EHWPACK: AN EVOLVABLE HARDWARE ENVIRONMENT USING THE SPICE SIMULATOR AND THE FIELD PROGRAMMABLE TRANSISTOR ARRAY

DIDIER KEYMEUL EN [*]
RICARDO ZEBULUM [*]
YILI JIN [±]
[±] California Institute of Technology
Pasadena, CA 91125

GERHARD KLIMECK [*]
ADRIAN STOICA [*]
CARLOS-SALAZAR LAZARO [*]
[*] Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91101

ABSTRACT
This paper describes the EHWPack development system, a tool that performs the evolutionary synthesis of electronic circuits, using the SPICE simulator and the Field Programmable Transistor Array hardware (FPTA) developed at JPL. EHWPack integrates free and commercial software packages such as PGAPack for the evolutionary algorithm, SPICE for the circuit evaluation, Tcl-Tk for the graphic interface, and LabView for the hardware evaluation. The paper investigates the performance of the tool in two typical problems of EHW: evolutionary synthesis of a Gaussian computational function and the evolution of a band-pass filter.

1. INTRODUCTION

This paper presents the tool EHWPack, a multi-tasking parallel software package targeted for Evolvable Hardware (EHW) experiments. EHW involves the investigation of the automatic synthesis of electronic circuits through evolutionary systems (Thompson, 1998). This field of research constitutes a new approach for automatic circuit design, where the process of circuit conception is interpreted as a search task. The use of evolutionary systems as a search technique provides an efficient way to sample the large search space associated with problems of electronic circuit synthesis. EHW can afford the synthesis of novel circuits that are comparable to the human designed ones in terms of such aspects as size, noise, power consumption and others (Koza, 1998). Nevertheless, EHW experiments need powerful computer resources and evolution-oriented programmable devices.

In order to handle the above issue, the EHWPack tool has been developed. The EHWPack is a distributed parallel software-hardware environment for evolutionary circuit design. It has been developed to facilitate experiments, both in simulated as well as hardware evolution, using SPICE circuit simulator and a Field Programmable Transistors Array (FPTA) respectively (Stoica, 1999b). The tool is used for the evolutionary synthesis, optimization and on-line adaptation (Keymeulen, 2000) of electronic circuits in *extrinsic*, *intrinsic* and *mixtrinsic* mode. Furthermore, it has also been used as a test-bed for new architectures of reconfigurable hardware (Stoica 1999b) and nano-electronic devices (Stoica, 1999c).

Although a variety of evolution-based software environments have successfully been developed for evolutionary designs (Levine, 1994) (Heitkötter, 1997) (van Lent, 1999) (Wall, 1999) (Bennett, 1999), a tool like EHWPack was needed due to a number of factors: (a) the currently available evolutionary software packages implement general-purpose genetic algorithms running on various workstations and under different operating systems, but a dedicated genetic algorithm is needed for circuit design; (b) public domain software is available for genetic algorithm, circuit simulation, graphical interface, PC-board control and network communication, however no software integrates all these components into a single environment; (c) the genetic algorithm for circuit design using both software simulation and hardware implementation must be evaluated on a single platform; (d) the tool must be user friendly and transparent, such that experimentalists (not necessarily experts in software simulation on supercomputer) located at different sites can use it; and finally, (e) the evolutionary design of portable circuits can only be achieved by integrating results from software simulation and hardware execution in the same experimental environment (Stoica, 2000).

The paper is organized as follows: Section 2 describes the EHWpack environment. Section 3 presents an analysis of the EHWPack performance in terms of speed to evolve circuits. The conclusions are discussed in section 4.

2. EHWPACK ENVIRONMENT

EHWPack implements the three main steps of an evolutionary design of electronic circuits. In the first step, a population of chromosomes is randomly generated and the chromosomes are converted into circuit models (voltages in netlists for extrinsic EHW) or control bit strings, downloaded to programmable hardware (FPTA devices for intrinsic EHW). In a second step the Netlists are further simulated by SPICE while an analog signal programmed by LabView is injected at the input of the hardware device. In the third step, circuit responses are compared against specifications of a target response, and individuals are ranked based on how close they come to satisfying it. In the fourth step, 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 are taken as they were and some are modified by genetic operators, such as chromosome crossover and mutation. The process is repeated for many generations, which results in increasingly better individuals (Stoica, 1999b). The process is usually stopped after a number of generations, or when the closeness to the target response has been reached to a sufficient degree. One or several solutions may be found among the individuals of the last generation.

In its current implementation, the tool uses the public domain Parallel Genetic Algorithm package, PGAPack, (Levine, 1994) a public domain version of SPICE 3F5 as circuit simulator and a FPTA (Stoica, 1999b) evolvable hardware test bed built around LabView. The FPTA is an array of transistors interconnected by programmable switches and will be discussed in details later.

An interface code links the GA with the circuit simulator and with the hardware where potential designs are evaluated, while a graphic user interface (GUI) allows easy problem formulation. The EHWPack was implemented on the HP Exemplar shared-memory supercomputer at the California Institute of Technology with 256 CPUs and 64 GB of memory. The hardware configuration is a board mounted with four FPTAs. The board is controlled by National Instruments data acquisition hardware and software (LabView). The LabView Software implements a TCP/IP client-server system where the server is the LabView and the master processor running on neptune is the client.

3. EXPERIMENTS/PERFORMANCE

This section presents an analysis of the EHWPack performance in terms of speed of circuit evolution. We start by reviewing the basic features of the architecture of the FPTA

The FPTA cell 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 FPTA cell consists of 8 transistors and 24 programmable switches. The FPTA cell was manufactured using 0.5 micron CMOS technology. The chip allowed us to use circuits obtained through evolution in simulations and validate them by downloading and evaluating their performance in hardware.

The objective of this investigation is to assess the performance of the tool in two typical problems of EHW: the evolutionary synthesis of a Gaussian computational function (Stoica, 1999b) and the evolution of a band-pass filter tuned to the AM band where the SPICE simulator is used to evaluate the circuit performance. The fitness value is the mean square error between the desired DC/AC signals and the DC/AC signals obtained by the circuit. We investigate the impact of varying the number of FPTA cells and the number of processors in the elapsed time.

# Cells	**Pop. x Gen.**	**#Proc**	**Elapsed Time (sec)**	**Norm. Time (msec)**	**#Trans**
1	50 x 200 (25)	1	407	81.4	8
		16	56	11.2	8
		32	46	9.2	8
		64	49	9.8	8
		128	63	12.6	8
2	400 x 200 (200)	1	5860	146.5	16
		16	491	12.3	16
		32	321	8.0	16
		64	240	6.0	16
		128	222	5.6	16

# Cells	Pop x Gen	#Proc	Elapsed Time (sec)	Norm. Time (msec)	#Trans
1	200 x 200 (100)	1	1445	72.3	8
		16	144	7.2	8
		32	101	5.1	8
		64	72	3.6	8
		128	79	4.0	8
2	200 x 200 (100)	1	2034	101.7	16
		16	214	10.7	16
		32	145	7.3	16
		64	115	5.8	16
4	100 x 200 (50)	1	2971	297.1	32
		16	288	28.8	32
		32	184	18.4	32
		64	146	14.6	32
		128	154	15.4	32

Table 1 – EHWPack performance in the evolution of a Gaussian circuit.

Table 1 shows time statistics associated with the synthesis of the Gaussian. We compare the tool speed when using 1, 16, 32, 64 and 128 processors of the supercomputer. We also vary the number of FPTA cells used in the netlist: 1, 2 and 4 cells.

The table 1 shown above provides the following information: number of FPTA cells #*Cells*; number of processors #*Proc*; population size *Pop*; number of generations *Gen*; elapsed time; normalized time *NormTime*; and the number of transistors #*Trans* in the SPICE netlist. The elapsed time is the total time taken to run the experiment. The normalized time is the elapsed time divided by the number of individuals evaluated (half of the product population x generations shown in the above table, since we replace only 50% of the population at each generation). The number of individuals evaluated per generation is given within parenthesis in the second column. As shown in the table, we used different population sizes in the experiments in order to assess the influence of this parameter in the performance. As expected, the normalized time decreases as we increase the number of processors. We can further observe that, as we increase the number of FPTA cells and, as a consequence, the number of transistors in the netlist, the normalized time increases as well. Finally, the fitness of the best circuit using 1,2 or 4 cells and running on 1, 16 to 128 processors are the same. It is interesting to observe that for small populations (50 individuals in the single cell experiments), increasing the number of processors improves the speed only to a certain extent: for more than 32 processors, the communication overhead produces an increase in the computation time. This stems from the fact that using the master/slaves algorithm only 25 individuals are evaluated in parallel per generation. In order to fully explore the parallelism, large populations should be used, as in the experiment with two cells (400 individuals). In this case, 128 circuits can be evaluated in parallel on 128 processors.

		128	94	4.7	16
4	200 x 200 (100)	1	4895	244.8	32
		16	402	20.1	32
		32	250	12.5	32
		64	146	7.3	32
		128	119	6.0	32

Table 2 – EHWPack performance in the evolution of a bandpass filter.

Table 2 is analogous to Table 1, displaying the evolution time for the synthesis of a band-pass filter. While the Gaussian circuit requires a DC transfer analysis, filters require the use of the frequency domain analysis of the SPICE simulator. In the case of the filter experiment, we kept the number of evaluated individuals constant, so that we can clearly observe the increase in evolution time as we include more cells in the netlist. In this case also the fitness of the best circuit using 1, 2 or 4 cells and running on 1, 16 to 128 processors are the same. If we compare the statistics shown in Tables 1 and 2, it can be observed that the filter experiment consumes less time than the Gaussian experiment. This is consistent with the fact that the AC analysis is less time consuming than the DC analysis.

Further, we compare the statistics shown for the *extrinsic* evolution in Table 1 with the ones measured in an *intrinsic* experiment, where two FPTA cells were used as a hardware evolution platform. A total of 20 individuals have been evaluated along 200 generations. The elapsed time was 272 seconds. The total time spent in the TCP/IP connection between the supercomputer and the PC was 5 seconds. The computation time spent on the PC is 180 seconds and on the supercomputer is 87 seconds. We should mention that most of the evaluation time on the PC is used by LabView to download the configuration bits into the chip and to acquire the data from the chip (153 seconds). The time needed to obtain a DC transfer analysis of the circuit is reduced to 27 seconds. This value can be normalized, dividing it by the number of individuals evaluated (4000), giving 6.75 ms. This number is one order of magnitude less than that obtained for the simulated experiments with 2 FPTA cells and using only one processor, and it is equivalent to the time observed when using 128 processors. Another advantage of hardware evolution is that the elapsed time does not increase with the number of cells. We are currently working on an enhanced version of the chip that will integrate 36 FPTA cells. The chip will be mounted on a dedicated board with processor, memory and analog/digital converter to accelerate the chip reconfiguration and the data acquisition.

4. CONCLUSIONS

A parallel evolutionary software/hardware environment, EHWPack, was developed around PGAPack, SPICE as a simulator, and FPTA as reconfigurable VLSI chip to facilitate experiments in simulated and hardware evolution on a single platform. It allows experimentalists located at different sites, to design, optimize and test circuits using evolutionary algorithms in a user friendly, transparent and expeditious manner. Using the EHWPack, we were able to synthesize a Gaussian computational function and a band-pass filter tuned to the AM band in less than 4 minutes using 1, 2 or 4 FPTA cells. The same job took 1 hour 30 minutes in simulation on a 1 processor.. Finally, we observe that the time needed to evaluate one individual using the reconfigurable hardware (6.75 ms) is as fast as the time observed when using 128 processors and twenty times (2 FPTAs: 146.5 ms) to fifty times (4 FPTAs: 297.1 ms) faster that the SPICE simulator running on one processor.

ACKNOWLEDGMENT

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 Defense Advanced Research Projects Agency (DARPA) under the Adaptive Computing Systems Program and the National Aeronautics and Space Administration.

REFERENCES

Bennett, Forrest H III, Koza, John R., Shipman, James, and Stiffelman, Oscar. (1999). Building a parallel computer system for $18,000 that performs a half peta-flop per day. In. *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, July 13-17, 1999, Orlando, Florida USA. San Francisco, CA: Morgan Kaufmann. Pages 1484 - 1490. (http://www.genetic-programming.com/machine1000.html)

Heitkötter, J. (1997), *The Hitch-Hiker's Guide to Evolutionary Computation*, at http://gnomics.udg.es/~encore/www/

D. Keymeulen, A. Stoica, R. Zebulum (2000). Fault-Tolerant Evolvable Hardware using Field Programmable Transistor Arrays. In *IEEE Transactions on Reliability, Special Issue on Fault-Tolerant VLSI Sytsems*, vol. 49, No. 3. IEEE Press

Koza, John R., Bennett III, Forrest H, Andre, David, Keane, Martin A, and Dunlap, Frank (1998), Automated synthesis of analog electrical circuits by means of genetic programming, In *Journal of IEEE Transactions on Evolutionary Computation*. 1(2). pp. 109-128.

Levine, D (1994), *PGAPack: Parallel Genetic Algorithm Library* http://www-unix.mcs.anl.gov/~levine/PGAPACK/index.html,.

Stoica, A, et al. (1999b). Evolutionary experiments with a fine-grained reconfigurable architecture for analog and digitl CMOS circuits. In, *Proceedings of the First NASA/DoD workhop on Evolvable Hardware*. Los Alamitos, CA: IEEE Computer Society Press.

Stoica, A et al. (1999c). Evolutionary Design of Electronic Devices and Circuit. In *Proceedings of the 1999 Congress on Evolutionary Conputation,* July 6th-9th, 1999, Washington DC. IEEE press.

Stoica, A, Zebulum R. and Keymeulen D. (2000) Mixtrinsic Evolution. In *Proceedings of the third International Conference on Evolvable Systems: from Biology to Hardware,* Edinburgh, Scotland, UK, April 17-19. Springer-Verlag.

Thompson, A. (1998). On the Automatic Design of Robust Electronics Through Artificial Evolution. In *Proceedings of the Second International Conference on Evolvable Systems: From Biology to Hardware (ICES98*), vol. 1478, pp. 13-24, LNCS, Springer-Verlag.

van Lent, M. (1999), *PARAGenesis: a Genesis on the CM-200 in C**, at ftp://ftp.aic.nrl.navy.mil/pub/galist/src/ga/paragenesis.tar.Z

Wall M. (1999), *C++ library for Genetic Algorithm*, at http://lancet.mit.edu/ga/

Zebulum, R.S., Stoica, A., Keymeulen, D. (2000) A Flexible Model of a Field Programmable Transistor Array Targeted For Hardware Evolution, In the *Proc. of the Third International Conference on Evolvable Systems (ICES2000)*, Edinburgh, Scotland, April 17-20. Springer Verlag.