

Evolutionary Design of Polymorphic Gates Using Ambipolar Transistors

Jan Nevoral

Brno University of Technology
Faculty of Information Technology
Czech Republic
Email: ineval@fit.vutbr.cz

Richard Ruzicka

Brno University of Technology
Faculty of Information Technology
Czech Republic
Email: ruzicka@fit.vutbr.cz

Vojtech Mrazek

Brno University of Technology
Faculty of Information Technology
Czech Republic
Email: imrazek@fit.vutbr.cz

Abstract—The objective of the paper is to introduce a new approach to the evolutionary design of polymorphic digital circuits conducted directly at transistor level. A discrete event-driven simulator was utilized to achieve reasonable trade-off between performance and precision. The proposed approach was evaluated on a set of polymorphic logic circuits controlled by switching the power rails. It was demonstrated that the proposed method is able to produce valid solutions. A lot of polymorphic gates based on ambipolar transistors were designed, which provide transistor savings compared to existing circuits. A new class of polymorphic gates was discovered thanks to the proposed system – gates based on conventional MOS transistors whose functions are changed by switching the power rails. They seem to have the best parameters among currently known polymorphic gates based on conventional transistors.

I. INTRODUCTION

In some situations it would be useful to have one more logic function implemented in a digital circuit that is not needed during normal operation of the circuit, but could be activated occasionally under special circumstances. For example, it could be a test, a watermark, an emergency function or simply any hidden secondary required function. A conventional solution is to attach an additional circuit to the main circuit and simply activate it (or switch the output) when necessary. Another approach that usually brings more efficient and smart solution could be so-called polymorphic or multifunctional electronics.

Polymorphic electronics as an approach to design and implementation of multifunctional digital circuits was proposed by the team of A. Stoica at the NASA JPL about 15 years ago [1]. The main idea behind the Polymorphic electronics is to have a single structure (circuit) that is able to perform more than one logic function. The function just performed by the circuit (only one function should be performed by the circuit of course) depends on the circumstances of the circuit, on the state of the environment. Polymorphic circuits are created from polymorphic logic gates; the change of the whole circuit function is caused by the change of Boolean functions performed by used polymorphic gates. Interconnection of the circuit components (gates) remains unchanged. Therefore, polymorphism does not mean reconfiguration, components just change their behavior [2].

Utilization of the polymorphic electronics concept is limited by the existence of effective implementation of polymorphic gates. Typical environment which affects the function of a polymorphic gate is power supply voltage. The power supply rail cannot be only a means of energy distribution but also another channel for a global information distribution. The main advantage of such a function driver is that all gates already have the power supply voltage connected – no special wiring for a “change function” signal is needed.

In the past, the change of the function was driven by the level of supply voltage. For example, if the V_{dd} line was 3.3V above the ground, all polymorphic gates employed in the circuit perform one certain function. When the V_{dd} was 5V above the ground, polymorphic gates began to perform another function [3]. As these gates were often implemented using conventional CMOS technology, which is elaborated to provide stable behavior of logic gates, results were not always as effective as desired. In fact, these gates were designed as analog circuits with all disadvantages like high power consumption, slow operation etc.

In recent years, the research on polymorphic gates was focused on the utilization of so called post-silicon devices [4]. It seems that some features of post-silicon devices like ambipolarity may play an important role in the field of polymorphic electronics [5]. It enables to implement polymorphic gates using ambipolar devices – devices that are also naturally able to change their behavior. By this, the principle of polymorphism (as intended by Stoica [1] and their followers [3]) would now be moved one step towards more effective implementation of polymorphic circuits. Utilization of ambipolar devices as basic building blocks of polymorphic gates requires another way of power supply driving than the one proposed by Stoica et al. To change the polarity of ambipolar transistors, the polarity of the power supply must be reversed. This way of polymorphic gate function change leads to very efficient and neat implementation of logic gates, very close to the purity of ordinary CMOS logic gates. Parameters of such designed gates are also very promising, as the design is wholly digital – transistors operate as switches in the saturation mode.

Although the power supply rails switching as the way of polymorphism function driving was developed primarily for the gates based on ambipolar transistors, some gates using

only ordinary silicon MOS transistors were also developed and they also seem to be promising.

As it is already obvious from the text above, the design of polymorphic gates (and polymorphic circuits in general) is not a trivial task for a human designer especially due to the fact that more than one function must be kept in mind while the structure of the gate is proposed. Especially due to this fact, most of existing polymorphic gates are results of evolutionary based design [6], [7]. At the time of writing this paper no article describing evolutionary design of polymorphic gates based on ambipolar transistors was published.

In this paper, the evolutionary design of novel polymorphic gates is described and some promising results are shown. The paper is organized as follows: Section II introduces the field of polymorphic electronics and the problematics of polymorphic gates. In Section III, ambipolar transistors as basic blocks of some newly created polymorphic gates are briefly described. Section IV shows how polymorphic gates were evolved using CGP and Section V presents the results of evolution – newly created gates. Section VI concludes the paper.

II. POLYMORPHIC ELECTRONICS

The key to the circuit multifunctionality lies in the components, because these are the devices that change their function. Moreover, the change of the function exhibited by the component may be done by various ways. It may be advantageous if each component has its own sense to a phenomenon, which causes function change and the phenomenon is inherently present in the circuit (like supply voltage level, temperature etc.), then no special signal (distributed by a global interconnection network) is needed to control the function and the implementation of multifunctionality could be very natural.

The set of components typically consists of logic gates, so the structure of a polymorphic circuit is studied on the gate level of abstraction. Just gates make the difference between conventional logic and polymorphic circuits. The main two problems of the polymorphic electronics are:

- The problem of design methods (synthesis) for polymorphic circuits, i.e. how to map a description in the behavioral domain to a description in the structural domain. A lot of polymorphic circuits were designed using evolutionary design methods, especially using Cartesian Genetic Programming [6]. However, the non-evolutionary (conventional) design methods were also proposed [8].
- Search for suitable polymorphic components (gates). Implementation of the gate should be efficient (in terms of occupied chip area or transistor count).

Whereas the first problem is addressed by several papers in recent years and new synthesis methods arose, new gates for polymorphic electronics were published rarely in last years. The reason is perhaps the fact that conventional MOS transistors are very stable devices. Therefore, a trial to develop a multifunctional gate using silicon MOS transistors leads to a complicated structure with inferior parameters. Fortunately, new ambipolar devices promise development of new efficient

TABLE I
A SURVEY OF EXISTING SILICON-BASED CMOS POLYMORPHIC GATES.

Gate	Control	Ctrl'd. by	Size	Ref.
NAND/NOR	3.3/1.8 V	V_{dd}	6	[2]
AND/OR	1.2/3.3 V	V_{dd}	8	[7]
NAND/NOR	5/3.3 V	V_{dd}	8	[3]
AND/OR	27/125 °C	temperature	6	[1]
AND/OR	5/90 °C	temperature	8	[1]
NAND/NOR	0/5 V	ext. voltage	10	[9]
NAND/NOR	5/0 V	ext. voltage	8	[10]
NAND/NOR	5/0 V	ext. voltage	10	[10]
NAND/XOR	5/0 V	ext. voltage	9	[10]
AND/OR	0/3.3 V	ext. voltage	6	[1]
AND/OR/XOR	3.3/1.5/0 V	ext. voltage	9	[1]
NAND/NOR	0/5 V	ext. voltage	10	[6]

and graceful gates with potentially very good electrical parameters and the research of them enables also a development of new gates based on ordinary silicon MOS transistors. So this paper contributes to the research just in the second of the two problems mentioned above.

A polymorphic gate is an element which realizes an elementary Boolean function, whereas the function may vary (for the same element) in accordance with the particular state of the environment. It is possible to say that the function of the gate is controlled by the environment. Such feature may be useful for variety of applications and may save the chip area and significantly reduce the global interconnections.

If the gate exhibits e.g. NAND function for some range of the power supply voltage (V_{dd}) and e.g. NOR function for another range of the V_{dd} , the gate could be specified as a NAND/NOR gate controlled by V_{dd} . It is assumed that polymorphic gate may perform no more than one function with respect to any particular instant during the course of time. Due to the fact that gate has just one output port available, it may deliver only single value from all the possible options at a time.

Table I surveys the polymorphic gates reported in literature. For each polymorphic gate, the logic functions performed by the gate are given together with recommended setting of the control signal variable. The number of transistors characterizes the size of polymorphic gates only partially (transistors occupy different areas, gates were fabricated using different fabrication technology).

Only two of the polymorphic gates have been fabricated so far; remaining polymorphic gates were either simulated or tested in a field programmable transistor array (FPTA-2). For instance, the 6-transistor NAND/NOR gate controlled by V_{dd} was fabricated in a 0.5-micron HP technology [2]. Another NAND/NOR gate controlled by V_{dd} and introduced in [3] was utilized in the REPOMO32 chip.

III. AMBIPOLAR TRANSISTORS

In the literature it was reported that many post-silicon devices developed in recent years exhibit so called ambipolar

behavior. The principle behind the ambipolar behavior of a transistor is that the transistor operates like P-type conventional MOS transistor under certain conditions, but for example a corresponding adjustment of gate bias triggers its transition into N-type mode – it changes the behavior and operates like N-type conventional MOS transistor. This behavior seems to be very advantageous for complementary (CMOS like) structures and the fabrication of such structures may be very simplified – just one type of the transistor is used everywhere. The polarity of transistors in the structure (if the transistor is of P- or N-type) is then determined by the gate bias or by a special electrode [11].

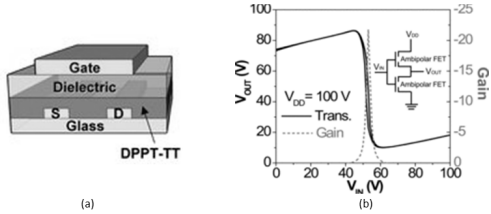


Fig. 1. (a) Structure of ambipolar transistor from [12], (b) typical step response and gain of ambipolar inverter formed by two identical transistors.

Figure 1 shows a scheme of an ambipolar transistor, which is formed from diketopyrrolopyrroles-Thieno[3,2-b]thiophene copolymer [12]. But what is more important from the utilization in polymorphic structures is the typical step response and a gain of ambipolar inverter formed by two such transistors, also shown in Figure 1. Thanks to ambipolarity one transistor acts as a P-type transistor and the second one as an N-type, although they are both manufactured as identical devices.

The fact that the polarity of the ambipolar transistor is not determined by the fabrication process, but it could be changed by the gate bias (position of the transistor in the circuit towards supply voltage poles) or by a special electrode, should be used in development of polymorphic gates. Note that the inverter depicted in Figure 1 may exhibit the same behavior for both possible polarities of the power supply. This is because the inverter is symmetric and both transistors are the same. For non-symmetric and more complex gate structures (e.g. those aptly developed by an evolutionary algorithm) the behavior should be apparently more interesting.

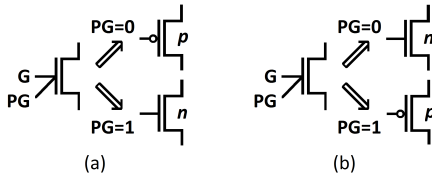


Fig. 2. Behavior of ambipolar transistor (a) type 1, (b) type 2.

The evolutionary design described in this document is based on ambipolar transistors with special electrode known as polarity gate (PG). These transistors have been reported in several emerging technologies such as carbon nanotubes (CNTs), graphene and silicon nanowires (SiNWs) [13].

The behavior of ambipolar transistors vary according to the used technologies [13], [14]. It can be divided into two different categories:

- **Type 1:** These transistors act as N-type in case of logic 1 at PG pin and P-type in case of logic 0 – Figure 2a).
- **Type 2:** These transistors act as P-type in case of logic 1 at PG pin and N-type in case of logic 0 – Figure 2b).

IV. EVOLUTIONARY DESIGN OF GATES

Several polymorphic gates using ambipolar transistors were already designed [15], but no systematic design methods were ever described. Moreover, only few papers were devoted to evolution of small digital circuits directly at transistor level in general utilizing standard p-MOS and n-MOS devices.

Zaloudek et al. published an approach based on a simple simulator which was designed to quickly evaluate the candidate solutions [6]. Unfortunately, a rough approximation of transistor behavior caused that this approach produced many incorrectly working circuits. Trefzer used another technique to evolve some basic logic gates [16]. Instead of using a time consuming analog circuit simulator, a reconfigurable analog transistor array was employed. However, it was shown that many of the discovered solutions relied on some properties of the utilized reconfigurable array. About 50% of the evolved circuits failed in the analog simulation. Walker et al. used a different technique to evolve transistor-level circuits [17]. In order to speed up the time consuming evaluation of candidate solutions, a cluster of SPICE-based simulators was utilized. Even if it was possible to evolve correct solutions, only small problem instances could be investigated due to the overhead of SPICE simulators. In order to obtain circuit working in the analog simulation in reasonable time, Mrazek and Vasicek proposed a discrete simulator with a switch-level transistor model extended by a threshold drop degradation effect to achieve a fast simulation with reasonable accuracy [18]. In this paper, such evolutionary method was proposed. The proposed approach reflects a different functionality of the ambipolar transistors and utilizes a new circuit representation and simulation.

A. Circuit representation

In order to evolve polymorphic circuits at the transistor level a suitable representation enabling to encode bidirectional graph structures containing junctions is needed. We utilized a Cartesian genetic programming (CGP) proposed by J. Miller [19].

The proposed representation proceeds from CGP representation of gate-level circuit. It is defined as follows. Each polymorphic digital circuit having n_i primary inputs and n_o primary outputs (i.e. a candidate solution) is represented using an array of nodes arranged in n_c columns and n_r rows. Each node consists of three source terminals and one output terminal. Each node can act as p-MOS transistor, n-MOS transistor, ambipolar transistor or a junction. Ambipolar transistor uses all three source terminals, whereas the rest of nodes two source terminals only. The utilized nodes are

shown in Figure 3. Source terminals of each node can be independently connected to the output terminal of any node placed in previous l_{back} columns. In addition to that, source terminals of each node can be connected to one of the primary circuit inputs.

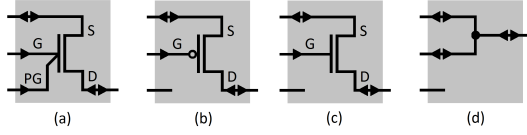


Fig. 3. Basic building blocks of transistor-level circuits: (a) ambipolar transistor, (b) p-MOS transistor, (c) n-MOS transistor and (d) junction that combines three signals together. If a proper voltage is applied on the gate electrodes denoted as G and PG, transistor connects its source electrode (S) with drain (D). The arrows denote the possible directions of signal flow which have to be considered during the evaluation.

The junction nodes combine two input signals and one output signal together. As a consequence of that, loops and multiple connections are natively supported.

The following integer encoding scheme so-called chromosome is utilized. The primary inputs are labeled from 0 to $n_i - 1$ and the node outputs from n_i to $n_i + n_c \cdot n_r - 1$. A candidate solution is represented in the chromosome by $n_c \cdot n_r$ quaternion (x_1, x_2, x_3, f) determining for each node its function $f \in \Gamma$, where $\Gamma = \{0^{ambipolar}, 1^{p-mos}, 2^{n-mos}, 3^{junction}\}$ is a function encoding, and label of nodes x_1, x_2 and x_3 connected to the input terminals. The last part of the chromosome contains n_o labels of nodes where the n_o primary outputs are connected to. Note that the power supply rails are explicitly encoded as the first two primary inputs.

Figure 4 demonstrates the principle of utilized encoding on a polymorphic inverter circuit which inverts the logical value of input signal independently of power rails switching. The shown chromosome encodes a candidate circuit using four nodes. However, only three of them contribute to the phenotype and are active.

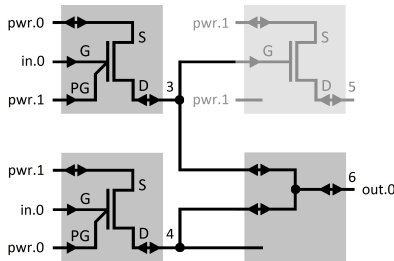


Fig. 4. Example of an encoding of a candidate circuit implementing polymorphic inverter using two ambipolar transistors. Parameters are as follows: $n_i = 3$ ($pwr.0, pwr.1, in.0$), $n_o = 1$ ($out.0$), $n_c = n_r = l_{back} = 2$. The integer chromosome is $(0, 2, 1, 0)(1, 2, 0, 0)(1, 3, 1, 2)(3, 4, 4, 3)(6)$.

B. Transistor discrete model

As stated in Section III, the research of ambipolar transistors started a few years ago. As a consequence of that, no exact models are still available. Therefore, we created two discrete

models of ambipolar transistor according to the expected behavior.

The first model of ambipolar transistor utilizes standard MOS transistor behavior extended to a special PG pin which switches the transistor polarity. The transistor behavior corresponds to the TSMC technology $\lambda = 0.25 \mu m$ used by Mrazek and Vasicek [18]. The MOS model is based on the switch-level transistor model extended to a threshold drop degradation effect, it is abstracted from dynamic parameters such as power consumption or delay. The simulation uses six voltage levels: logic 0 (denoted as ‘0’), logic 1 (‘1’), degraded 0 (‘L’), degraded 1 (‘H’), high impedance (‘Z’) and undefined value (‘X’). The behavior of n-MOS and p-MOS transistors is defined by Table II, which determines the drain level value from level values of gate and source. The ambipolarity of the transistor is obtained by the state of the PG pin. We defined two types of the model: type 1 is defined as follows: (a) If value at PG pin is ‘1’ or ‘H’, transistor acts as n-MOS, (b) if value at PG pin is ‘0’ or ‘L’, transistor acts as p-MOS (c) otherwise drain level voltage is ‘X’. Similarly, the behavior of ambipolar transistor type 2 is defined as follows: (a) If value at PG pin is ‘1’ or ‘H’, transistor acts as p-MOS, (b) if value at PG pin is ‘0’ or ‘L’, transistor acts as n-MOS (c) otherwise drain level voltage is ‘X’. In this paper, polymorphic circuits evolved using this model are called as circuits with six-state logic.

TABLE II
BEHAVIOR OF N-MOS AND P-MOS TRANSISTORS MODELED USING SIX DISCRETE VALUES.

		n-MOS						p-MOS											
gate		source						gate		source									
		1	H	L	0	Z	X	1	H	L	0	Z	X	1	H	L	0	Z	X
1	H	X	L	0	Z	X	1	Z	Z	Z	Z	Z	X	H	Z	Z	Z	Z	X
H	X	X	L	0	Z	X	H	Z	Z	Z	Z	Z	X	L	1	H	X	X	Z
L	Z	Z	Z	Z	Z	X	L	1	H	X	X	Z	X	0	1	H	X	Z	X
0	Z	Z	Z	Z	Z	X	0	1	H	X	Z	Z	X	Z	Z	Z	Z	Z	X
Z	Z	Z	Z	Z	Z	X	Z	Z	Z	X	Z	Z	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Table III defines the n-MOS and p-MOS transistors behavior of the second model. This model is stricter than the first one – the transistor does not have to recognize the degraded value at G and PG pins well. The degraded values are not propagated from source to drain. The behavior of ambipolar transistor type 1 is defines as follows: (a) If value at PG pin is ‘1’, transistor acts as n-MOS, (b) if value at PG pin is ‘0’, transistor acts as p-MOS (c) otherwise drain level voltage is ‘X’. Similarly, the behavior of ambipolar transistor type 2 is defined as follows: (a) If value at PG pin is ‘1’, transistor acts as p-MOS, (b) if value at PG pin is ‘0’, transistor acts as n-MOS (c) otherwise drain level voltage is ‘X’.

The second model utilizing six logic values can be optimized to four-state logic as follows. When the value at both G and S pins is ‘1’, the value of n-MOS drain is ‘H’ (similarly, ‘0’ at both gate and source terminals of p-MOS produces ‘L’ at drain). These degraded values cannot be used in the evolved

TABLE III

BEHAVIOR OF N-MOS AND P-MOS TRANSISTORS MODELED USING SIX DISCRETE VALUES. THIS BEHAVIOR IS USED TO GET FOUR-STATE LOGIC.

n-MOS		p-MOS											
gate	source						gate	source					
	1	H	L	0	Z	X		1	H	L	0	Z	X
1	H	X	X	0	Z	X	1	Z	Z	Z	Z	Z	X
H	X	X	X	X	Z	X	H	X	X	X	X	Z	X
L	X	X	X	X	Z	X	L	X	X	X	X	Z	X
0	Z	Z	Z	Z	Z	X	0	1	X	X	L	Z	X
Z	Z	Z	Z	Z	Z	X	Z	Z	Z	Z	Z	Z	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X

circuit, because the transistors do not accept these values and a degraded value at circuit primary output is also undesirable. The only one possibility of occurrence is a junction of the degraded and non-degraded value. As a consequence, more circuits can be evolved by this model than by a pure four-state model. Because the transistors are not controlled by degraded values, polymorphic circuits evolved using this model are called as circuits with four-state logic in this paper.

C. Evaluation of the candidate solutions

The goal of the evaluation is to determine that the candidate circuit meets the requirements i.e. there is no constraint violated and the circuit works correctly w.r.t. definition. Evaluation of the candidate solutions encoded using the proposed representation consists of two steps.

Firstly, set of active nodes is determined. This operation is performed due to speed optimization and because of skipping of exceptions like short circuits in the unused part of the circuit. Only the active nodes represent (i.e. they are in a path from input nodes to outputs) the evaluated circuit. Inactive nodes are ignored. A node is active if either (a) its output is connected to any of the primary outputs, or (b) its output is connected to an active node, or (c) it is a junction node whose at least one source terminal is connected to an active node.

Then, multi-level discrete event-driven simulator is utilized to determine the response for each input combination. The advantage of this approach is that only necessary nodes are updated if there is a change of a value.

The following steps are used to determine output value for a given input combination. Firstly, outputs of all nodes are initialized to the value 'Z'. Then, values 0 and 1 (according to the currently simulated input combination) are assigned to the primary inputs. This change triggers re-evaluation of all the nodes connected directly to primary inputs. Each node determines its new output value and propagates it to all related nodes. As an open transistor connect source with drain, bidirectional data-flow have to be utilized. It means that the new value must be propagated to the nodes connected not only to the drain but also to source terminal. Similarly, junctions have to propagate the new value to all terminals. The new value of a junction node is calculated as the strongest value presented on all the terminals. The new value of a transistor node is determined according to the value connected to the source as

well as drain. During the evaluation of a new output value of a transistor node, the new calculated value is compared with current value at drain terminal. If the values are not compatible, short circuit exception is raised. Otherwise, the stronger value is propagated to all related nodes. The relations between the discrete values are as follows: 'Z' < 'L' < '0' < 'X'; 'Z' < 'H' < '1' < 'X'. It means that if at least one of the values is equal to 'X', 'X' is propagated to all related nodes.

D. Search strategy

The overall CGP design algorithm is shown in Algorithm 1. As a search algorithm, $1 + \lambda$ evolutionary strategy is utilized [19]. The initial population is randomly generated. Every new population consists of the best individual and λ offspring created using a point mutation operator which modifies h randomly selected genes. When two or more best individuals of the population receive the same fitness as the highest score of the previous population, one of the best individuals which did not serve as a parent in the previous population is randomly selected as a new parent. This strategy is used to ensure the diversity of population. The evolution is terminated when a predefined number of generations is reached.

Algorithm 1: CGP design

Input: CGP parameters, fitness function
Output: The highest scored individual p and its fitness

- 1 $P \leftarrow$ randomly generated p and its λ offspring;
- 2 EvaluatePopulation(P);
- 3 **while** *(terminating condition not satisfied)* **do**
- 4 $\alpha \leftarrow$ highest-scored-individual(P);
- 5 **if** $fitness(\alpha) \geq fitness(p)$ **then**
- 6 $p \leftarrow \alpha$;
- 7 $P \leftarrow$ create λ offspring of p using mutation;
- 8 EvaluatePopulation(P);
- 9 **return** p , $fitness(p)$;

Quality of each candidate solution is determined by the *fitness function*. For evolution of polymorphic circuits, all possible input combinations have to be applied at the candidate circuit inputs. The obtained output values are compared with corresponding required truth table and the goal is to minimize the Hamming weight. The fitness value is constructed as follows: If an obtained output value corresponds to the expected one, 5 points are added to the fitness value. If the calculated value exhibits the same polarity but represents degraded voltage, 2 points are used. Otherwise, no point is added because the response is invalid. If there is a short-circuit exception during the simulation, the simulation is terminated and 3 points penalty is subtracted from the total fitness value. Similarly, if the simulator exceeds the predefined number of steps (i.e. node outputs are not in stable state), the simulation is terminated and the fitness value is penalized by 3 points.

Evolved circuits can connect the primary input signals through the transistors directly to the primary outputs. Such

circuits does not provide high input impedance and low output impedance, which is required. In order to avoid this fact, we defined two types of constraints for evolved circuits:

- *High-Impedance (HI)* - Primary signal inputs can be routed to source and drain transistor terminals, but no input combination can cause connecting of any primary input signal to primary output of the evolved circuit. This constraint is for *high input-impedance and low output-impedance*.
- *Extra High-Impedance (Extra-HI)* - Primary signal inputs can be routed to gates and polarity gates of transistors only. This constraint is used for the same impedance requirements, but it is stricter than previous one.

As soon as a fully working solution is found, additional requirements (like e.g. high input impedance and low output impedance) for circuit properties can be checked. Only a fully working candidate circuit with desired properties may replace the current best individual.

As soon as a fully working solution with desired properties is found, i.e. it meets the specification, optimization of circuit size starts. Only a fully working candidate circuit with desired properties and equal or less utilized transistor may replace the current best individual.

V. EVOLVED GATES

The proposed method was evaluated in the evolution of basic polymorphic logic circuits controlled by switching the power rails. The goal of the experiments was to evolve fully functional implementations of 36 polymorphic circuits exhibiting full voltage swing on the outputs. These circuits were evolved with different transistor types – (a) MOSFET transistors only, (b) ambipolar transistors only and (c) MOSFET and ambipolar transistors. In addition to that, the impact of High-Impedance and Extra High-Impedance constraints were investigated. Finally, the ability of evolutionary algorithm to find the solution is demonstrated.

The results were obtained from more than 300 independent runs for each combination of polymorphic gates, used types of transistors and additional constraints using the following experimental setup: CGP matrix size: 5×5 , $l_{back} = 5$, mutation: $3 \simeq 4\%$, number of generations: 2,000,000.

A. Polymorphic gates with MOSFET and ambipolar transistors

Firstly, it was necessary to find out, which polymorphic gates we are able to evolve with six-state logic transistor model without any constraints. We chose all 36 polymorphic gates created as a product of identity (denoted as ID), negation (NOT) and basic two input gates (AND, NAND, OR, NOR, XOR, XNOR). Note that ID/ID gate marks polymorphic multiplexer and NOT/NOT marks polymorphic inverter. As you can see count of transistors in Table IV, all evolved polymorphic gates using six-state logic and both MOSFET and ambipolar transistors type 1 were found. Moreover, the most difficult OR/OR gate needs 9 transistors only.

TABLE IV
SIZE OF THE SMALLEST FOUND SOLUTION OF POLYMORPHIC GATES f_1/f_2 UTILIZING MOSFET AND AMBIPOLAR TRANSISTORS TYPE 1

f_1	f_2							
	ID	NOT	AND	NAND	OR	NOR	XOR	XNOR
ID	4	4	7	6	7	6	8	8
NOT		2	6	6	6	6	7	7
AND			5	7	5	7	7	7
NAND				4	7	4	7	7
OR					9	7	7	7
NOR						4	7	7
XOR							6	4
XNOR								6

When the circuits are designed, it is appropriate to use the same types of transistors and not to combine different technologies, because the combination of technologies complicates the manufacture of the circuit.

B. Polymorphic gates with MOSFET transistors only

As it was written in Section II, several polymorphic gates using MOSFET transistors were already published. These gates were analog designs with all failings like high power consumption, slow operation etc. However, digital approach to polymorphic circuits based on MOSFET have not ever been presented.

Table V shows that a lot of such polymorphic gates based on TSMC 0.25 μm transistors can be created. We can see that it is more difficult to find polymorphic gates using MOSFET transistors only, because the found solution utilize more transistors. Moreover AND/OR and NAND/NOR gates were not found in any of 300 runs.

TABLE V
SIZE OF THE SMALLEST FOUND SOLUTION OF POLYMORPHIC GATES f_1/f_2 UTILIZING MOSFET TRANSISTOR ONLY

f_1	f_2							
	ID	NOT	AND	NAND	OR	NOR	XOR	XNOR
ID	4	4	7	7	7	7	8	8
NOT		6	6	8	6	8	10	10
AND			9	8	-	9	12	10
NAND				8	9	-	10	12
OR					9	8	10	11
NOR						8	11	10
XOR							9	12
XNOR								9

C. Polymorphic gates with ambipolar transistors only

Many digital circuits based on ambipolar transistors have already been published [14]. However, only few polymorphic circuits composed of these transistors were designed. Yang et al. presented NAND/NOR and XOR/XNOR gates [15]. Both gates use 4 transistors only. However, the second one expects the presence of both input signal negation. Therefore, 4 transistors and 2 inverters are needed (i.e. 8 ambipolar transistors in total). AND/OR gate have not ever been presented, but it can be simply assembled using 6 transistors from NAND/NOR gate and polymorphic inverter.

TABLE VI
SIZE OF THE SMALLEST SOLUTION OF SELECTED POLYMORPHIC GATES
UTILIZING AMBIPOLAR TRANSISTOR ONLY

Gate	Constraint	Type 1 6-state	Type 1 4-state	Type 2 6-state	Type 2 4-state
NOT/NOT	-	2	2	2	2
NOT/NOT	Extra-HI	2	2	2	2
NAND/NOR	-	4	4	4	4
NAND/NOR	Extra-HI	4	4	4	4
AND/OR	-	5	5	3	3
AND/OR	Extra-HI	6	6	4	4
XOR/XNOR	-	4	5	5	5
XOR/XNOR	Extra-HI	6	8	5	6

We also tried to evolve polymorphic circuits based on ambipolar transistors only. Four polymorphic gates from 36 mentioned above were successfully evolved: NAND/NOR, AND/OR, XOR/XNOR and polymorphic inverter. Table VI shows the minimal transistor count needed for design of these gates. As you can see, Extra-HI constraint causes increase of utilized transistors. Similarly, 4-state logic uses equal or more resources.

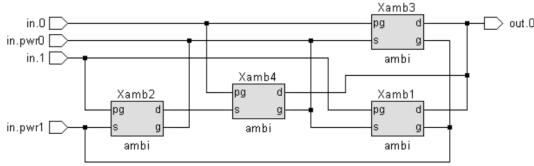


Fig. 5. AND/OR polymorphic gate with Extra-HI restriction based on ambipolar transistors type 2.

All the evolved gates need equal or less ambipolar transistors in comparison with gates mentioned above. Evolved inverter and NAND/NOR gate do not provide any transistor savings compared to the known solutions. However, various combinations of transistor models and impedance constraints provide polymorphic gates with significantly lower number of ambipolar transistors. Four transistors are needed to design the XOR/XNOR gate and even just three transistors are needed to design the AND/OR polymorphic gate.

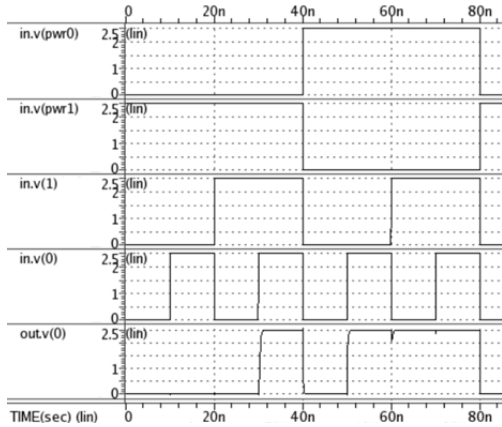


Fig. 6. HSPICE simulation result of polymorphic and/or gate from Figure 5.

As an example, Figure 5 shows the AND/OR polymorphic circuit with Extra-HI constraint controlled by switching the power rails. It is composed of four ambipolar transistor type 2.

All discovered circuits were verified using a HSPICE simulator. Nowadays, HSPICE model for ambipolar transistor with special electrode is not available. Ambipolar behavior was emulated by a circuit composed of two MOSFET transistors, two transmission gates and one inverter. All the circuits were valid and operated correctly. Figure 6 shows the HSPICE simulation results of the AND/OR gate from Figure 5.

D. Space-Search method results

The aim of the paper is to prove that modified CGP is able to design a new polymorphic circuits utilizing models of ambipolar transistors. In previous sections, we have shown that the proposed method have found working solutions for various types of transistors.

Figure 7 describes the following parameters during the evolution of all 300 independent runs of design. First parameter is size of fully functional solutions. We observe minimal, maximal and average count of transistors. We also observe the success rate, which represents how many runs have found fully functional circuit that meet the specified constraint in current generation.

We can see that Extra-HI constraint reduces the performance of the evolution. Substantially higher number of generations is needed to achieve the same success rate. Moreover, when a new working solution is found, the size quickly decreases to the minimal one (see the width of size-peeks). In addition to that XOR/XNOR gate is a more difficult problem to solve than AND/OR.

Interestingly, when we compare unconstrained runs and different types of models, the 4-state logic provides worse success rate shape than the 6-state logic. Moreover, ambipolar transistor type 2 provides better success rate than type 1 – approx. twice more generations are needed to obtain the same success rate. However, all configurations tend towards the same results, but the speed is different.

VI. CONCLUSION

A new approach suitable to the evolutionary design of polymorphic digital circuits conducted directly at transistor level was introduced in this paper. In order to improve the time consuming evaluation of candidate solutions, a discrete event-driven simulator was utilized. The proposed simulator operates on multiple logic levels to achieve reasonable trade-off between performance and precision. Two discrete voltage level models of polymorphic logic were presented, which have a potential to be used in many other simulations of polymorphic circuits. The goal of the evolutionary algorithm is to design a circuit having the minimal number of transistors. In addition to that, various constraints on the evolved circuits are investigated to increase the success rate of the evolutionary design.

The proposed approach was evaluated in the evolution of polymorphic logic circuits controlled by switching the power

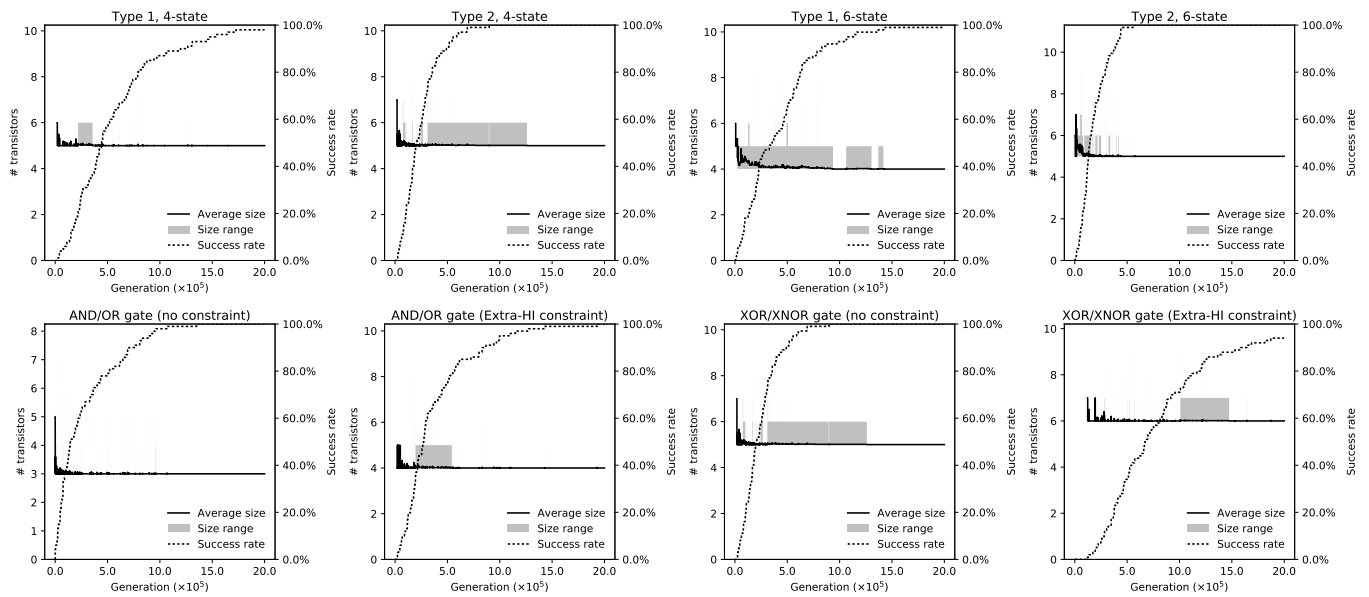


Fig. 7. Success rate and circuit size (minimal, maximal and average). First row shows values for various ambipolar models for XOR/XNOR unconstrained runs. The second row shows results for impedance constraints used in evolution of AND/OR and XOR/XNOR ambipolar circuits utilizing 4-value logic type 2 ambipolar transistors.

rails. It was demonstrated that the proposed method is able to produce valid solutions. A lot of polymorphic gates controlled by switching the power rails and utilizing ambipolar transistors were designed, which provide transistor savings compared to existing circuits.

Thanks to the proposed system, a new class of polymorphic gates was discovered – gates based on conventional MOS transistors whose functions are changed by switching the power rails. They seem to have the best parameters among currently known polymorphic gates based on conventional transistors. This class of polymorphic gates may allow to use the polymorphic electronics also in existing chips (f. e. when additional PUF, watermark or hidden diagnostics is required) without any change of chip manufacturing technology.

ACKNOWLEDGMENT

This work was supported by the national COST grant Unconventional Design Techniques for Intrinsic Reconfiguration of Digital Circuits: From Materials to Implementation (no. LD14055).

REFERENCES

- [1] A. Stoica, R. Zebulum, and D. Keymeulen, "Polymorphic electronics," in *Int. Conf. on Evolvable Systems*. Springer, 2001, pp. 291–302.
- [2] A. Stoica, R. Zebulum *et al.*, "Taking evolutionary circuit design from experimentation to implementation: Some useful techniques and a silicon demonstration," *IEEE Proc. Computers and Digital Techniques*, vol. 151, no. 4, pp. 295–300, 2004.
- [3] R. Růžička, L. Sekanina, and R. Prokop, "Physical demonstration of polymorphic self-checking circuits," in *Proc. of the 14th IEEE Int. On-Line Testing Symposium*. IEEE Computer Society, 2008, pp. 31–36.
- [4] J. M. Rabaey and S. Malik, "Challenges and solutions for late- and post-silicon design," *IEEE Design Test of Computers*, vol. 25, no. 4, pp. 296–302, July 2008.

- [5] R. Růžička and R. Tesař, "Lets move polymorphism downwards: On the multifunctional logic based on ambipolar behaviour of semiconductor devices," in *Proc. 11th Int. Conf. Design & Technology of Integrated Systems in Nanoscale Era*, 2016, pp. 275–279.
- [6] L. Žaloudek and L. Sekanina, "Transistor-level evolution of digital circuits using a special circuit simulator," in *Evolvable Systems: From Biology to Hardware*, ser. LNCS 5216. Springer Verlag, 2008, pp. 320–331.
- [7] A. Stoica, R. Zebulum *et al.*, "On polymorphic circuits and their design using evolutionary algorithms," 2002.
- [8] A. Crha, R. Růžička, and V. Šimek, "Synthesis methodology of polymorphic circuits using polymorphic nand/nor gates," in *Proc. UKSim-AMSS 17th Int. Conf. Computer Modelling and Simulation*, 2015, pp. 612–617.
- [9] M. McDermott and J. Turner, "Configurable nand/nor element," Jan. 7 1997, uS Patent 5,592,107. [Online]. Available: <http://www.google.com/gh/patents/US5592107>
- [10] R. Růžička, "On bifunctional polymorphic gates controlled by a special signal," *WSEAS Trans. on Circuits*, vol. 7, no. 3, pp. 96–101, 2008.
- [11] A. Heinzig, S. Slesazek *et al.*, "Reconfigurable silicon nanowire transistors," *Nano Letters*, vol. 12, no. 1, pp. 119–124, 2012, pMID: 22111808.
- [12] Z. Chen, M. Lee *et al.*, "High-performance ambipolar diketopyrrolopyrrole-thieno[3,2-b]thiophene copolymer field-effect transistors with balanced hole and electron mobilities," *ADVANCED MATERIALS*, vol. 24, pp. 647–+, 2012.
- [13] O. Turkyilmaz, F. Clermidy *et al.*, "Self-checking ripple-carry adder with ambipolar silicon nanowire fet," in *2013 IEEE Int. Symp. Circuits and Systems*, May 2013, pp. 2127–2130.
- [14] M. H. Ben-Jamaa, K. Mohanram, and G. D. Micheli, "An efficient gate library for ambipolar cntfet logic," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 2, pp. 242–255, Feb 2011.
- [15] X. Yang and K. Mohanram, "Ambipolar electronics," 2010.
- [16] M. Trefzer, "Evolution of Transistor Circuits," Ph.D. dissertation, Ruprecht-Karls-Universitt Heidelberg, 2006.
- [17] J. Walker, J. Hilder, and A. Tyrrell, "Evolving variability-tolerant cmos designs," in *Evolvable Systems: From Biology to Hardware*, ser. LNCS, vol. 5216. Springer Berlin Heidelberg, 2008, pp. 308–319.
- [18] V. Mrazek and Z. Vasicek, "Evolutionary design of transistor level digital circuits using discrete simulation," in *Proc. of European Conf. on Genetic Programming*, ser. LCNS 9025, 2015, pp. 66–77.
- [19] J. F. Miller, *Cartesian Genetic Programming*. Springer Verlag, 2011.