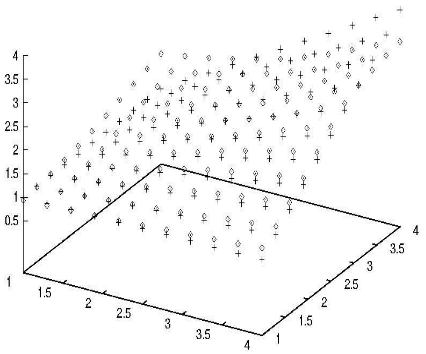


Figure 3. Response of a circuit implementing the fundamental T-norm for s=1(◇). Target characteristic shown with (+).

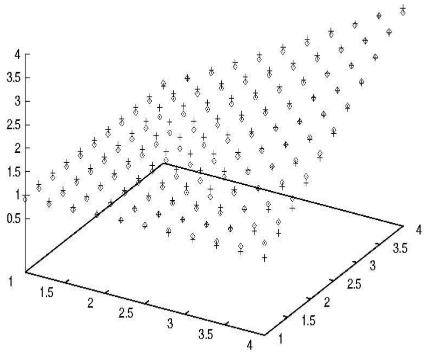


Figure 4. Response of a circuit implementing the fundamental T-norm for s=100 (◇). Target characteristic shown with (+).

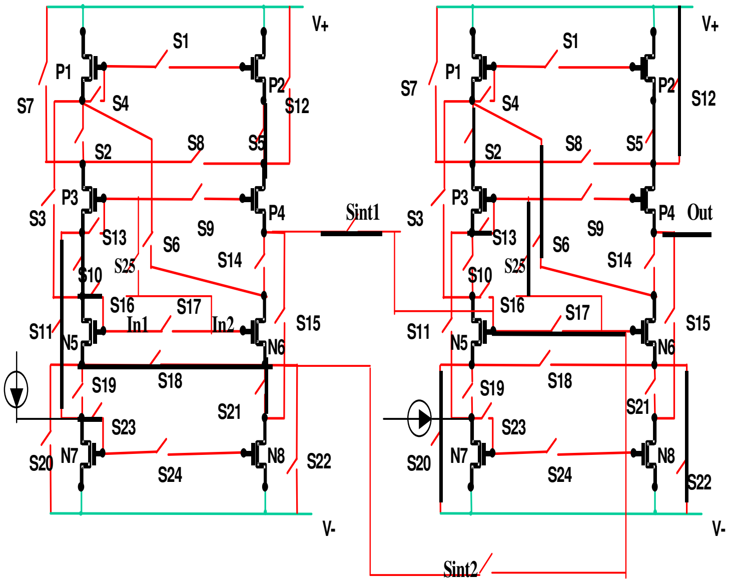


Figure 5. Example of evolved circuit implementing a fundamental T-norm circuit.

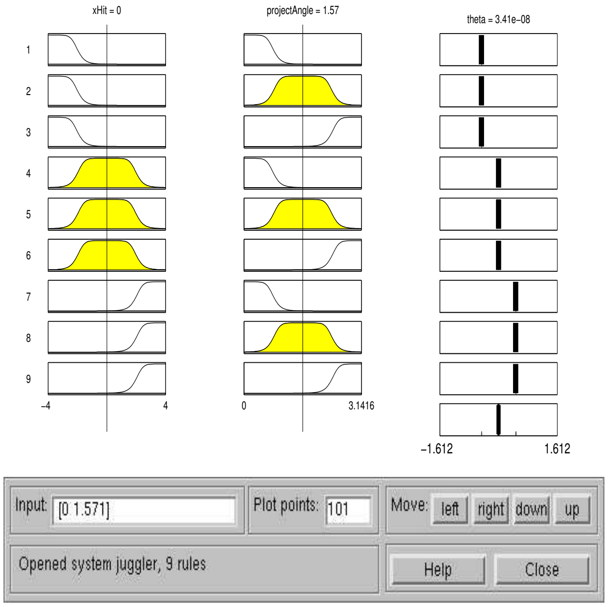
5 Evolution of analog circuits approximations of complete fuzzy systems

The results above, related to analog circuit of implementations for fuzzy operators are particularly relevant from the point of view of building fuzzy computation engines, especially those of programmable nature, meaning that some software code would change the function of the system. Most of the traditional fuzzy systems in use however, are quite simple in nature and the computation can be expressed in terms of a simple surface. An example is the control surface of a two-input fuzzy controller. A fuzzy circuit could be synthesized to approximate this surface. If the circuit is synthesized on a programmable cell structure, such as the FPTA cell, it can in fact be "programmed" to change function by a new encoding bitstring found through the evolutionary search; in this case the search replaces a conventional compiler taking inputs from specifications and mapping to a configuration based on the very clear architecture of the device. If a fixed solution suffices, then a very compact circuit that could be implemented in an ASIC, integrated with other silicon components is possible. Since the previous examples illustrated evolution on a programmable architecture, the following example illustrates evolution of a circuit with fixed topology.

The example chosen is that of a fuzzy controller provided as a demo for the popular MATLAB software [12]. The "ball juggler" is one of the demos of the MATLAB Fuzzy Logic Toolbox. The fuzzy controller for the ball juggler has two inputs and one control output. A screen capture illustrating the membership functions is shown in Figure 6. The controller is a simple Sugeno-type with 9 rules. A screen capture of the control surface is shown in Figure 7.

A circuit approximating the control surface was evolved and is presented in Figure 8. The response, presented together with the target surface for comparison are shown in Figure 9. The average error achieved was of 1.93%, and the maximum error to the target surface was 6.7%.

The circuit is rather robust, and was tested at variations in transistor sizes, supply voltage and temperature, with the following results:

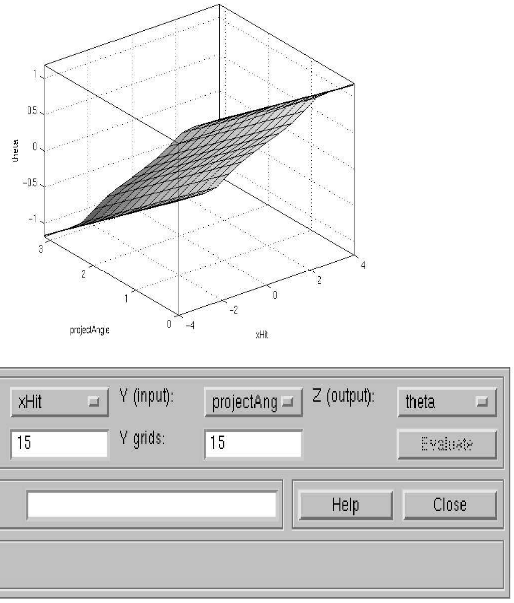


Input: [0 1.571]

Plot points: 101

Opened system juggler, 9 rules

Figure 6. Membership functions for the ball juggler fuzzy controller as seen on screen capture from Matlab Demo.



X (input): xHit
Y (input): projectAng
Z (output): theta
X grids: 15
Y grids: 15
Ref. Input:
Ready

Figure 7. Surface of the ball juggler fuzzy controller.

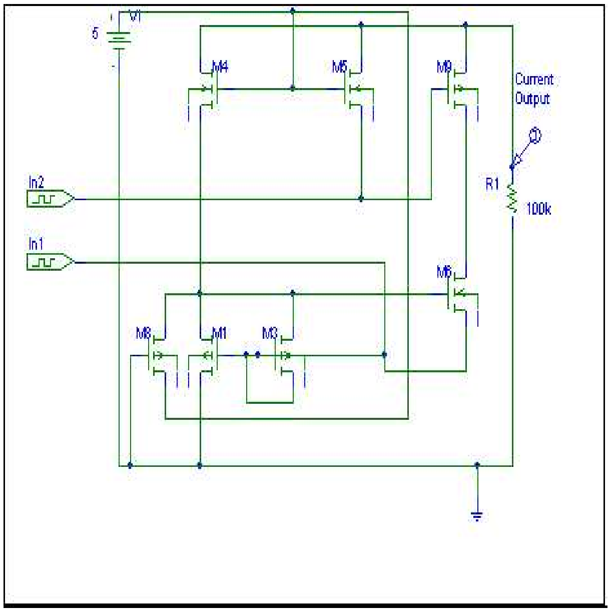


Figure 8. Evolved circuit realizing the ball juggler fuzzy controller. Transistors substrate connections at 5V for PMOS and ground for NMOS.

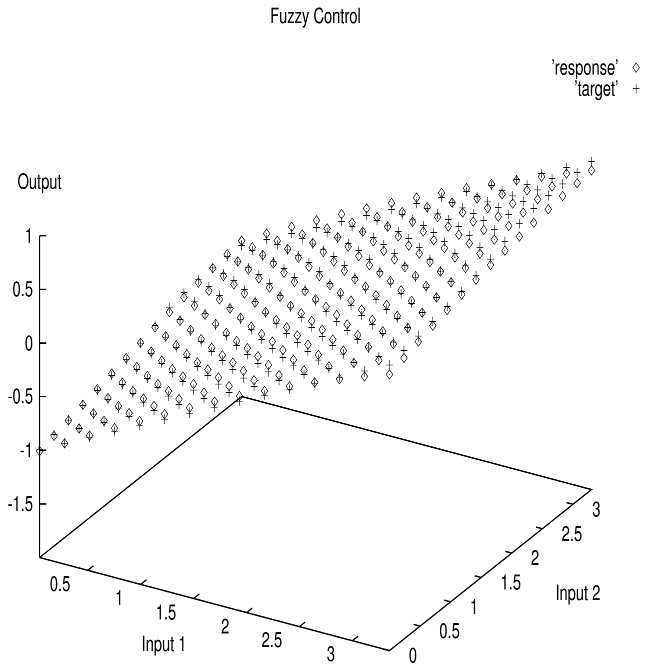


Figure 9. Comparison between response and target for the evolved fuzzy controller

decreasing the transistor sizes by a factor of 10 did not change the circuit response and the deviation from the target; average error of 1.98% and maximum error of 6.96% when decreasing the power supply voltage to 4.75V; average error of 1.94% and maximum error of 6.65% when increasing the power supply voltage to 5.25V; average error of 1.89% and maximum error of 6.3% when decreasing the temperature to 0C; average error of 1.98% and maximum error of 7.2% when increasing the temperature to 55C.

Finally, a different model, (specific for a HP 0.5 MOS fabrication process) led to qualitatively the same result, with slight increase in the error. That error became small again when evolution targeted a circuit in that specific process.

6 Discussion

The results presented here were obtained at the first attempt, with no prior knowledge of the circuit solution, with no optimization in terms of Width and Length of transistor channels, with limited resources (only transistors found in two FPTA cells in FPTA case and imposing a limit of 10 transistors in approximating the fuzzy system). Several methods can be used to reduce the approximation error. One of them is to allow more flexibility in the selection of the points where the signals are applied, and where the output is collected. In this experiment these were considered predetermined, however it is possible to let evolution decide where to interface the circuit with the input/output. Another way to increase the approximation power is to allow more resources, e.g. allow resources from more than 2 cells. This is similar to increasing the approximation power of neural networks when extra neurons are added. The described experiments do not have any parametric adjustment. The width and length of the transistor channel were considered fixed. However, previous results indicate that parametric optimization can produce good adjustments after the topology has been determined [13].

Finally, these results are preliminary and are presented mainly to illustrate some aspects of the application of EHW to synthesis of electronic circuits implementing fuzzy systems. No comparison with any state-of-the-art design tools is made. Although the performance of (computer-assisted) human solutions could exceed the performance of the totally automated solutions illustrated here, there are no tools at present that would allow the complete automated design of the type presented here.

7 Conclusion

One can look at circuits with transistors as functional approximators -how many transistors are needed is only a question of how accurate a function needs to be. Two main results are suggested by the experiments in this paper: that very compact solutions for complete fuzzy systems can be custom designed by evolutionary algorithms, and that specific programmable analog devices could be used as a general purpose platform to rapidly prototype and deploy any fuzzy system. Thus, the programmable analog device solution comes between the high-performance (in speed and power) but very expensive and inflexible full ASIC solution and the less-performance but cheaper and flexible microprocessor solution.

8 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] Koza J. R., Bennett III F. H., Andre, D. , Keane, M. A., *Four Problems for which a Computer Problem Evolved By Genetic Programming is Competitive with Human Performance*, Proc. of 1996 IEEE International Conference on Evolutionary Computation, IEEE Press., pp. 1–10.

[2] Thompson, A. , Layzell , P., Zebulum, R.S., *Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution*, published at: IEEE Transactions on Evolutionary Computation, Special Issue in "Evolvable Hardware", Moshe Sipper (Ed), pp.167–196, V.3, N.3, September, 1999.

[3] A. Stoica. *Toward evolvable hardware chips: experiments with a programmable transistor array.* Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, April 7–9, IEEE Comp Sci. Press, 1999. pp 156–162.

[4] Zebulum, R.S., Pacheco, M.A., Vellasco, M., *Evolutionary Systems Applied to the Synthesis of a CPU Controller*, Second Asia-Pacific Conference on Simulated Evolution and Learning, Canberra, SEAL98, Australia, Novembro, Lecture Notes in Artificial Intelligence, Springer Verlag, 1998.

[5] Thompson, A., Harvey, I., Husbands, P., *Unconstrained Evolution and Hard Consequences*, em Towards Evolvable Hardware: An International Workshop, Lausanne, Swiss, in Towards Evolvable Hardware: The evolutionary engineering approach, pp. 136–165, E. Sanchez and M. Tomassini (eds), Springer-Verlag LNCS 1062, 1996.

[6] Higuchi T. et al. *Real-World Applications of Analog and Digital Evolvable Hardware.* In IEEE Transactions on Evolutionary Computation, Vol.3, No. 3, September 1999. IEEE Press.pp.220–235.

[7] A. Stoica , D. Keymeulen , R. Tawel , C. Salazar-Lazaro, W. Li. *Evolutionary Experiments with a Fine-Grained Reconfigurable Architecture for Analog and Digital CMOS Circuits.* In Proceedings of the First NASA/DoD Worshop on Evolvable Hardware, pages 76–84. IEEE Computer Society Press, 1999.

[8] Proceedings of the Third NASA/DoD Workshop on Evolvable Hardware, D. Keymeulen, A. Stoica, J. Lohn, R. Zebulum (Eds), IEEE Computer Society, Long Beach, CA, USA, 12–14 July, 2001

[9] Proceedings of the 4th International Conference on Evolvable Systems: From Biology to Hardware, ICES2001, Y. Liu, K. Tanaka, M. Iwata, T. Higuchi, M. Yasunaga (Eds), Springer-Verlag, LNCS 2210, Tokyo, Japan, 2001.

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

[11] Adrian Stoica, R. Zebulum, D. Keymeulen, Y. Jin and T. Daud *Evolving Electronic Circuits for Computational Intelligence Hardware ISA'2000,* International ICSC Congress on Intelligent Systems and Applications, Wollongong , Australia, 12-15 December, 2000

[12] http://www.mathworks.com/

[13] Stoica, D. Keymeulen, R. Zebulum, A. Thakoor, T. Daud, G. Klimeck, Y. Jin, R. Tawel and V. Duong, *Evolution of analog circuits on Field Programmable Transistor Arrays Proceedings of the Second NASA/DOD Workshop on Evolvable Hardware,* Palo Alto, CA, July 13-15, 2000, IEEE Computer Society Press, pp. 99–108.

A. Stoica, R. Zebulum,

Received December 6, 2001

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
Phone: (818) 354–2190
E–mail: *adrian.stoica@jpl.nasa.gov*