Evolvable Hardware: From On-Chip Circuit Synthesis to Evolvable Space Systems

Adrian Stoica
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
818-354-2190
adrian.stoica@jpl.nasa.gov

Abstract—Evolvable Hardware (EHW) refers to HW design and self-reconfiguration using evolutionary/genetic mechanisms. The paper overviews some key concepts of EHW, comments on selected applications, and presents a perspective on the development of the field. A fine-grained Field Programmable Transistor Array (FPTA) architecture for reconfigurable hardware is presented as an example of an initial effort toward evolution-oriented devices. Evolutionary experiments in simulations and with a FPTA chip in-the-loop demonstrate automatic synthesis of electronic circuits. Unconventional circuits, for which there are no textbook design guidelines, are particularly appealing to evolvable hardware. To illustrate this situation, one demonstrates here the evolution of circuits implementing parametrical connectives for fuzzy logics. In addition to synthesizing circuits for new functions, evolvable hardware can be used to preserve existing functions, and achieve fault-tolerance determining circuit configurations that circumvent the faults. These characteristics are extremely important for enabling spacecraft to survive harsh environments and to have long life. Expanding reconfiguration to other types of spacecraft hardware (i.e. optics, MEMS, etc) could lead to evolvable space systems.

1. INTRODUCTION

The application of evolution-inspired formalisms to hardware design and self-configuration lead 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 overviews some key concepts of EHW, comments on selected applications, and presents a perspective on the development of the field. It then describes an effort toward building evolution-oriented devices and an evolvable system on a chip. A Field Programmable Transistor Array architecture is used as the experimental platform for evolutionary experiments. The platform is quite flexible and supports implementation of both analog and digital circuits. While previous work [1] [2] illustrated implementation of several conventional building blocks for electronic circuits such as logical gates, transconductance amplifier, filters, gaussian neuron, etc. 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, providing a perspective on the evolution of the field. Section 3 overviews some important evolutionary experiments and applications of evolvable hardware. Section 4 presents an evolution-oriented architecture based on the concept of 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. Section 6 presents considerations related to the application of evolvable hardware to space systems.

2. EVOLVABLE HARDWARE: FROM ROOTS TO BUDS

The main idea of evolutionary/genetic algorithms is inspired from 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 in meeting the design specifications) being selected for being further investigated. The offspring of this elite in which pairs of parents were randomly selected for "mating" combine genetic material from two parents (or inherit from one parent only), and may 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 roots of EHW can be traced to the 1960s, when Evolutionary Strategies were invented to perform continuous

parameter optimization problems for a variety of designs and laboratory experiments. In about the same time Evolutionary Programming was conceived on similar principles to evolve finite state machines, while Genetic Algorithms were introduced as a model of adaptation. Moving upwards on the schematic illustration in Figure 1, the next step toward EHW was the idea of evolving computer programs coming from Genetic Programming.

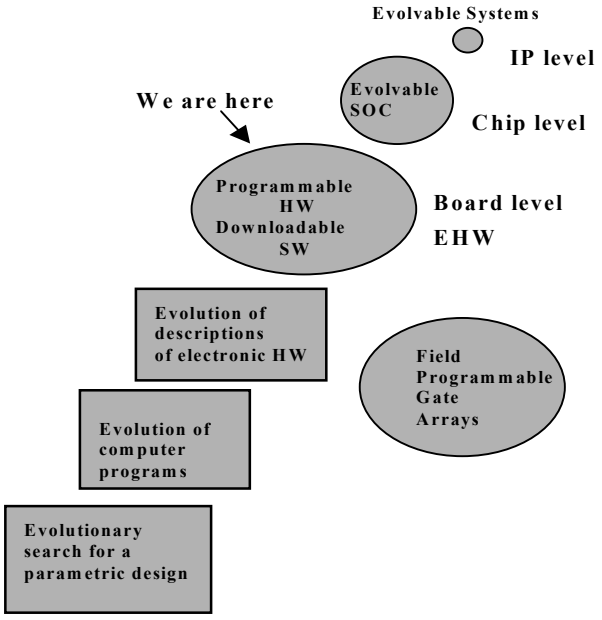

Figure 1 Evolutionary path for the evolvable hardware field: from design optimization to hardware IP cores for evolvable systems.

The concept of evolvable hardware was born partly inspired by the above search/optimization/adaptation mechanisms and partly by the availability of reconfigurable devices such as Field Programmable Gate Arrays (FPGA). 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). Currently evolutionary platforms are board level, including programmable hardware that is reconfigured under the control of configurations bits determined by the evolutionary algorithms running in software. It is likely that in the next 1-3 years more platforms will integrate the reconfigurable hardware and the reconfiguration mechanism in an evolvable system on a chip (SOC) solution. Finally, the path leads to the Intellectual Property (IP) level and EHW solutions will become an integrated component in a variety of systems that will thus have an evolvable feature.

EHW has the potential to bring an important contribution to several domains, from more conventional ones like communications, household appliances and Internet to more exotic ones like micro/nano-scale systems and biological/artificial hybrids. Adaptive/evolvable hardware have great potential for commercial applications in communications. Several areas include data compression, reconfigurable antennas, adaptive signal processing. For example, evolutionary techniques were shown to outperform current best techniques in image compression [3]. The price paid is an increase in computation since adaptive parameter changes in the compression algorithm need to be made for each individual image or set of images. However doing it in hardware may provide the sufficient speed-up to make the technique real-time and economically efficient. Evolutionary algorithms have shown excellent potential in designing new antenna configurations and controlling reconfigurable antennas. In adaptive equalization for radio communication evolvable hardware could dynamically perform adjustments to compensate for changing transmission path characteristics. It would thus maintain the system transfer function characteristics within specified limits by modifying circuit parameters such as resistance, inductance, or capacitance.

There is a good potential for commercial applications of Internet adaptive devices. Reconfigurable hardware is playing an increasing role in the Internet infrastructure, allowing the possibility of hardware adaptation. Adaptive reconfiguration needs to be done automatically, and EHW could help. Not only can Internet devices be involved in local individual evolution but large populations can interconnect through Internet. Evolution can run at Internet scale searching for optimal configuration solutions. Once a solution is found, it can be rapidly shared. In fact evolution involving large numbers of computers running software simulations has already been experimented with.

An example of a micro/nano-scale system which can be enhanced by evolvable hardware are future miniature complex sensing, diagnosis and monitoring system. The future "Lab on a Chip" would perform adaptive detection of chemical and biological materials. Example of applications include food preservation, virus and bacteria detection, and adaptive dosage of medicine. The design and adaptation of biological/artificial hybrids could be influence by evolvable hardware. This includes system configuration and adaptation of interfaces between the biological and artificial materials, etc.

Figure 2 illustrates the main steps of evolutionary design of electronic circuits. Each candidate circuit design is associated 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 like SPICE) and produces responses that are compared against specifications.

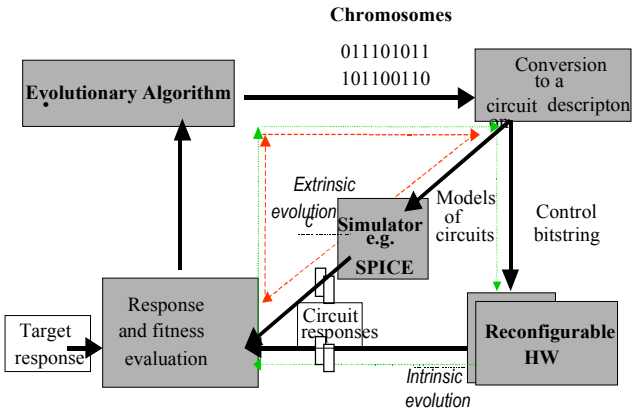

Figure 2 Evolutionary synthesis of electronic circuits

A solution determined by extrinsic evolution may eventually be downloaded or become blueprint for hardware. In intrinsic evolution the chromosomes are converted into control bitstrings, which are downloaded to program the reconfigurable device (e.g. a Field Programmable Gate Array). 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 generation of a new population of individuals from the pool of the best individuals in the previous generation. Here, some individuals are taken as they were and some are 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. 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 [4]. On-chip evolution was demonstrated by Thompson [5] using an FPGA as the programmable device, and a Genetic Algorithm (GA) as the evolutionary mechanism. More details on current work in evolvable hardware can be found in [6] and [7]. Evolutions of analog circuits reported in [4] 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, or even exceed in performance those of humans. Current programmable analog devices are very limited in capabilities and do not support the implementation of the resulted design (but, in principle, one can test their validity in circuits built from discrete components, or in an ASIC). More recently, evolutionary experiments were performed on COTS Field Programmable Analog Arrays [18] and custom-designed ASIC [11]. Figure 3 illustrates a plethora of devices platforms that were used for EHW experiments. The hardware devices include FPGAs, FPAA, Analog ASICs, etc.

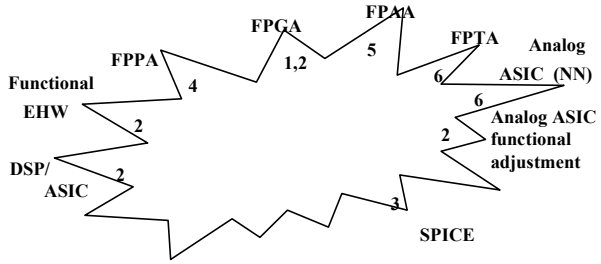

Figure 3 Multitude of platforms for EHW experiments

4. BUILDING AN EVOLVABLE SYSTEM-ON-A-CHIP

Current 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. This precludes searches directly in hardware and requires evolving on hardware models. Such models require evaluation with circuit simulators such as SPICE; the simulators 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 offers a big advantage in evaluation time for a circuit; the time for evaluation is determined by the goal function. For example, considering an A/D converter operating at a 100 kHz sampling rate, its electronic response is available within 10 microseconds, compared to (an over-optimistic) 1 second on a fast computer running Spice; this advantage increases with the complexity of the circuits. In this case the 10^5 speedup would allow evaluations of populations of millions of individuals in seconds instead of days.

Increasingly more complex Field Programmable Devices (FPGA, FPAA, etc) offer powerful solutions to applications in digital signal processing, programmable interfaces, filtering, etc. However, for efficiency in EHW applications, future devices would benefit from implementing evolution-oriented reconfigurable architectures (EORA). One of the most important features for EORA relates to the *granularity* of the programmable chip. FPAA offer only coarse granularity which is a clear limitation; FPGAs are offered both in versions with coarse grained and fine grained architectures (going to gate level as the lowest level of granularity). From the EHW perspective, it is interesting to have *programmable granularity*, allowing the sampling of novel architectures together with the possibility of implementing standard ones. The optimal choice of elementary block type and granularity is task dependent. At least for experimental work in EHW, it appears a good choice to build reconfigurable hardware based on elements

of the lowest level of granularity. Virtual higher-level building blocks can be considered by imposing programming constraints. Ideally, the "virtual blocks" for evolution should be automatically defined/clustered during evolution. EORA should be *transparent architectures*, allowing the analysis and simulation of the evolved circuits. They should also be robust enough not to be *damaged* by any configuration existent in the search space, potentially sampled by evolution. Finally EORA should allow evolution of both analog and digital functions.

An evolvable system-on-a-chip architecture is presented in Figure 4. It includes a Field Programmable Transistor Array and a Genetic Processor. The idea of a field programmable transistor array was introduced in [8] as a first step toward EORA. 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" interconnected by other transistors that act as signal passing devices (gray-level switches), with islands of RC resources in between.

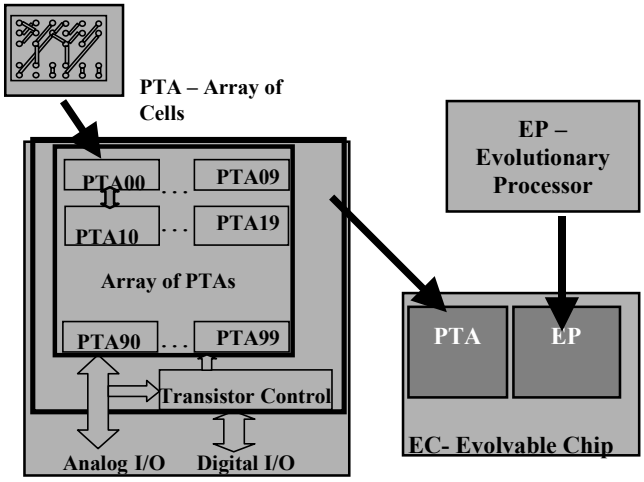

Figure 4 An evolvable SOC

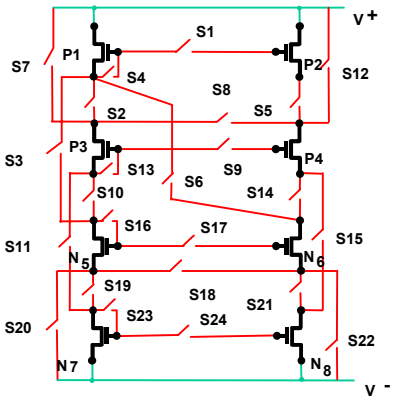

Figure 5. Module of the Programmable Transistor Array

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. Figure 5 illustrates a FPTA cell consisting of 8 transistors and 24 programmable switches. 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.

5. Evolving Reconfigurable Circuits for Fuzzy Logics

This section illustrates the evolutionary design of circuits for multi-valued 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. Frank's parametric T-norms and T-conorms (also refered to as fundamental T-norms/conorms in [9]) were the selected choice for modeling the logical connectives. The family of Frank T-norms is given by

$$T_s(x,y)=\begin{cases} MIN(x,y) & if\ (s=0) \\ x\cdot y & if\ (s=1) \\ \log_s\left(1+\dfrac{(s^x-1)(s^y-1)}{s-1}\right) & if\ ((0<s<\infty),s\neq 1) \\ MIN(1,x+y) & if\ (s=\infty) \end{cases} \quad (1)$$

The family of Frank T-conorms is given by

$$S_s(x,y)=\begin{cases} MAX(x,y) & if\ (s=0) \\ x+y-x\cdot y & if\ (s=1) \\ 1-\log_s\left(1+\dfrac{(s^{1-x}-1)(s^{1-y}-1)}{s-1}\right) & if\ ((0<s<\infty),s\neq 1) \\ MIN(1,x+y) & if\ (s=\infty) \end{cases} \quad (2)$$

Electronic circuits can be used in implementations of multi-valued logic computations or in implementing fuzzy S-T neurons. One interesting application made possible by this implementation is to select the most appropriate s-parameter for the application at hand. Examples of the influence of

various t-noms and s-norms in control applications can be found in [10] [11] and for learning in fuzzy neurons in [12].

The following preliminary results illustrate the possibility of evolving circuits that implement T and S for various values of the parameter s. The circuits are powered at 5V and the signal excursion is chosen between 1V (for logical level "0") and 4V (for logical level "1"). Intermediary values are 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.

Figures 6,7,8 show the response of circuits targeting the implementation of fundamental T-norms for s=0, s=1, and s=100 respectively. The circuit for T-norm with s=100 is shown mapped on 2 FPTA cells in Figure 9. Figure 10 shows the response of the circuit implementing the fundamental s-norm for s=100. Figure 11 shows the diagonal cut for the same S-norm. All these responses were for circuits evolved in software; for comparison the response of a circuit evolved in hardware (for s=100) is shown in Figure 12.

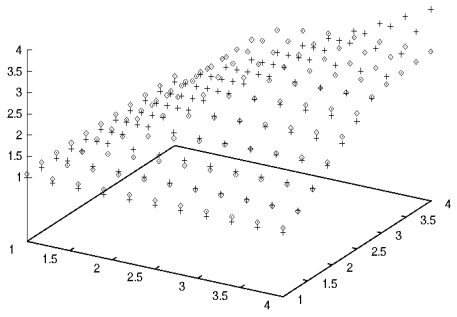

Figure 6 Response of a circuit implementing the fundamental T-norm for s=0 (o). Target characteristic shown with (+).

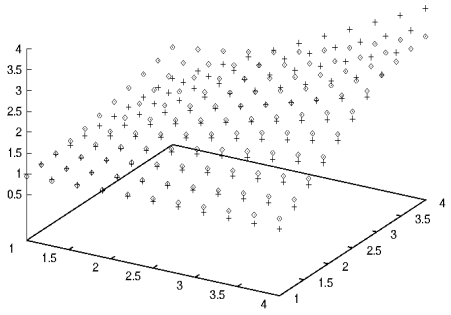

Figure 7 Response of a circuit implementing the fundamental T-norm for s=1 (o). Target characteristic shown with (+).

The results presented here are a first attempt at evolving these type of circuits. Their purpose is to illustrate what you can obtain in a rapid evolution, with no prior knowledge on the circuit solution, with no optimization in terms of Width and Length (W,L) of transistor channels, with limited resources (only those found in 2 FPTA cells). One limitation is the approximation error, ranging 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 factors can contribute to reducing the approximation error. One of them is to allow more flexibility in the selection of the points where the inputs 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.

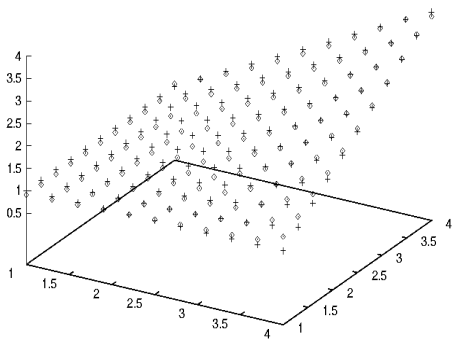

Figure 8 Response of a circuit implementing the fundamental T-norm for s=100 (o). Target characteristic shown with (+).

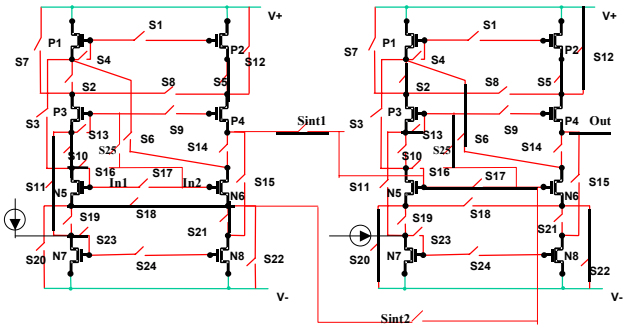

Figure 9 Evolved circuit implementing the fundamental T-norm for s=100 (with the response in Figure c).

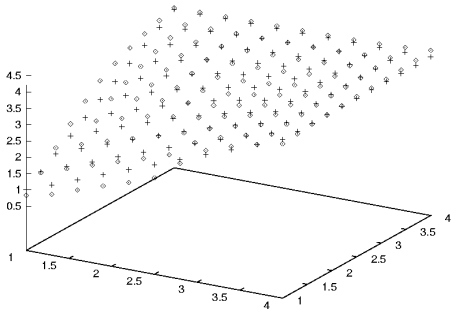

Figure 10 Response of a circuit implementing the fundamental S-norm for s=100 (o). Target characteristic shown with (+).

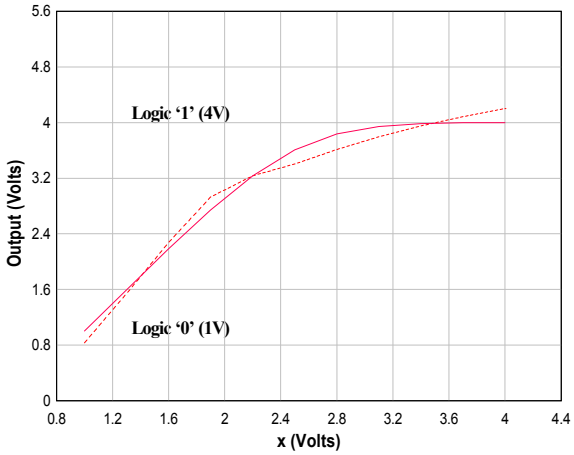

Figure 11 Diagonal cut for the response in Figure e. Circuit implementing the fundamental s-norm for s=100. Target characteristic shown with full line.

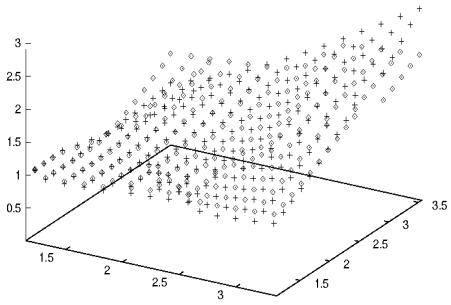

Figure 12 Response of a hardware-evolved circuit implementing the fundamental T-norm for s=100 (o). Target characteristic shown with (+).

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]. This will also be possible in hardware since the new version of the chip will allow switch-selectable transistors with different W/L in the same cell.

5. TOWARD EVOLVABLE SPACE SYSTEMS

EHW can to bring two key benefits to spacecraft survivability. Firstly, EHW can help preserving existing functions, in conditions where hardware is subject to faults, aging, temperature drifts and radiation, etc. The environmental conditions, in particular the extreme temperatures and radiation effects can have catastrophic impacts on the spacecraft. Interstellar missions or extended missions to other planets in our solar system, with lifetimes in excess of 100 years, put great challenges on the on-board electronics. Secondly, new functions can be generated (more precisely new hardware configurations can be synthesized to provide required functionality) when needed.

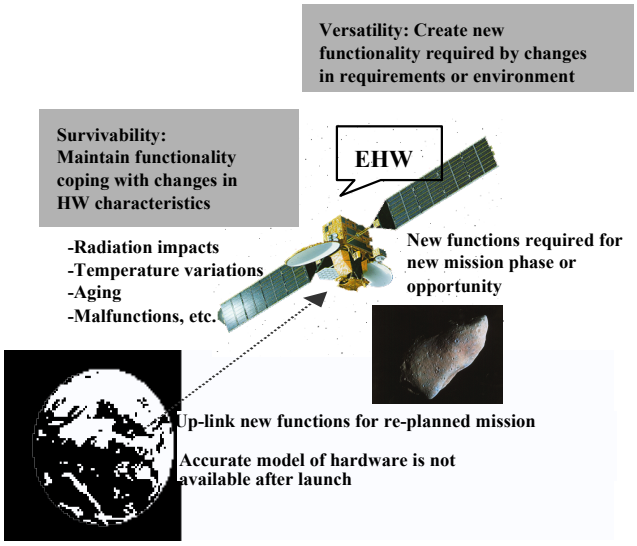

Figure 13 EHW can contribute to increase spacecraft survivability and flexibility

Previous sections of this paper illustrated how EHW can be used to automatically synthesize circuits implementing new functions. In the remainder of this section a fault-tolerance experiment presented in detail in [ ] is summarized. The experiment shows how EHW can recover functionality after being lost due to faults, by finding new circuit configurations that circumvent the faults. In the experiment, which targeted a circuit implementing a gaussian input-output DC response, the performance of the chip continued to be monitored using the fitness function even after a solution was determined.

When the performance decreased below a certain threshold (e.g. when a fault was injected), the evolution process restarted the search for a new circuit configuration, taking into account the previous circuit configurations in the population. Faults were injecting by disconnecting external wires between FPTAs. At that time a lowering of performance but not a complete failure was observed. The reason for the graceful degradation is that the population of circuits obtained by the evolution process contains mutants insensitive to faults having the same phenotypic effect as a genetic mutation. When the fault was injected the GA restarted with the population of its last run, which included the currently affected by fault and some of its mutants. The faulty part became just another component to be used: the evolutionary algorithm did not "know" that the part was supposed to do something else. While starting with a random population took about the same time as finding a solution in the first place (not shown), starting with the last available population led to recovery in about 1/3 of the time while the circuit performance recovered to 90%.

6. CONCLUSION

This paper presented some highlights in the history of the field of evolvable hardware and presents a possible path for its evolution in the future. It presented an effort of building evolution-oriented devices and demonstrates how electronic circuits can be automatically be synthesized, on-the-chip, to produce a desired functionality. It illustrates the simplicity with which evolvable hardware can be used to design unconventional circuits such as combinatorial circuits for fuzzy logics. It addresses the benefits evolvable hardware may bring in flexibility and survivability of future space hardware.

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." In [6].

[2] R. Zebulum, A. Stoica and D. Keymeulen. A Flexible Model of a CMOS Field Programmable Transistor Array Targeted for Hardware Evolution. Third Int. Conference on Evolvable Systems: From Biology to Hardware (ICES2000)

[3] A. Fukunaga and A. Stechert. Evolving nonlinear predictive models for lossless image compression with genetic programming GP-98

[4] 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", *Proceedings of Genetic Programming Conference*, Stanford, CA , pp. 28-31, 1996

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

[6] M. Sipper, D. Mange, A. Perez-Uribe (Eds.) Evolvable Systems: From Biology To Hardware, *Proc. of the Second International Conference, ICES 98*, Lausanne, Switzerland, Springer-Verlag Lecture Notes in Computer Science, 1998.

[7] A. Stoica, D. Keymeulen and J. Lohn (Eds.) *Proc. of the First NASA/DoD Workshop on Evolvable Hardware*, July 19-21, 1999, Pasadena, CA IEEE Computer Society Press.

[8] Stoica, A. 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.

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

[10]M.M. Gupta and J. Qi, Design of fuzzy logic controllers based on generalized t-operators. Fuzzy Sets and Systems, Vol. 40 pp 473-389

[11] [10]M.M. Gupta and J. Qi, Theory of t-norms and fuzzy inference methods. Fuzzy Sets and Systems, Vol. 40, 431-450

[12] Stoica, A 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

[13] Stoica, A. On hardware evolvability and levels of granularity. Proc. of the International Conference "Intelligent Systems and Semiotics 97: A Learning Perspective, NIST, Gaithersburg, MD, Sept. 22-25, 1997

[14] A. Stoica, D. Keymeulen, V. Duong and C. Lazaro. Automatic Synthesis and Fault-Tolerant Experiments on an Evolvable Hardware Platform. In R. Profet et al.(eds.), Proc. of IEEE Aerospace Conf., March 18-25, 2000, Big Sky, MT, IEEE Press