THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 1999 YEAR, ISSN 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS

EVOLUTIONARY DESIGN OF DIGITAL CIRCUITS - AN ALTERNATE APPROACH TO THE CONVENTIONAL LOGIC DESIGN

Rustem POPA

“Dunărea de Jos” University of Galati, Automatic Control and Electronics Department, Domnească Street - 111, Galați - 6200, Romania, Tel./fax: 036.460182, e - mail: Rustem.Popa@.ugal.ro

Abstract: Conventional design is based on top-down method employed by human designers. This kind of design employs a previous experience and a mysterious process of discovery, which originates in the human mind and is often called intuition or creativity. Unfortunately, this previous experience is in fact a collection of rules, concepts, and principles, which is a strong limiting factor in creativity and invention. Evolutionary design is in fact an Invention Machine that generates new unexpected and usually useful circuits. We discuss in this paper some successful evolutionary designs, and present an original example of evolutionary design of a Finite State Machine (FSM).

Keywords: Genetic Algorithms, Evolvable Hardware, Digital Circuits.

1. INTRODUCTION

The conventional design process is top-down and begins with a precise specification. In contrast with this approach, the “designs” of living matter are evolved by a process of natural selection. The design starts as a set of instructions encoded in the DNA of the cell. The DNA carries the instructions for building molecules using sequences of aminoacids and after a huge number of biochemical reactions an entire living organism is created.

Miller *et al.* (1999), considered that “it is indeed curious that organisms such as ourselves which are capable of imagining the world which operates according to definite laws and abstract design process were themselves produced by a mechanism which is entirely blind and has no particular object other than survivability”. 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. The concept of *assemble-and-test* together with an evolutionary algorithm can explore the entire design space because of the absence of imposed rules of design.

This concept has been adopted in the field of Evolvable Hardware(EHW), where the task is to build new electronic circuits. Research in EHW can be divided into *intrinsic evolution* which refers to an evolutionary process in which each phenotype is built in electronic hardware and tested, and *extrinsic evolution* that uses a model of the hardware and evaluates the phenotypes in software.

The rest of this paper is structured as follows. Section 2 discuss some transistor-level evolutionary designs. Section 3 presents some results in gate-level evolutionary design with emphasis on an evolved structure of a FSM. Section 4 points the conclusions of presented experiments.

2. TRANSISTOR-LEVEL EHW

Extrinsic EHW is the simpler in that only a computer and appropriate simulation software is necessary to carry out the evolutionary process. The circuit produced is easy to analyse by using an adequate software, but the computing power required to model even a small subset of the physical properties of semiconductors poses severe limitations on the speed of evolution. Intrinsic EHW offers two major advantages over extrinsic EHW: the speed evolution can be much higher, and the evolved circuit works in a real environment.

Layzell (1998), has developed a motherboard with a number of Programmable Crosspoint Switch Arrays that can be configured by direct writes to the PC’s internal I/O ports. In this way genotypes can be instantiated in hardware, in a very short time. The switches behave like low-value resistors (about 50Ω) when are closed, and like open circuits when they are open. Each switch can handle only a relatively small current (about 30mA), and for this reason the power supply of the evolved circuit does not exceed 3 Volts. The various values of resistances are obtained by combination series-parallel of the closed switches.

The author has decided to evolve on this motherboard a NOT gate using bipolar transistors as the evolutionary building block. This circuit is an interesting test, firstly, because as with many digital functions evolution can easily tend to a local optimum, which does not achieve the required behaviour, and secondly, because it’s difficult to envisage, how gradual improvement can be catered for by the fitness function.

The evolutionary algorithm used by the author was a genetic algorithm with single point crossover, rank-based selection, and elitism. To evaluate the curent circuit, a series of 100 test inputs was applied sequential, in a random order. The result of this intrinsic hardware evolution is given in Figure 1. We can see that transistor T2 is configured in an entirely unconventional manner. The only path to ground is achieved via a 10 MΩ input impedance of the oscilloscope. It’s very interesting how intrinsic evolution has exploited the oscilloscope input in this manner.

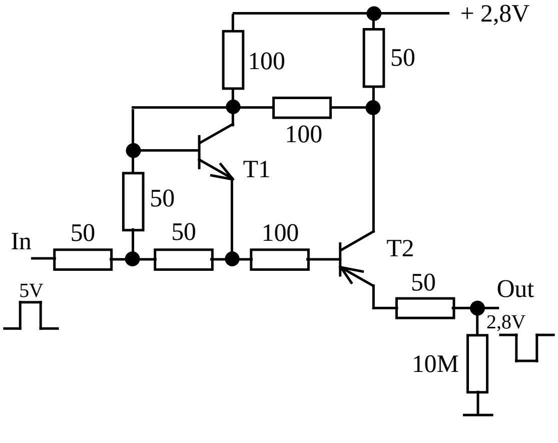

Figure 1. Intrinsically evolved NOT gate.

The upper limit of the operation frequency of this circuit is not specified by the author, but our SPICE simulations suggest a very low normal operation frequency (about 1 KHz).

Bennett *et al.* (1999), has developed an extrinsically evolved NAND gate on the basis of Genetic Programming(GP), an extension of the Genetic Algorithms, in which the population consists of computer programs. The procedure breeds a population of rooted, point-labeled trees with ordered branches, whereas electrical circuits consist of line-labeled cyclic graphs. GP can be applied to circuits, if a mapping is established between the program trees used in GP, and the cyclic graphs of electrical circuits. The principles of developmental biology suggest a way to map program trees into circuits, via a growth process that begins with a simple “embryonic circuit”. The population size is 132,000 and the maximum size of 300 points was established for each branch of each tree. The result of this evolution can be seen in Figure 2.

The wellknown human design of this TTL gate includes at least five transistors, four resistors and one diode. This evolved circuit has very good parameters approved by our SPICE simulations: a higher input impedance than a TTL gate, an acceptable output impedance of about 1KΩ, and an operation frequency of about 10MHz, three times lower than a standard TTL gate. The threshold voltage is about 4V, much higher than in TTL gates, and this is maybe the only questionable thing.

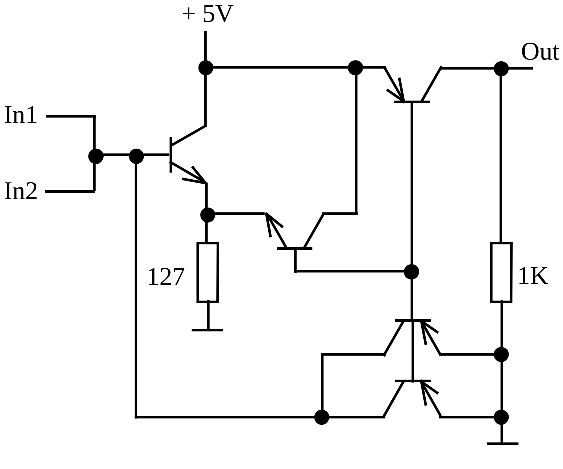

Figure 2. Extrinsically evolved NAND gate.

3. GATE-LEVEL EHW

Miller *et al*. (1999), has used the actual internal structure of the logical gates and an Evolutionary Strategy to find the optimum linkages between the inputs and outputs of the gates.

The proposed example is an extrinsically evolved one-bit full adder with carry. The resulting circuit is represented in Figure 3. We can see that the author has used also multiplexers. In fact a multiplexer can be built with three gates (2AND and 1OR), but some modern devices use the MUX "gate" as an atomic device, in that all other gates are synthesised using this.

An interesting feature of this circuit is that the sum and carry parts of the circuit are completely decoupled and there is a nice symmetry. This suggests that XOR gates naturally carry out elementary adition, while the MUX gates naturally synthesises the carry process of addition. The author has considered that this might be thought as a potential new principle which has not been observed by human designers.

It should be noted that the circuit was actually evolved by the separation of the carry function from the sum, and evolving them separately. When both are evolved together it becomes more difficult for the (1+4)ES evolutionary strategy to correctly synthesise the circuit. The optimum circuit from the Figure 4 is obtained after 20 runs of 2000 generations. The algorithm has used a population size of 50, 100% crossover, and 5% mutation.

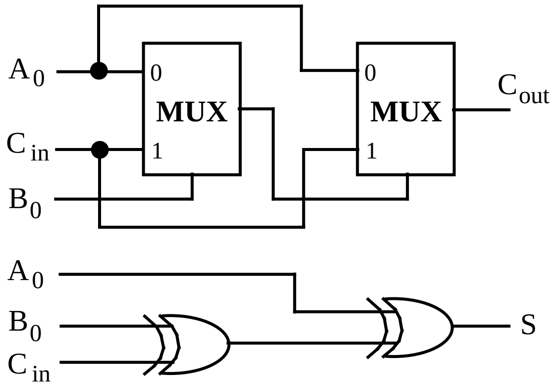

Figure 3. Extrinsically evolved one-bit full adder.

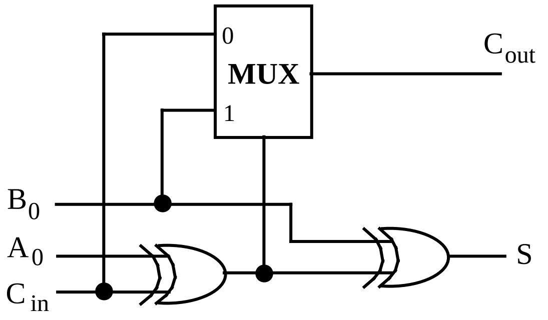

Figure 4. The best solution for one-bit full adder.

This is a remarkable circuit that was only recently discovered by the human designers. This circuit is still generally unfamiliar in the logic circuits designers community. However, certain FPGA manufacturers use this circuit as the basic unit in their adder macros. Evolved two bit adder is generated with two one-bit adders, using the ripple-carry principle, already known by human designers.

Popa, *et al.* (1999) proposed a comprehensive genetic algorithm for the evolvable synthesis of a FSM in a gate-level evolvable hardware implemented with a programmable logic device. We have taken as example a computer interface with 6 states, 4 inputs and 4 outputs, with state diagram shown in Figure 5. We have a maximum number of 5 inputs and a maximum number of 4 minterms for each function. Therefore, the number of fuse array links is $2 \cdot 5 \cdot 4 = 40$, and we have considered this number as the total length of the chromosome.

Our genetic algorithm is a standard one, with the population size of 100, and each chromosome has 40 bits. One point crossover is executed with a probability of 80%, and the mutation rate is 1%. A number of 10 worse chromosomes are replaced each generation. The stop criterion is the number of generations.

The evolved circuit is based on a GAL16V8 chip, which contains an AND array, and 8 logic cells configurable as OR gate, and a register device

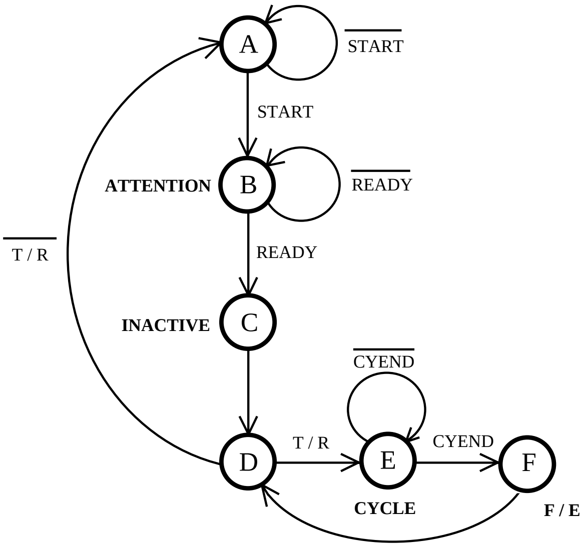
A
$\overline{\text{START}}$
START
ATTENTION
B
$\overline{\text{READY}}$
$\overline{\text{T / R}}$
READY
INACTIVE
C
$\overline{\text{CYEND}}$
T / R
CYEND
D
E
F
CYCLE
F / E

Figure 5. State diagram of the FSM

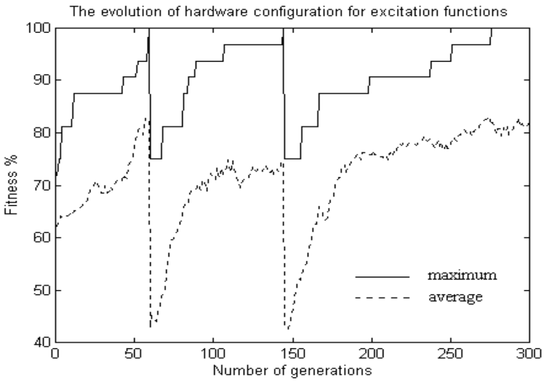

Figure 6. Evolvable synthesis of excitation functions

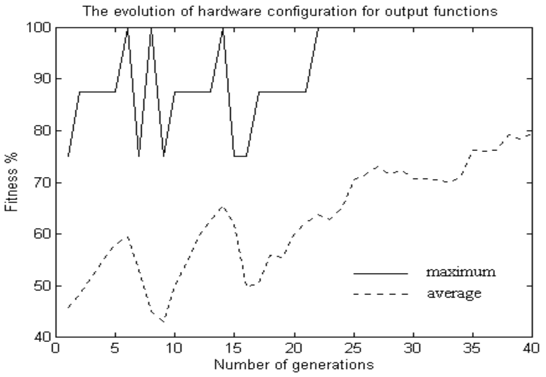

Figure 7. Evolvable synthesis of output functions

through some special configuration bits. The proposed FSM has 3 excitations functions, Q_i^+ with i = 1, 2, 3, and 4 output functions: ATTENTION, INACTIVE, CYCLE and F/E. Evolution may provide some non-minimal expressions for these boolean functions, but the minimization is not necessary for PLD implementations.

Figure 6 shows the evolution of the circuit for the first 3 functions. However, this circuit is built from 3 independent circuits, each generating one output bit. Therefore, the evolution of a circuit with one output bit is repeated 3 times. The Y axis is the correct answer rate. If it reaches 100%, then the hardware evolution succeeds. All 3 circuits are successfully obtained in less than 300 generations.

In the same way, Figure 7 reflects the evolution of the circuit for the output functions. The evolution succeeds after a less number of generations, because the total search space is in this case much smaller than in previous case.

Figure 8 shows the result of evolvable synthesis of this circuit. SPICE simulations show that timing diagram is just the same with the diagram achieved in a classical synthesis of the FSM.

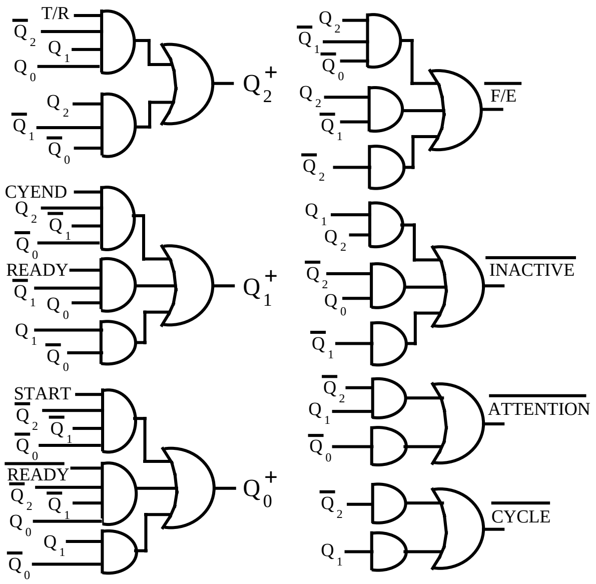

Figure 8. The evolved functions of the FSM.

4. CONCLUSIONS

We have presented in this paper some evolutionary designs of digital circuits. One of them, is an evolvable synthesis of a FSM, based on a standard genetic algorithm, and this is the main contribution of this paper. We have shown that evolutionary design is in fact a creative machine for new designs. Altough the evolutionary designs may have some limitations, the new principles may be able to be discovered by examining them. We think that the learning of new principles from a blind evolutionary process is a way for the future designs.

REFERENCES

Bennett, F. H. III, J. R. Koza, M. A. Keane, J. Yu, W. Mydlowec and O. Stiffelman (1999). Evolution by Means of Genetic Programming of Analog Circuits that Perform Digital Functions. In: *Proc. of the Genetic and Evolutionary Computation Conf, GECCO'99, Orlando, Florida, USA, July 1999,* 1477-1483

Layzell, P. (1998). A New Research Tool for Intrinsic Hardware Evolution. In: *Proc. of the Second Int. Conf. on Evolvable Systems, ICES'98, Springer-Verlag, 1998,* 47-56

Miller, J. F., T. Kalganova, N. Lipnitskaia and D. Job (1999). The Genetic Algorithm as a Discovery Engine: Strange Circuits and New Principles. In: *Proc. of the AISB'99, Edinburgh, UK, April 1999,* 65-74

Popa, R. and M. Iliev (1999). Evolvable Synthesis of Digital Circuits. In: *Proc. of the Fifth International Workshop on Bases of Electronics, SBE'99, Cluj-Napoca, Romania, June 1999,* 206-211