Evolutionary Discovering of the Concept of the Discrete State at the Transistor Level

Lukas Sekanina
Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic
sekanina@fit.vutbr.cz

Ricardo S. Zebulum
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA
rzebulum@mail2.jpl.nasa.gov

Abstract

This paper shows that the evolutionary approach can discover the concept of the discrete state in a physical hardware which can be reconfigured at the transistor level. Evolution is able to recognize that the required output values are not pure combinations of the input values and to build internal structures to store the state. In particular, the Reset-Set circuit and the D-latch circuit are investigated using the field programmable transistor array FPTA-2.

1. Introduction

Reconfigurable analog devices, such as Field Programmable Analog Arrays (FPAA) or Field Programmable Transistor Arrays (FPTA), are usually utilized for the evolutionary design of analog or mixed circuits in the field of evolvable hardware [1]. Only in a few cases *digital* circuits were successfully evolved in an analog reconfigurable device so far. In particular, papers [2, 3, 4] describe the evolution of simple logic gates at the transistor level. The evolution of molecular logic gates in the so-called NanoCell was introduced in [9]. The approach is referred to as *intrinsic* evolution because candidate circuits are evaluated directly in the reconfigurable hardware. Thompson evolved a single-electron NOR gate using a circuit simulator, i.e. extrinsically [10].

In general, most research in the area of the digital circuit evolution deals with *combinational* circuits. The evolutionary algorithm composes the combinational circuits either of elementary gates or high-level functional blocks such as adders, multiplexers or comparators. The evolutionary design of *sequential* circuits is considerably less mature (see discussion in [5]). However, sequential circuits are crucial for implementation of digital computational devices because they represent the circuit implementation of an abstract concept of the discrete state. The behavior of a computational system is usually defined as a sequence of values of discrete states which the computational system undergoes in order to transform the input data onto the output data. In general, the output data does not depend only on the current input data (as in the case of combinational circuits), but also on the internal state of the system. The internal state is stored in memory elements implemented using flip-flops or latches, i.e. as digital sequential circuits [6]. Only in some cases the sequential circuits were evolved. As a typical example, reference [5] describes the evolutionary circuit design for finite state machines (FSM) and counters. Flip-flops, registers, and elementary gates are used as building blocks. Various approaches were proposed to synthesize finite state machines from partial input/output sequences [7]. The evolution is typically performed using a circuit simulator, i.e. in the process of *extrinsic* evolution.

According to the knowledge of the authors of this paper, it is unknown whether the evolutionary approach can discover an implementation of a sequential circuit (for instance, a flip-flop) at the level of transistors or molecules, i.e. the levels different from the gate level. Sequential circuit evolution at the transistor level will be considered in this paper. Note that in order to conventionally implement the simplest sequential circuit in the most straightforward way, two digital gates (e.g. two NANDs) are needed. At the transistor level, a CMOS NAND (or NOR) gate can be implemented using four transistors, AND (or OR) using six transistors and XOR using 10 transistors. In another conventional imple-mentation, a static CMOS memory cell requires six transistors to be implemented [6].

The main objective of this paper is to explore whether the evolutionary approach can discover the concept of the discrete state in a physical hardware, which can be reconfigured at the transistor level. No information will be given about the state into the system a priori. The evolution must be able to recognize that the required outputs are not

pure combinations of the input values and to build internal structures to store the state. One possible application is that this technology may open the door to building sequential circuits with a rich/non-linear dynamics in a more compact way than traditional combinatorial-memory way. One of the first steps in this direction is to show that we are able to evolve a novel memory implementation. The field programmable transistor array FPTA-2 [2] will be utilized to conduct these experiments.

The rest of this paper is structured as follows. Section 2 surveys the sequential circuits designed by means of an evolutionary design approach. In Section 3, the experimental platform (FPTA) is introduced. Section 4 deals with the target circuits we are going to evolve and with the proposed design method. The obtained results are summarized in Section 5 and discussed in Section 6. Conclusions are given in Section 7.

2. A brief survey of relevant research

Table 1 surveys the sequential circuits evolved in the past years. We can observe that only one result dealing with the elementary sequential circuit (D-latch) is available. No result is available for the evolution at the transistor level. Table 1 also demonstrates that researchers are interested in more complicated circuits considered at higher levels of abstraction.

Circuits	Platform	Level	Ref.
D-latch	Extrinsic	Gate	[11]
Freq. divider, detectors, serial adder	Extrinsic	PLD	[7]
Seq. Filters	Extrinsic	Functional	[13]
Quadrature decoder	Extrinsic	Look-Up Tables	[12]
Counters, detectors	Extrinsic	Gate, flip-flop	[5]

Table 1. Typical sequential circuits designed using an evolutionary approach

The evolution of sequential circuits is usually considered as more difficult than the evolution of combinational circuits, because: (1) Longer chromosomes are needed in order to encode feedbacks. Longer chromosomes usually imply larger search spaces that are usually difficult to search. (2) The fitness calculation is more complicated, because in addition to testing all input/output combinations, it has to take into account the internal states. The evaluation time doubles by incrementing the number of internal states.

3. Evolvable platform: FPTA-2 and SABLES

A complete stand-alone board-level evolvable system (SABLES) is built by integrating the FPTA and a DSP implementing the Evolutionary design algorithm [2]. The system is connected to the PC only for the purpose of receiving specifications and communicating back the result of evolution for analysis. The system fits in a box 8" x 8" x 3". Communication between DSP and FPTA is very fast with a 32-bit bus operating at 7.5MHz. The evaluation time depends on the tests performed on the circuit. Many of the tests attempted here require less than two milliseconds per individual, and runs of populations of 100 individuals from 100 to 200 generations require only 20 seconds.

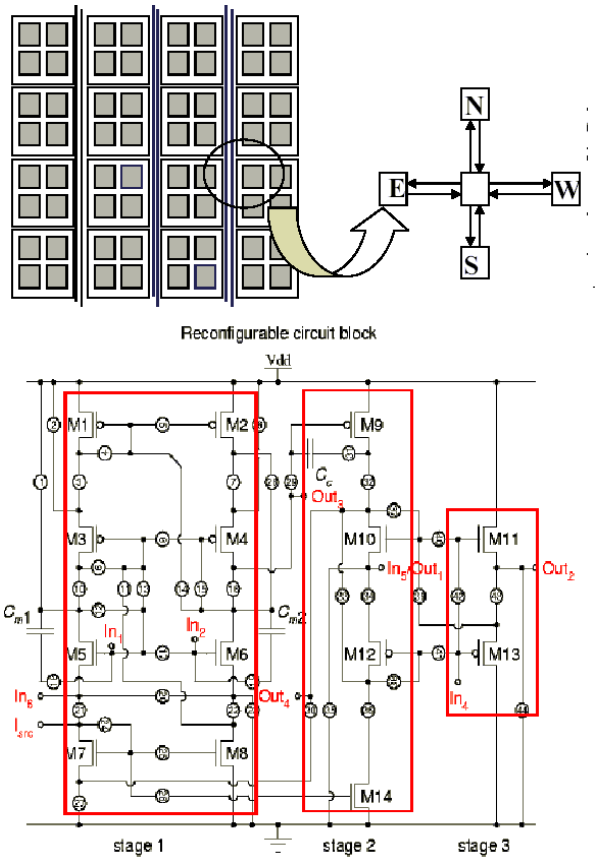


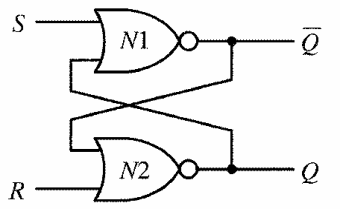
Figure 1: FPTA-2 architecture (top) and schematic of cell transistor array (down). The cell contains additional capacitors and programmable resistors (not shown).

The FPTA is an evolution-oriented reconfigurable architecture (EORA). It has a configurable granularity at the transistor level. It can map analog, digital and mixed signal circuits. The architecture of the FPTA consists of an 8x8 array of re-configurable cells. Each cell has a transistor array as well as a set of programmable resources, including programmable resistors and static capacitors. Figure 1 provides a broad view of the chip architecture together with a detailed view of the reconfigurable transistor array cell. The reconfigurable circuitry consists of 14 transistors connected through 44 switches. A total of ~5000 bits is used to program the whole chip. The pattern of interconnection between cells is similar to the one used in commercial FPGAs: each cell interconnects with its north,

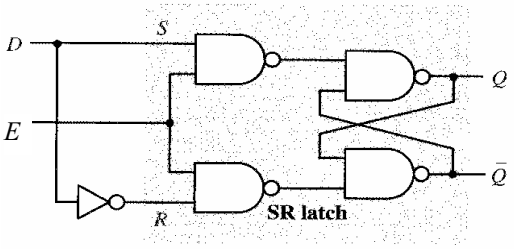
south, east and west neighbors. The reader can refer to [2] for more information on the FPTA-2.

4. Target circuits and the design method used

This section defines the objectives and describes a strategy used to achieve the objectives. The objective is to evolve Reset-Set circuit (RS) and D (data) latch circuit in the FPTA. The RS circuit is the simplest sequential circuit which is able to hold a logic value. The D-latch is a sequential circuit typically utilized in registers and counters. Figure 2 shows the specification and conventional implementation of these circuits at the gate level. CMOS-level implementations of their basic components, i.e. NAND and NOR gates, are shown in Fig. 3.

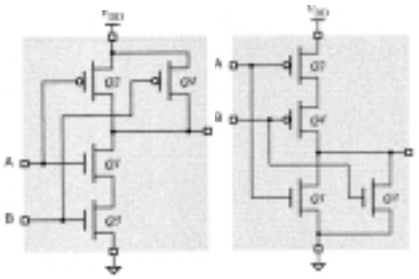


(a) RS circuit
If R=1 then Q=0
If S=1 then Q=1
If (S=0 and R=0) then hold the previous state



(b) D-latch circuit
If E=1 then Q=D
If E=0 then hold previous state

Figure 2: Specification and conventional implementation of (a) the RS circuit and (b) D-latch circuit.



(a) (b)

Figure 3: A conventional CMOS implementation of (a) NAND gate and (b) NOR gate.

The RS and D-latch circuits will be designed using a standard genetic algorithm operating directly with configurations of FPTA-2 as chromosomes. Only a few cells of the FPTA will be utilized for the experiments. The genetic algorithm running in a DSP uses the roulette-wheel selection, crossover and mutation. Candidate solutions are evaluated directly in FPTA-2. In this process a sequence of input signals consisting of 12 combinations of logic values (a training set) is applied at the circuit inputs. The Genetic algorithm must minimize the differences between the produced and required output values. In particular K=240 values are sampled, digitized and utilized during the evaluation of a candidate circuit. The input values can be seen in figures of Section 5. In contrast to combinational circuits, the evolved sequential circuits must be able to produce different output values for the identical input values, depending on the state of the circuit. It represents the main difficulty. The evolved circuits are also verified using various test input sequences. In general, the fitness function is as follows:

$$fitness = \sum_{i=1}^{K} | P(i) - T(i) |$$

where P(i) is the i-th circuit output value and T(i) is the i-th target output value.

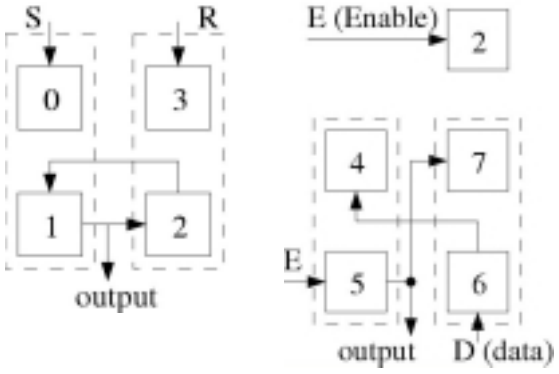
5. Experimental results

The following subsections describe the experimental setup and the results we obtained. All the circuits were evolved from scratch.

5.1 RS circuit

Fig. 4a shows the experimental setup used to evolve the RS circuit from scratch. Only the configuration bits of cells 0 and 1 (i.e. 2x77 bits) are stored in a chromosome. In order to establish a candidate circuit consisting of four cells the configuration bits of cells 0 and 1 are copied into cells 2 and 3. The solid lines in Fig. 4 denote external physical connections (wires) used to connect the cells. These connections were utilized to promote a specific design pattern which is typical for elementary sequential circuits (see Fig. 2). In addition to these connections, the evolution could interconnect the cells using the internal switches of the FPTA. Parameters of GA are as follows: the population size = 100, the crossover probability = 70%, and the mutation probability = 10%. Depending on experiment 300-1000 generations were produced.

Figure 5 shows the behavior of two of the best RS circuits we evolved. When both input values are at logic 0 and the previous output value is at logic 1, the circuit is still able to hold the logic 1. This value is not as strong as if S=1; however, it is still possible to consider the output value as correct. As Fig. 5 shows it is easy to improve the output value using an additional standard inverter gate. The evolved circuits were tested with various input sequences generated for the same time domain. We found very difficult to evolve a correct RS circuit (approximately one successful run out of 30 runs).



(a) (b)

Figure 4: Experimental setup for the evolution of (a) the RS circuit using 4 cells and (b) D-latch using 5 cells of the FPTA-2 (from scratch).

5.2 D latch

We used a similar experimental setup as for the previous problem. The differences are summarized in this paragraph: Figure 4b shows five cells and their physical interconnection by means of external wires. Evolution could also interconnect the cells using the internal switches. The chromosome contains configuration bits of the cells 2, 4 and 5. The configuration bits for the cell 7 (6, respectively) are copied from the cell 4 (5, respectively).

Although we performed more than 100 experiments, we obtained only one close-to-perfect D latch. Figure 6 shows its behavior for the training input sequence. The output values are not perfect; however they can be improved by means of two conventional inverters connected to the output (see the upper signal in Fig 6). Figure 7a illustrates that the circuit also works for a test input sequence. However, we were able to find a specific case for which the circuit does not work (see Fig. 7b). Hence the circuit can not be considered as a perfect D-latch.

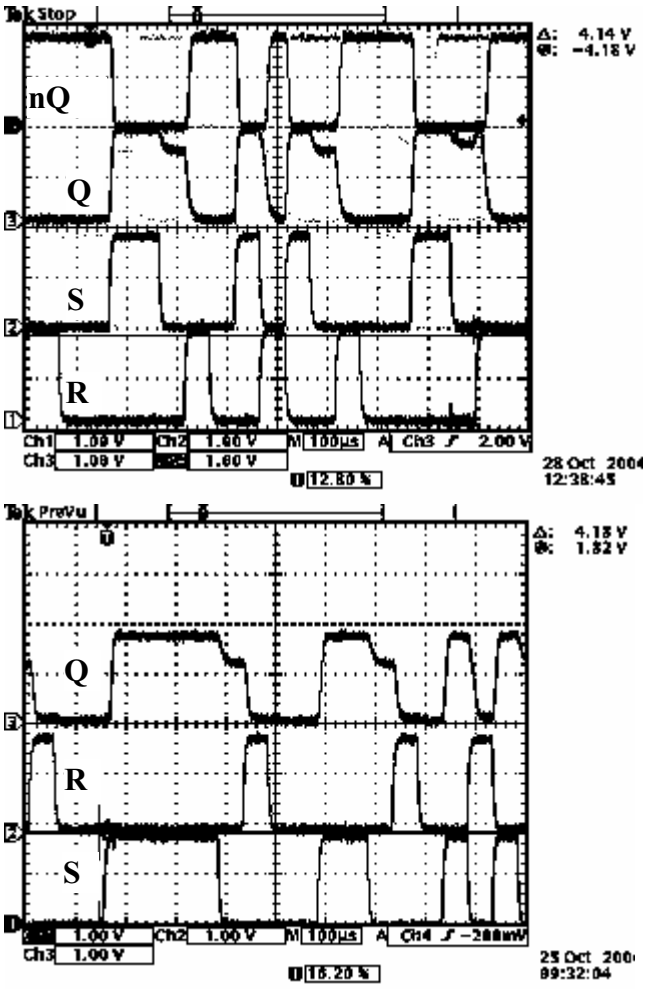


Figure 5: Behavior of two different RS circuits evolved form scratch. nQ is obtained from a conventional inverter connected to Q.

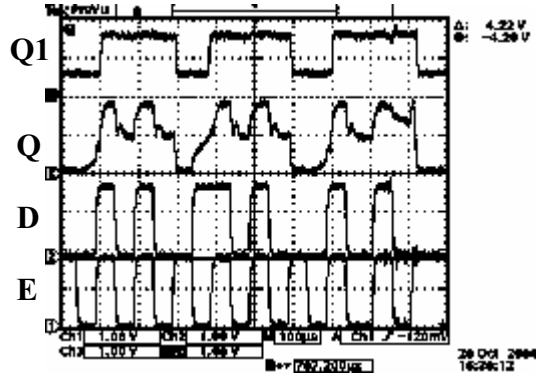
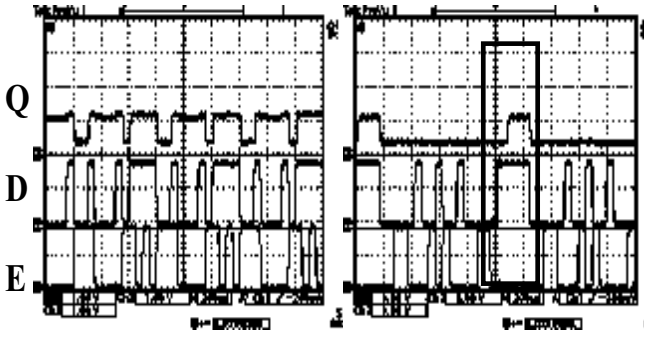


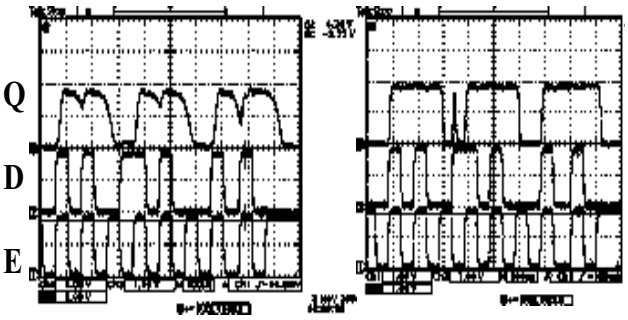
Figure 6: Behavior of an imperfect D-latch evolved form scratch. Q1 is obtained from two conventional inverters serially connected to Q.



(a) (b)

Figure 7: Analysis of an imperfect D-latch evolved form scratch: (a) test - OK, (b) test – failed (the output should be at logic 0).

Fig. 8 shows two typical imperfect behaviors corresponding to two different "D-latch" circuits we evolved very often. In the case (a), logic 1 is weak, which means that the circuit has problems to hold logic 1 when the both inputs are set at logic 0. There are no problems to hold logic 0. On the other hand, in the case (b), logic 0 is weak and there are no problems to hold logic 1. The evolution very often converges to one of these results. It seems very difficult for our GA and FPTA (perhaps impossible using the considered cells) to obtain something "between" which corresponds to a perfect D-latch.



(a) (b)

Figure 8: Typical behaviors which the evolution of a D-latch often converges to: (a) weak logic 1, (b) weak logic 0.

6. Discussion

The presented work has addressed the fundamental question whether the evolutionary approach is able to discover the concept of the discrete state at the transistor level. Although the resulting circuits do not work perfectly, the answer is positive, i.e. the transistors available for the evolutionary design can be composed together by means of an automated evolutionary process in order to establish a simple sequential circuit. No surprise that it was easier to evolve the RS circuit than D-latch. The resulting circuits are not area-optimal, they do not probably operate correctly for various time domains and they can not easily be connected to some other circuits. However, those features were not required. We supplied sufficient resources and the evolution was able to discover the crucial concept – the internal state – directly in the reconfigurable transistor array.

In case of combinational circuits there are usually many options how to put the available components together to obtain the required behavior. It seems that only several options exist for the sequential circuits. Their connection is very tricky and difficult to discover. This is why we were not able to evolve these circuits routinely. Although we consider the method used in Section 5 as the evolution from scratch, we had in fact to supply some little domain knowledge in the form of "promoted design pattern" (i.e. the connection of external wires etc.). No sequential circuits were evolved without this domain knowledge.

In usual CMOS logic gates, upper PMOS circuits and lower NMOS circuit have dual relationship (see Fig. 3). This relationship prevents shorting of power supply to ground and resulted in lower power consumption and saturated output voltage. On the other hand, evolved circuits seem to have turned-on PMOS and NMOS that short power supply and ground in somewhere because output voltage sometimes does not fully saturate to VDD or ground level. Introducing such dual restriction to the transistor level circuits would effectively narrow the search space and

might results in successful synthesis. This is one of possible directions for future research.

7. Conclusions

Simple sequential circuits were evolved at the transistor level directly in FPTA. That means that the concept of discrete state was discovered automatically by means of an evolutionary design process. Unfortunately, we were not able to connect the evolved circuits together in order to create more complicated sequential circuits. Further research is needed to find reliable sequential modules at the transistor level.

Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration (NASA).
Lukas Sekanina was supported by the Fulbright scholarship and from the research project of the Grant Agency of the Czech Republic under No. 102/03/P004 *Evolvable hardware based application design methods*.

References

[1] R. Zebulum, M. Pacheco and M. Vellasco. "Evolutionary Electronics – Automatic Design of Electronic Circuits and Systems by Genetic Algorithms". CRC Press, Boca Raton 2002
[2] A. Stoica, R. S. Zebulum, M. I. Ferguson, D. Keymeulen and V. Duong. "Evolving Circuits in Seconds: Experiments with a Stand-Alone Board Level Evolvable System". In Proc of the 2002 NASA/DoD Conference on Evolvable Hardware, Alexandria Virginia, USA, IEEE Computer Society, p. 67-74, 2002
[3] J. Langeheine, K. Meier, and J. Schemmel. "Intrinsic Evolution of Quasi DC Solutions for Transistor Level Analog Electronic Circuits Using a CMOS FPTA Chip". In Proc. of the 2002 NASA/DoD Conference on Evolvable Hardware, IEEE Computer Society, p. 75-84, 2002
[4] P. Layzell. "A New research Tool for Intrinsic Hardware Evolution". In Proc. of Evolvable Systems: From Biology to Hardware Conference, ICES 1998, LNCS 1478, Springer-Verlag, p. 47-56, 1998
[5] B. Ali, A. Almaini, and T. Kalganova. "Evolutionary algorithms and their use in the design of sequential logic circuits". Genetic Programming and Evolvable Machines, Vol. 5, No. 1, p. 11-29, 2004
[6] J. Wakerly. "Digital Design: Principles and Practices". Prentice-Hall, London 2000
[7] Ch. Manovit, Ch. Aporntewan and P. Chongstitvatana. "Synthesis of Synchronous Sequential Logic Circuits from Parital Input/Output Sequences". In Proc. of Evolvable Systems: From Biology to Hardware Conference, ICES 1998, LNCS 1478, Springer-Verlag, p. 98-105, 1998
[8] A. Thompson. "Silicon Evolution". In Proc. of the Genetic Programming Conference, MIT Press, p. 444-452, 1996
[9] J. M. Tour. "Molecular Electronics". World Scientific, 2003
[10] A. Thompson. „Evolutionary Design of Single Electron Systems". In Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware,
Palo Alto, CA, IEEE Computer Society, 109-116, 2000
[11] M. Garvie and A. Thompson. "Evolution of Combinational and Sequential On-line Self-Diagnosing Hardware". In Proc. of the 2003 NASA/DoD Conference on Evolvable Hardware, IEEE Computer Society, p. 167-173, 2003
[12] J. Lohn, G. Larchev and R. DeMara. "A Genetic Representation for Evolutionary Fault Recovery in Virtex FPGAs". In Proc of the Evolvable Systems: From Biology ti Hardware Conference, ICES 2003, LNCS 2606, p. 47-56, 2003
[13] R. Thomson and T. Arslan. "Evolvable Hardware for the Generation of Sequential Filter Circuits". In Proc. of the 2002 NASA/DoD Conference on Evolvable Hardware, IEEE Computer Society, p. 17-25, 2002