Evolution of CMOS Circuits in Simulations and Directly in Hardware on a Programmable Chip

Adrian Stoica, Carlos-Salazar Lazaro, Didier Keymeulen[1] and Ken Hayworth

Jet Propulsion Laboratory

California Institute of Technology

Pasadena, CA 91109

Abstract

The paper introduces an approach to automated synthesis of CMOS circuits, based on evolution on a Programmable Transistor Array (PTA). The PTA is a reconfigurable architecture that allows evolutionary synthesis in simulations as well as on a reconfigurable chip implementing the PTA. Thus, the PTA allows, for the first time, analog circuits obtained by evolutionary design to be immediately validated in hardware on the programmable PTA chip. The paper describes a software experiment showing evolutionary synthesis of a circuit with a desired DC characteristic. The hardware implementation of a PTA chip is then described, and the same evolutionary experiment is performed directly on the chip demonstrating circuit synthesis/self-configuration in hardware. The experiment shows that some solutions obtained in simulated evolutions may not be valid when programmed in hardware.

1 INTRODUCTION

In *evolutionary electronics*, the search for an electronic circuit configuration can be made in software and the final solution downloaded or become blueprint for hardware. Alternatively, evolution in hardware (directly on the chip), can speed-up the search for a solution circuit by a few orders of magnitude compared to evolution in software simulations, specially if one simulates large, complex analog circuits. Moreover, since the software simulation relies on models of physical hardware with limited accuracy, a solution evolved in software may behave differently when downloaded in programmable hardware; such mismatches are avoided when evolution takes place directly in hardware.

A variety of circuits have been synthesized through evolutionary means. Koza used Genetic Programming (GP) to grow an “embryonic” circuit to a circuit that satisfies desired requirements (Koza 1996). This approach was used for evolving a variety of circuits, including filters and computational circuits. Koza’s evolutions were performed in simulations, without concern of a physical implementation, but rather as a proof-of-concept that evolution can lead to designs that compete or even exceed in performance the human designs. No analog programmable devices exist that would support the implementation of the resulting design (but, in principle, one can test their validity in circuits built from discrete components, or in an ASIC), and thus intrinsic evolution was not possible. An alternative encoding technique for analog circuit synthesis, which has the advantage of reduced computational load was used in (Lohn, 1998) for automated filter design. On-chip evolution was demonstrated by Thompson (Thompson, 1996) using an FPGA as the programmable device, and a Genetic Algorithm (GA) as the evolutionary mechanism.

In particular, it is interesting to evolve circuits based on CMOS transistors. CMOS Transistors are the elementary building block of the majority of current microelectronics and addressing evolution at this low level allows most flexibility for synthesizing analog, digital and mixed signal designs. Although for many functions it is easier to synthesize based on higher-level dedicated blocks, the lessons learned in synthesizing at this level can be extended to evolution of circuits systems made of other devices and materials/structures. It is interesting to develop dedicated hardware capable of evolution of both analog and digital circuits, directly on the chip; such a platform can be used also for validating circuits obtained in simulations.

This paper proposes a Programmable Transistor Array (PTA) as a platform for experiments in evolutionary synthesis of CMOS electronic circuits. On-chip evolutionary experiments with the PTA are expected to lead to design guides for a true stand-alone evolvable

[1] Member of the Electrotechnical Laboratory, Tsukuba, Ibaraki 305, Japan.

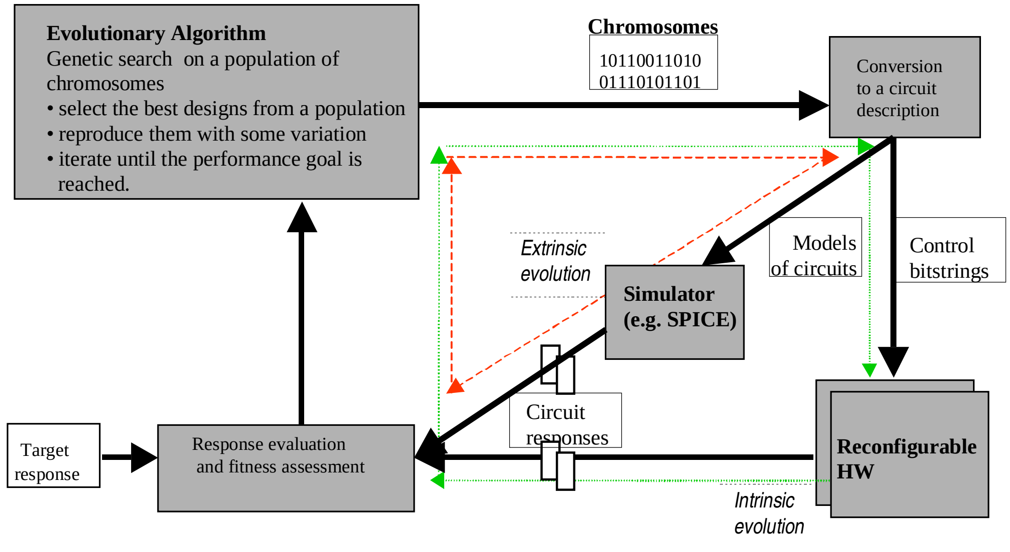

Figure 1: Evolutionary synthesis of electronic hardware

chip. An evolution on a simulated PTA illustrates the feasibility of automated synthesis. A chip was designed and fabricated to validate the results in real hardware.

The paper is organized as follows: Section 2 illustrates the main steps of evolutionary synthesis of electronic circuits and refers to an evolutionary design tool developed around a parallel GA package and a circuit simulator. Section 3 proposes a Programmable Transistor Array as an experimental platform for evolutionary synthesis of CMOS circuits. Section 4 describes a software experiment in which a CMOS circuit with a Gaussian I-V imposed characteristic was synthesized by evolution. Section 5 discusses hardware aspects related to the implementation of the PTA on a 0.5 micron CMOS test chip and describes the evolution directly on the PTA chip. Section 6 compares the software and hardware experiments and presents some lessons learned from the experiments. Section 7 presents related works, while Section 7 presents the conclusions of the paper.

2 EVOLUTIONARY SYNTHESIS OF ELECTRONIC CIRCUITS

The main steps of evolutionary synthesis are illustrated in Figure 1. First, a population of chromosomes is randomly generated. The chromosomes are converted into circuit models (for extrinsic EHW) or control bitstrings downloaded to programmable hardware (intrinsic EHW). Circuit responses are compared against specifications of a target response, and individuals are ranked based on how close they come to satisfying it. Preparing for a new iteration loop, a new population of individuals is generated from the pool of best individuals in the previous generation, some individuals being taken as they were and some being modified by genetic operators, such as chromosome crossover and mutation. The process is repeated for many generations, and results in increasingly better individuals. The process is usually stopped after a number of generations, or when the closeness to the target response has reached a sufficient degree. One or several solutions may be found among the individuals of the last generation.

A variety of Evolutionary Algorithms (including GA and GP) have been used successfully for evolution of circuits. A GA was chosen here because (a) previous work has demonstrated its efficiency in evolutionary circuit

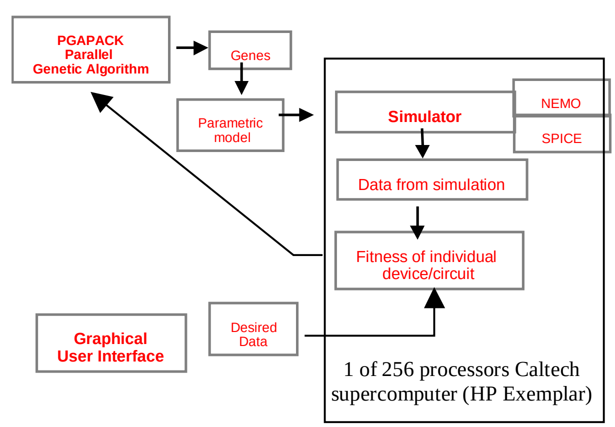

Figure 2: An Evolutionary Design Tool

synthesis, (b) the mechanism is simple to understand and implement, (c) public domain software exists and saves development time, and (d) the focus was on the reconfigurable hardware and not on the reconfiguration mechanism. It is likely that more intelligence can be inserted into the search mechanism.

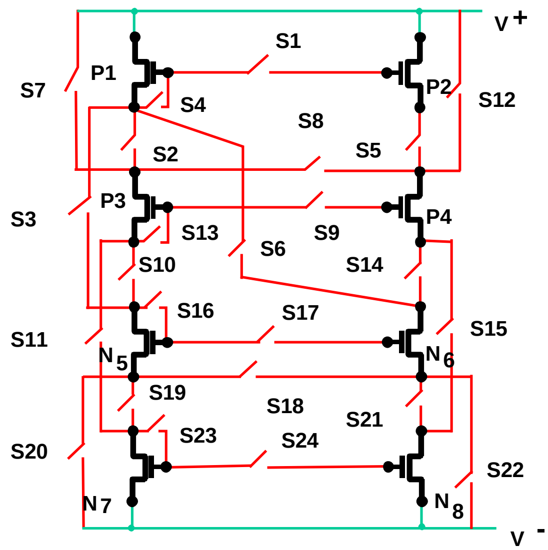

Figure 3: Module of the Programmable Transistor Array

An evolutionary design tool was developed to facilitate experiments in simulated evolution. The tool illustrated in Figure 2 can be used for synthesis and optimization of new devices, circuits, or architectures for reconfigurable hardware. These operations get performed before any hardware gets fabricated. The tool proved very useful in testing architectures of reconfigurable HW and demonstrating evolution on them before the fabrication of a dedicated reconfigurable chip. The tool can also be used in hardware-software co-design. In its current implementation the tool uses the public domain Parallel Genetic Algorithm package PGAPack and a public domain version of SPICE 3F5 as circuit simulator. An interface code links the GA with the simulator where potential designs are evaluated, while a GUI allows easy problem formulation and visualization of results. Each generation the GA produces a new population of binary chromosomes, which get converted into voltages in netlists that describe candidate circuit designs, netlists further simulated by SPICE. More details about the tool are given in (Stoica, 1999b).

3 THE PTA ARCHITECTURE AS A PLATFORM FOR EVOLUTIONARY DESIGN EXPERIMENTS

The PTA idea was introduced in (Stoica, 1996), and expanded in (Stoica, 1998). The proposed PTA is an array of transistors interconnected by 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. The PTA is a modular architecture, in which modules can be cascaded to determine a more complicated circuit topology. Figure 3 illustrates an example of a PTA module consisting of 8 transistors and 24 programmable switches. In this example the transistors P1-P4 are PMOS and N5-N8 are NMOS, and the switch based-connections are in sufficient number to allow a majority of meaningful topologies for the given transistors arrangement, and yet less than the total number of possible connections. Programming the switches On and Off determines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation.

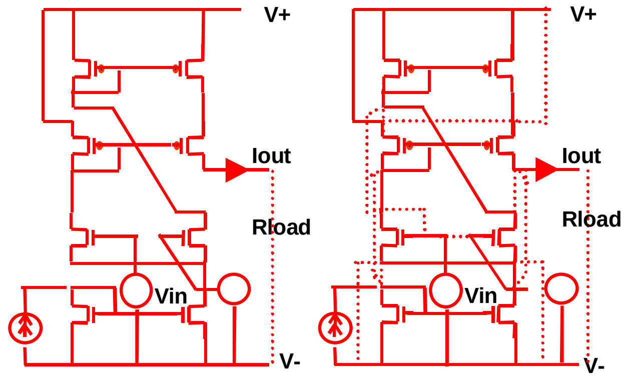

Figure 4: Schematic of a simple circuit implemented on the PTA module (with finite resistance of Off switches as dotted lines on the right figure)

An example of a circuit drawn with this simplification is given in Figure 4. The left drawing illustrates the ideal circuit, the right drawing shows with dotted lines the finite resistance of open switches. A power supply, input signals and a biasing current source have been added. In this implementation four layers of transistors (two PMOS and two NMOS) were chosen, but this can be increased, for example to 6 or 8. More details of the hardware aspects of the PTA chip are detailed in (Stoica, 1999a).

4 AN EXPERIMENT IN EVOLUTIONARY CMOS CIRCUIT SYNTHESIS ON A SIMULATED PTA

The following experiment illustrates the evolutionary synthesis of a computational circuit. The goal of evolution was to synthesize a circuit which exhibits a Gaussian I-V (current-voltage) input-output characteristic. In a previous experiment (Stoica, 1997) the circuit topology was fixed and the search search/optimization addressed transistor parameters (channel length and width); such evolution proved quite simple. The search for a topology turned out to be a much harder problem and several architectures were unsuccessfully attempted before the PTA was conceived. In the PTA case, the transistor parameters were kept fixed and the search was performed for the 24 binary parameters characterizing switches status. The fitness function was specified as a weighted combination of parameters x1,…,x7 in Figure 5.

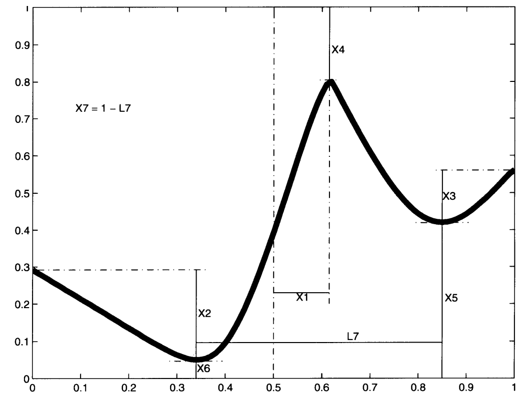

Figure 5: Parameters used for the specification of the fitness function. Fitness =f(x1,…x7)

The evolution was simulated on a Caltech supercomputer (HP-Exemplar), using the Evolutionary Design Tool. Successful evolution was demonstrated on multiple runs with populations between 50 and 512, evolving for 50 or 100 generations. The execution time depends on the above variables and on the number of processors used (usually 64 out of the 256 available), averaging around 20 minutes (the same evolutions took about 2 days on a SUN SPARC 10). In some runs the solution circuit shown in

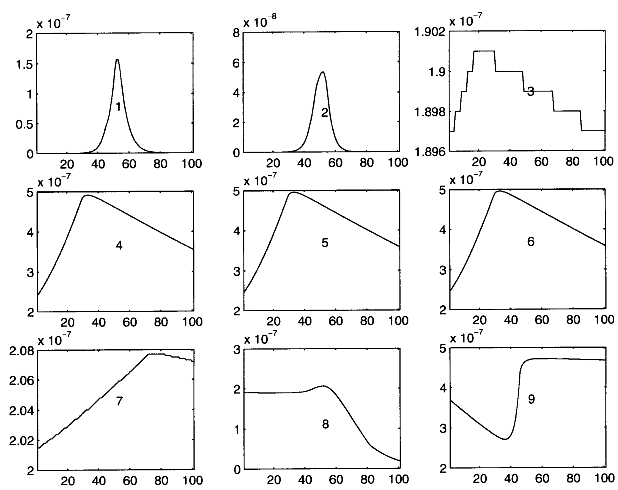

Figure 6: Best circuit responses in a simulated evolution

Figure 4 (human designed) was rediscovered by evolution.

Other solutions found include the circuits illustrated in Figure 7, which produce the first two responses in Figure 6; some other responses from the same generation are illustrated in Figure 6 for comparison. It is interesting to analyze in more detail the unusual solutions found by evolution. Circuits like those illustrated in Figure 7 resulted from evolutionary synthesis are very similar

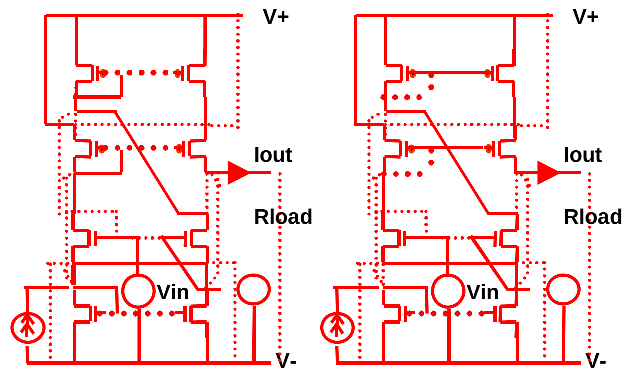

Figure 7: Circuits obtained by evolution; their design is unusual for common practice

(under certain test conditions) to that of the circuit shown in Figure 4. Thicker dotted lines show connections that existed in the circuit in Figure 4, but are missing in the circuits in Figure 7. As it is easy to observe these circuits are outside normal design practices, e.g., the transistors P2, P4 and N8 on the left circuit in Figure 7 have floating gates. The reality is that the switches have a big, but finite, resistance in the Off state (~MOhms or GOhms) and a non-zero resistance/impedance in the On state (~ tens of Ohms). *One observation from here is that while the effects of non-perfect switches may be negligible in a first approximation for many digital circuits, such effects may fundamentally affect analog programmable circuits.*

5 HARDWARE IMPLEMENTATION AND EVOLUTION ON THE PTA CHIP

Successful evolution on simulated PTA encouraged the development of a test chip implementing the PTA architecture. The chip would offer an estimate on how reliable is evolution on software models. More importantly, evolution of the circuit directly on the chip becomes possible, and at an expected accelerated pace of over 100 times compared to the simulation (estimated ~5 seconds compared to ~20 minutes on the supercomputer for the experiment described). As in the experiments performed in simulations the size of transistors was fixed. The programmable switches were implemented with transistors, acting as simple T-gate switches. One should mention that the analog gradual switches act in circuit evolution very much like resistive weights in a neural network implementation.

Each chip implements one PTA module. Issues related to chip expandability are treated in (Stoica, 1999a). The chip was fabricated as a Tiny Chip through MOSIS, using 0.5-micron CMOS technology. The test board with four chips mounted on it is illustrated in Figure 8.

The same evolutionary experiment, aiming at the synthesis of a DC circuit with a Gaussian response, was performed in hardware on the PTA chips, (the GA was implemented in LabView). Four chips were programmed in parallel with bit-string configurations corresponding to

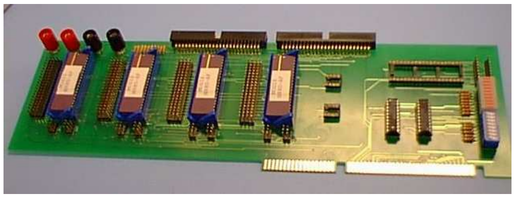

Figure 8: A test board with four PTA Chips

four individuals of a population of 100; then, the next four were programmed, and so on until all 100 in one generation were tested. As in simulation, evolution led to “Gaussian” circuit solutions within 200-300 generations in 4.5 min using the four PTA chips in parallel.

The response of four mutants is illustrated in the screen capture shown in Figure 9 (LabView display of the signals captured by the data acquisition boards). Notice the “mutations” in the genetic code of the solutions obtained by evolution (vertical chromosomes R24 – R1 reading from top to bottom – these correspond to switches S24 - S1 in Figure 4) compared with the human-designed solution of the Gaussian circuit.

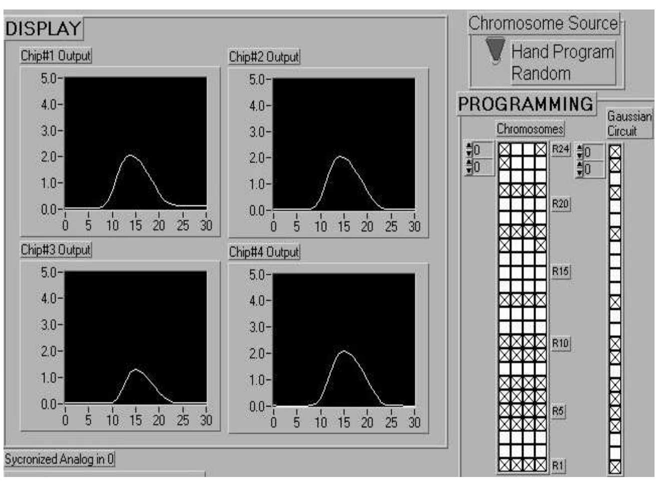

Figure 9: The “Gaussian” response of four “mutants” and their “genetic code” compared to the generic solution

6 LESSONS LEARNED

1. An interesting observation was that, *other than the “correct” human-designed solution rediscovered by evolution*, the solutions evolved on the PTA chip are different than those evolved in simulations. (At least the few of them that were tested; additional circuits solutions may exist that lead to the same response both in the Spice simulation and programmed on the chip). It would thus appear that different effects are exploited to lead to solutions in the model and in the silicon implementation. More precisely, the circuit solutions evolved in simulations (with Spice resistive models for On/Off switches) did not prove to be solutions when programmed on the PTA chip, and vice-versa, the configuration solutions evolved directly on the PTA chip (e.g. those in Figure 10) did not simulate as Gaussians. (Further experiments using more accurate models of the PTA silicon implementation are in progress). Thus, it appears justifiable to express reserve on the validity of a solution obtained by “extrinsic” evolution of analog circuits until is verified in hardware (at least for particular PTA discussed here and with the limited accuracy model used).
2. The original intent was to speed-up the evolution from ~20 minutes on the supercomputer to about 5 seconds on the PTA chip (reducing the evaluation of a circuit to ~0.25ms). At this moment, LabView (running on a 300 MHz Pentium) presents some communication bottlenecks that only allowed reaching about the same evolution time as on supercomputer. In the quest for faster circuit evaluation on the chip a further limitation was however noticed, ignored when running Spice DC analysis only: the circuits have own frequency response and there are limits of possible speed-up for which the response is the same as in DC/low frequency. The output of the Gaussian circuit on the PTA starts attenuation when the input ramp signals exceed 1kHz, meaning that no more than 1000 circuits per second could be reliably evaluated. Even though this may be an artifact of the particular PTA design and load choice, it appears natural that evaluating the circuits at a different frequency than that of intended functioning may introduce errors. Evaluation in parallel is an alternative speed-up technique, and at least in the experiments with the PTA chips no significant differences were noted between the instantiations of the same circuit on different chips.

7 RELATED WORK

We started work on evolution of circuits reconfigurable at transistor level in 1996 (Stoica, 1996), (Stoica, 1998) with the first results published in 1997 (Stoica, 1997). Our work focused from the beginning on evolution of CMOS circuits. The same goal of exploring evolution at transistor level can be found in more recent publications from Sussex University, in particular the work of Layzell (Layzell, 1998). The main distinction comes from the fact that our work focuses on CMOS only. This has some important consequences, for example the fact that the small leakage current through OFF switches can be sufficient for CMOS but will not affect bipolars, hence some mutant solutions appear only in CMOS for the described topology.

8 CONCLUSION

Automatic synthesis/self -configuration of analog circuits was demonstrated on an experimental CMOS chip implementing a Programmable Transistor Array architecture proposed as reconfigurable hardware platform for evolutionary synthesis experiments. The experiments bring further testimony to the feasibility of using evolutionary algorithms for automated synthesis of electronic circuits. A comparison of the simulated and on-chip experiments indicates limitations of the extrinsic evolutionary method; the solutions obtained in simulations were not validated when programmed on the chip. However, different solutions have evolved on the chip, and proved robust when transferred to other chips from the same fabrication lot.

Acknowledgements

The research 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

J. Koza, F.H. Bennett, D. Andre, and M.A Keane (1996). Automated WYWIWYG design of both the topology and component values of analog electrical circuits using genetic programming. In *Proceedings of Genetic Programming Conference*, pp. 28-31. AAAI Press.

Layzell, P. (1998). A New Research tool for Intrinsic Hardware Evolution. In M. Sipper, D. Mange and A. Perez-Uribe (Eds.) *Evolvable Systems: From Biology to Hardware*. ICES'98, Lausanne, Switzerland. Springer-Verlag Lecture Notes in Computer Science, Berlin.

J. Lohn, J. and S. Colombano (1998). Automated Analog Circuit Synthesis using a linear representation. In M. Sipper, D. Mange and A. Perez-Uribe (Eds.) *Evolvable Systems: From Biology to Hardware*, pp. 125-133. Springer-Verlag Lecture Notes in Computer Science Berlin.

Stoica, A. (1996). *Reconfigurable Transistor Array for Evolvable Hardware*, July 26. Caltech/JPL Novel Technology Report.

Stoica, A (1997). On hardware evolvability and levels of granularity. In *International Conference on Intelligent Systems and Semiotics*. Sept. NIST Gaithersburg, VA.

Stoica, A. and Salazar-Lazaro C. (1998). *Evolutionary technique for automated synthesis of electronic circuits*, Sept. 4. Caltech/JPL Novel Technology Report.

Stoica, A (1999a). Toward evolvable hardware chips: experiments with a programmable transistor array. In *Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*. April 7-9, Granada, Spain.

Stoica, A. Klimeck, G. Salazar-Lazaro, C. Keymeulen, D. and Thakoor, A (1999b). Evolutionary design of electronic devices and circuits. In *Congress on Evolutionary Computation*. July 6-9, Washington, DC. IEEE Press.

A. Thompson (1996). An evolved circuit, intrinsic in silicon, entwined in physics. In *International Conference on Evolvable Systems*, pp 390-405. Springer-Verlag Lecture Notes in Computer Science.