Evolved Synthesis of Digital Circuits

Laurenţiu Ionescu
University of Pitesti, Romania

Alin Mazare
University of Pitesti, Romania

Gheorghe Şerban
University of Pitesti, Romania

Emil Sofron
University of Pitesti, Romania

INTRODUCTION

Traditionally physical systems have been designed by engineers using complex collections of rules and principles. The design process is top-down in nature and begins with a precise specification. This contrasts very strongly with the mechanisms which have produced the extraordinary diversity and sophistication of living creatures. In this case the ''designs'' are evolved by a process of natural selection. The design starts as a set of instructions encoded in the DNA whose coding regions are first transcribed into RNA in the cell nucleus and then later translated into proteins in the cell cytoplasm. The DNA carries the instructions for building molecules using sequences of amino acids. Eventually after a number of extraordinarily complex and subtle biochemical reactions an entire living organism is created. The survivability of the organism can be seen as a process of assembling a larger system from a number of component parts and then testing the organism in the environment in which it finds itself (Miller, 2000).

The main target of the **evolvable hardware** is to build a digital circuit using bio inspired methods like genetic algorithms. Here the potential solutions are coded like configuration vectors which command interconnection between logical cells inside the reconfigurable circuit. All configuration vectors represent the genotype and one single configuration vector is the individual with its own characteristics (like chromosome).

The individuals are generated by genetic operators like crossover or mutation. One individual give one solution circuit which is tested in **evaluation** module. The circuit obtained from the individual consist the phenotype. The circuit behavior is compared with target functions, which we desire to implement. The result is fitness: if the circuit approximates the behavior of the target function, we have a good fitness for the individual which generate the circuit. Then each individual whit its fitness gets into selection module where the future parents in crossover and mutation are decided. Finally we have a circuit solution which implements the target function. We have an evolved synthesis of digital circuit – a method like assemble and test.

This method can be useful because explore the design space beyond the limits imposed by traditional design methods. Two research directions are developed in evolvable hardware. In extrinsic evolvable hardware the individuals are obtained from software implementation on computer and phenotype consist in high level abstract circuits like SPICE object files or FPGA configuration files (.bit). The **intrinsic evolution**, on the other hand, supposes that entire evolution process is inside one or more chips (FPGA): the hardware implementation of evolved hardware.

The challenge is to design an intrinsic evolution because can be used for applications like robots control system. But this involves implementation of the software based algorithms in hardware modules.

BACKGROUND

The **dynamic reconfigurable hardware** area and evolvable hardware knows, in the last years, a fast evolution. Ten years ago the digital circuit implementation, with high degree of complexity involve more problems caused specially by technologies limits. The

market was up most by complex programmable gates array or by the low grains field programmable gates array where upon the main problems are the number of Boolean cells available on chip and the delay time. The fast evolution of the technologies increases in our day the performance of programmable circuits. Thus, today is possible to implement a high speed central processing units core which is comparable with the application specific integrated circuit implementations. Therewith the low product costs make that a modern programmable digital circuit can be purchased by end users like students and researchers. Thus an evolution in designing, synthesis and implementation techniques with programmable logic circuits is required. One very attractive direction of research is implementation of hardware bio-inspired systems on programmable logic circuits like neural networks or evolutionary algorithms (e.g. genetic algorithms).

The first research direction in this area is to find solutions for improve the genetic algorithm performance by hardware implementation. Software implementations have the advantage of flexibility and easily configuration. However, the convergence speed is slow because the serial execution of the steps whiles the algorithm run. To increase the speed a parallel implementation of the modules is required (Goldberg, 1995). This is done by hardware implementation on programmable logic circuits. More investigations are done in this area.

The second research direction is to join the concept of assemble-and-test together with an evolutionary algorithm to gradually improve the quality of a design has largely been adopted in the nascent field of **evolvable hardware** where the task is to build an electronic circuit.

Thompson (Thompson, 1999) makes the first research, uses a reconfigurable platform and showed that is possible to evolve a circuit which could discriminate between two square wave signals. He demonstrates that is possible to design digital circuit using evolved algorithm. The evolutionary process had utilized physical proprieties of the underlying silicon substrate to produce an efficient circuit. He argued that artificial evolution can produce design for electronics circuit which lies outside the scope of conventional methods. Koza (Koza, 1997) have pioneered the extrinsic evolution of analogue circuit using SPICE and have automatically generated circuit which are competitive with human designer.

Most workers are content with the extrinsic evolution because the evolutionary algorithm is software – based. But Scott and others achieve different solutions for hardware implementation of evolutionary algorithms. In (Scott, 1997) is design a **hardware genetic algorithm** implemented as pipe line hardware structure in FPGA. His work is a demonstration that full integrated **evolved hardware** (**intrinsic**) solution can be implemented. To design hardware modules for crossover, selection and mutation are used **combinational network**s such as systolic arrays presented by (Bland2001).

Miller et al. (Miller,2000) give a reference with his work concerned of the evolution of combinational digital circuit to implement arithmetic functions. First he uses the gates networks which evolve in arithmetic circuit but demonstrate that is possible to use evolution of some sub-circuits to achieve more complex circuits.

Another example of extrinsic evolved synthesis is give by Martin's work (Martin 2001). Here the phenotype is give by a hardware description language (HandleC) sequences.

But utilization of genetic algorithm in hardware design is in any more areas. In (Shaaban 2001) is used in integrated circuit design in semiconductor detectors. Yasunaga (Yasunaga 2001) use a hardware synthesis of digital circuit in reconfigurable programmable logic array structure to implement speach recognition system. Recently evovable synthesis is used for sequential circuit design(Ali 2004) or in digital filters design (Iana2006).

The evolved combinational and sequential synthesis for digital circuits is included as control module for self-contained mobile system to execute more tasks like obstacle avoidance and target hit in (Sharabi2006). In fact the solving of the multi objective (multi task) problems in hardware system by evolutionary is treated by Coello (Collelo 2002) and used in more recent works (Zhao 2006).

The question is: is possible to design digital circuits using an evolutionary algorithm like genetic algorithm (GA)?

To answer of this question, first, the design of a reconfigurable circuit which can be programmed by evolutionary algorithm is required. A solution can be reconfigurable multilayer gates network. Each gate in layer x can be connected or not whit a gate in layer x+1.