Evolvable Hardware — A Short Introduction

Jim Torresen

NERA Telecommunications, P.O. Box 91, N-1361 Billingstad, NORWAY

E-mail: jim@nera.no

Abstract

This paper introduces a new hardware design scheme inspired by biological systems. The method is named Evolvable Hardware (EHW), since evolutionary algorithms are applied for the circuit design. The idea is that the system should consist of a large set of simple hardware units with connections only to their nearest neighbors. All the units operate concurrently with no global control. However, the local behavior of the complete system determines the global behavior of the system. The connections between the units and the operation of each individual unit are determined using genetic algorithms. Thus, the system is evolved until it attains a structure performing as initially specified.

This paper surveys the undertaken work on EHW. So far few experiments have been reported. One of the reasons to this is that little applicable hardware has been available. The experiments which have been reported are on simple problems. However, the results are promising and many have great expectations on the future of EHW.

1 Introduction

The traditional way of designing hardware has been by drawing schematics or by using a hardware description language. This involves many design considerations to be taken care of like timing constraints and possible glitches. Another possible approach is by *evolving* the circuit description. By this method, first a set of circuits are randomly generated. The behavior of each circuit is evaluated and the best circuits are combined to generate new and hopefully even better circuits. The evaluation is according to a behavior initially specified by the user. After a sufficient number of generations the fittest circuit is to behave according to the initial specification.

Natural evolution is based on Darwin's thinking about development where strong individuals survive and give rise to new generations. In relation to evolution, undertaken in a large population, we speak about collective behavior arising from simple interaction between individuals. One example of collective behavior is an ant society, where the global behavior emerges from local interactions between individual ants. These principles from the nature inspire researchers to develop new architectures where the configuration is determined by evolution and with no global control of each building block. This in contrast to the general computers used today which are all based on global control mechanisms.

This paper is a shortened version of the complete paper presented in [Torresen, 1997]. Related conference proceedings are [Sanchez and Tomassini, 1996] and [Higuchi et al., 1997].

Evolvable hardware (EHW) is reconfigurable hardware that can be evolved. The idea of EHW was first introduced for about four years ago. Little has happened through the years since then, thus, it is still quite open how evolvable hardware — if any, will be in the future. EHW is based on evolving the circuit configuration instead of designing it the ordinary way. That is, different randomly generated configurations are tested in a programmable hardware device. The configurations that make the device output responses closest to the wanted response are combined to make even better configurations until an usable device is achieved.

The evolution can also be undertaken completely in software. Today, software simulation is by far the most used method, since little EHW has been available. See [Torresen, 1997] for an introduction to research projects using software based evolution.

A large range of applications have been proposed for EHW:

- Autonomous robots:
  - Vacuum cleaners
  - Surveillance robots
  - Automatic De-mining Vehicles
  - Maintenance robots
- Pattern recognition
- Signal processing

One hopes that EHW can solve the difficult problems in these applications better than other methods like ordinary artificial neural networks. Autonomous robots requires such complex systems that it has not yet been possible to develop functional systems. By evolving a system instead of designing or programming it, the complexity can be highly increased. To use such a system in real time, special hardware is required.

1.1 Genetic Algorithms

Genetic Algorithms (GA) are the most commonly used algorithms for evolutionary computing [Holland, 1975]. The algorithm is illustrated in Figure 1.

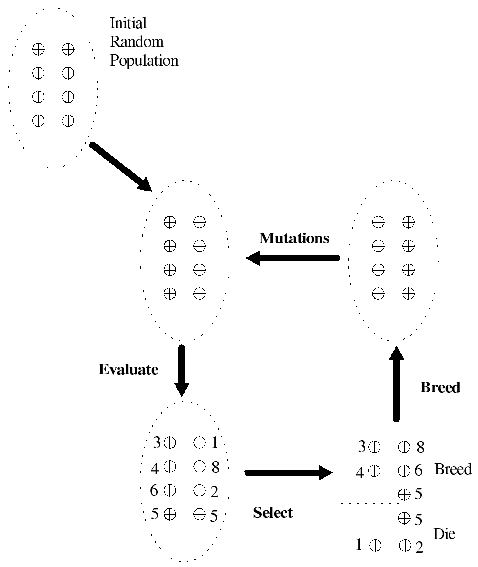

Figure 1: The genetic algorithm principle.

In GA, the members of a population are initially randomly generated. A member of the population may contain a program, hardware configuration or other kinds of representation and is often called a chromosome or a genotype. The members are evaluated and sorted on fitness according to a given fitness criteria. This is usually a high-level function and may be e.g. the deviation from a specified behavior of a robot. The parameters of chromosomes of the fittest members are exchanged to generate — for each couple, two new offsprings – preferably fitter than the parents, in the next generation. In the figure, no parents are transferred to the next generation. Another variant of GA combines a few of the best members, and the children are exchanged with the lesser fit members of the population. Mutation may also occur and introduces changes in the chromosomes that may not be in the parents.

1.2 Methods in Evolvable Hardware

Hardware evolution can be divided into two sub-groups:

- **Off-line EHW:** The evolution is simulated in software, and only the elite chromosome is written to the hardware device (off-chip evolution).
- **On-line EHW:** The hardware device gets configured for each chromosome for each generation (on-chip evolution). Thus, the genetic operations are done in software, while evolvable hardware is used to test the fitness of each member of the population.

In the future, one may design devices which allow the complete evolution on-chip. Then, it is possible to continue the evolution when the device is operating in its working environment. However, in today's systems the evolution is undertaken once with no continued evolution while the device is in use.

So far, all systems are based on one or a few circuits. In the future the systems can become more complex, where subsystems are separately evolved and then put together to a larger system. For hardware evolution, building blocks are required and so far gate level (AND,OR,..) is most commonly used, while higher level functions are possible too.

2 A survey of EHW Research

This following sections contain a survey of various approaches to evolvable hardware. There is a limited number of projects where evolutionary hardware has been a part of the evolutionary process. Furthermore, the various projects are quite different from each other and they have got few common properties. The first projects are based on evolution using analog technology, while the latter projects are using digital hardware.

2.1 Analog Devices

This section contains a description of various evolutionary approaches based on analog technology. All work done so far use off-line EHW. This is due to the overhead required for designing the circuit. However, recently analog programmable Field-Programmable Analog Arrays (FPAA) were introduced from several companies. These should make analog on-line evolution possible.

Analog Circuit Synthesis

Automatic analog network synthesis using genetic algorithms has been proposed by several researchers [Grimbleby, 1995][Koza et al., 1996]. Koza et al. generate circuits by using a modified SPICE simulator for measuring fitness of each individual. They apply Koza's proposed genetic programming approach, i.e. the population consists of computer programs of varying sizes and shapes. They have successfully evolved a low pass filter and an Op Amp. Grimbleby generates analog circuits using the ordinary genetic algorithm.

2.2 Digital Approaches to Evolvable Hardware

Programmable semiconductor devices have existed for a fairly long time. About ten years ago the more complex

Field Programmable Gate Arrays (FPGAs) were introduced. A string of configuration bits is downloaded to the FPGA and it determines the connections and functions of the internal gates and thus the operation of the device. The configuration bit string makes the device easily reconfigurable.

However, earlier programmable devices were of limited use as EHW due to several problems:

- Limitation in number of possible re-programming operations.
- Possibility of damaging the device by an in-valid configuration.
- Long down-loading time for the configuration.
- Inhibition of partial downloading of the configuration string.
- Slow speed of operation.
- High cost.

These limitations have severely limited the interest in evolvable hardware research. However, recently new devices have been introduced were these problems are reduced. The complexity and speed is increasing and the price per gate is decreasing.

The problem of long downloading time for an FPGA configuration is partly because of the inhibition of partial downloading. Recently however, a new device — XC6200, from Xilinx were announced. This device is partial programmable, i.e. each cell configuration can be individually programmed ($ns - \mu s$). This makes mutation-operations simple to implement. Further, the speed is high with 220 MHz flip-flop toggle rate.

The cells communicate to their nearest neighbors and each cell is of simple nature. Both an advantage for evolution. Any configuration string may be downloaded to the device without risk of damaging the device. However, the configuration must be down-loaded *into* the device, thus, a cell can not reconfigure itself.

In the following sections, the various research projects based on digital evolvable hardware are described.

Embryonics

Embryonics are proposed to be specially designed FPGAs. They behave like biological cell groups, with ability to reproduce, differentiate and self repair [Marchal, 1994]. The computer architecture embryonics are inspired by processes in molecular biology. There are several problems, e.g. that the large storage required to keep the complete chromosome in each cell. No special FPGAs have been produces so far, but commercial devices have been used to validate the different steps in the scheme, using a prototype design named *biodule*.

The Firefly Machine

An evolving, one-dimensional, non-uniform cellular machine has been implemented [Sipper, 1997]. It is based on several XC6200 devices on a single board. The system consists of 56 binary-state cells, each containing a rule table. The evolution is completely on-line, i.e. also the genetic operations are carried out on the board. The system is evolving a configuration where the state of each cell oscillate between all zeros and all ones on successive steps. Thus, it behaves like a swarm of fireflies — flashing on and off in a unison way.

Evolution by HDL

A Hardware Description Language (HDL) is a programming language and a hardware design tool. A system is proposed where HDL programs are evolved to generate hardware structures and behavior [Hemmi et al., 1994]. The target architecture for an evolved circuit is an FPGA. The approach has been successfully used to evolve a binary adder performing summation free of error after 251 generations. Only off-line evolution is used.

On-line Evolution of FPGAs

The first experiment based on on-line EHW using FPGAs was conducted by Thompson [Thompson, 1996]. He refrains from the digital aspects of a XC6200 FPGA and treats it like an analog, dynamic and asynchronous device. The evolution is allowed to explore the chip without concern about normal design restrictions like switching transients and clock delay. The chromosome was defined by 100 gates and their input connections. Thus, this is gate level evolution. No flip-flops were used in the evolution.

The purpose of the first experiment was to evolve an oscillator without external signal input. This showed to be possible with fairly high precision and oscillators with frequencies of 10 Hz and 1 kHz were evolved.

In a later experiment, Thompson has evolved a tone-discriminator in the XC6216 FPGA [Thompson, 1996b]. A 10 x 10 corner of the FPGA were used and the configuration was evolved to discriminate between square waves of 1 kHz and 10 kHz presented at the single input. The final circuit — the result of evolving 5000 generation, appeared to be perfect when observed by eye on an oscilloscope.

Functional Level Evolution

If hardware is genetically synthesized from higher level hardware functions — higher than gate level, more complex applications can be evolved [Higuchi, 1996].

Thus to conduct experiments on functional-level evolution, a new FPGA model — F^2PGA, is proposed by Higuchi and shown in Figure 2.

It consists of a number of Programmable Floating Units (PFUs) interconnected by crossbar switches. The function of the device is dependent on the PFU's function and

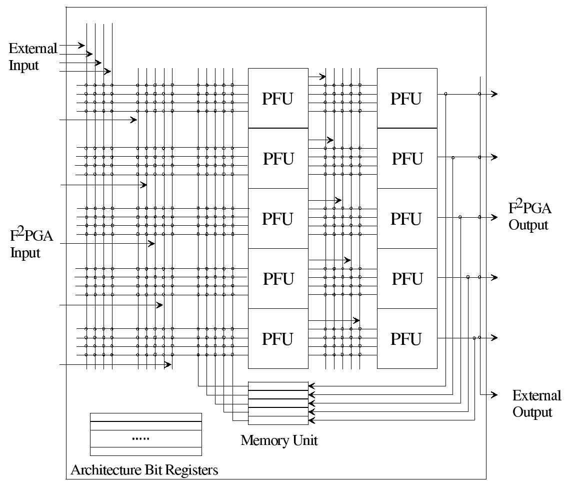

Figure 2: Function based FPGA.

the crossbar switch settings, which are represented in the chromosome. The length of the chromosome is variable to obtain faster GA execution and allow for large scale evolution. The Memory unit stores the "state" of the system. The Architecture Bit Registers store the configuration bits for the F^2PGA. Each PFU can perform functions like adding, subtracting, multiplication, cosine and sine using floating point numbers.

The GA operations are selection and mutation, and no crossover is used. After the fittest chromosomes have been selected they all undergoes mutation. This may be either mutation of the operand — i.e. mutation of a crossbar setting or mutation of a function — i.e. mutation of the PFU operation.

The proposed scheme will be tested by a newly designed ASIC. This will be one of the few devices specially designed for evolution. Simulations have been conducted and indicates performance equal to artificial neural networks.

3 Conclusion

Research on evolvable hardware can be summarized in the following conclusions:

- Evolvable hardware is proposed to be a method for designing complex systems.
- Several different evolutionary methods are proposed and under development at the time being.
- Few experiments involving on-chip evolution have been undertaken.
- Both evolutionary techniques and evolvable hardware circuits have to be further improved to be used for complex real world applications.

Today the achievements are few and the small number of experiments undertaken by EHW show that it is still a way to go before we have an evolved vacuum cleaner moving around in our homes doing the cleaning while we are at work.

At the same time, it seems like most researchers in the field of evolvable systems are using software simulation, while it is probably the development in hardware that will determine how large the progress will be.

References

[Grimbleby, 1995] J. B. Grimbleby: "Automatic analogue network synthesis using genetic algorithms." In Proc. of the 1st IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA95),pp 53-58. IEE Conf. Publ. No. 414, 1995.

[Hemmi et al., 1994] Hitoshi Hemmi et al., "Development and evolution of hardware behaviours." In R. Brooks and P. Maes, editors, Artificial Life IV, pp 371-376. MIT Press, 1994.

[Higuchi et al., 1996] T. Higuchi et al., "Evolvable Hardware with Genetic Learning". IEEE Int. Symp on Circuits and Systems, ISCAS96, May 13th, 1996, Atlanta.

[Higuchi et al., 1997] T. Higuchi et al., "Evolvable Systems: From Biology to Hardware". First Int. Conf., ICES 96. LNCS, vol 1259, Springer-Verlag 1997.

[Holland, 1975] J.H. Holland, "Adaption in Natural and Artificial Systems". The University of Michigan Press 1975, Ann Arbor, Michigan.

[Marchal et al., 1994] P. Marchal et al. (1994) "Embryological Development on Silicon." In Proc. of ALife IV.

[Koza et al., 1996] JR Koza et al., "Automated design of both the topology and sizing of analog electrical circuits using genetic programming". In Gero, JS and Sudweeks, F (eds). Artificial Intelligence in Design '96. Dordrecht: Kluwer Academic Publishers.

[Sanchez and Tomassini, 1996] E. Sanchez and M. Tomassini, Eds., "Towards Evolvable Hardware", Lecture Notes in Computer Science, vol. 1062. Heidelberg, Germany: Spinger-Verlag, 1996.

[Sipper, 1997] Moshe Sipper, "Evolution of Parallel Cellular Machines: The Cellular Programming Approach" Lecture Notes in Computer Science, Springer-Verlag, 1997.

[Thompson, 1996] Adrian Thompson, "Silicon Evolution". Proceedings of Genetic Programming 1996.

[Thompson, 1996b] Adrian Thompson, "An evolved circuit, intrinsic in silicon, entwined with physics". Proc. 1st Int. Conf. on Evolvable Systems (ICES96). Springer LNCS.

[Torresen, 1997] Jim Torresen. "Evolvable Hardware — the Coming Hardware Design Method?". In N. Kasabov, editor, "Neuro-fuzzy tools and techniques". Springer-Verlag Group (Physica-Verlag) in the series "Studies in Fuzziness and Soft Computing". To appear.