Digital circuits test optimization by multifunctional components

Lukáš Stareček*
Department of computer systems
Faculty of Information Technology
Brno University of Technology in Brno
Božetěchova 1/2, 612 66 Brno, Czech Republic
starecek@fit.vutbr.cz

Abstract

This paper deals with the possibility of digital circuits test optimization using multifunctional logic gates. The most important part of thesis is explanation of the optimization principle. Based on this principle, the work presented several options to use. The optimization of testability analogous to inserting test points and simple methodology based on SCOAP is shown. The focus of work is a methodology created to optimize circuit test. It was implemented in the form of software tools. In the work are presented the results of using these tools to reduce test vectors volume while maintaining fault coverage on various circuits, including circuits from ISCAS 85 test set.

Part of the work is devoted to the various principles and technology of creating of multifunctional logic gates. Some selected gates of these technologies are subject to simulations of electronic properties in SPICE. Based on the principles of presented methodology and results of simulations of multifunctional gates is also made an analysis of various problems such as validity of the test of modified circuit and the suitability of each multifunctional gate technology for the methodology.

Categories and Subject Descriptors

B.6.2 [**Logic design**]: Reliability and Testing; B.7.3 [**Integrated circuits**]: Reliability and Testing; I.6.m [**Simulation and modeling**]: Miscellaneous; G.1.6 [**Numerical analysis**]: Optimization

Keywords

digital circuit, test vectors, test optimization, multifunctional gates, polymorphic gates

1. Introduction

One characteristic of the twentieth century is rapid development in various technologies. One of the biggest booms reached the electronics. The begginig of electronics is often considered to be in the year 1906, when Lee De Forest developed a triode, which was patented in 1907 [3]. Since then a large number of electronics components which complexity is continually increasing has been developed. The most complex integrated circuits today contain billions of transistors (eg NVIDIA Fermi with about three billion transistors [10]). The characteristics of exponentially increasing complexity of electronics was first formulated by Intel founder Gordon E. Moore in 1965 when he said that the complexity of components is approximately doubling every year while maintaining the same price [8]. This statement was later reformulated into a form called a "Moore's Law" [7] as "number of transistors that can be inserted into the integrated circuit, approximately doubles every two years". Since the formulation of Moore's Law has many times been assumed that it's validity will soon be over, mostly because of technological limits. However, these days technological complications were resolved, limits have been broken and is now assumed that the validity of this law will remain until at least 2015 [5]. Many scientists believe that this validity will remain for at least another two decades.

In order to maintain the growth rate of integration (and thus validity of Moore's Law) it is necessary to constantly improve the creation of electronic components in all aspects from design to use. Different levels of abstractions and models in combination with advanced design tools are used in designing circuits and manufacturing is looking for new technologies and procedures.

A large area is diagnostics. The complexity of the current circuits is so large that it is not technologically possible that every manufactured piece is functional. From an economic point of view it is very important to be able to detect defective pieces as soon as possible. The later a defect is discovered, the redress is more expensive. And even if the circuit is found to be fully functional and is deployed in a real application, it can be in the future damaged by a hidden error, external effect or by aging in such a way that it is unable to fulfill its function. Depending on the importance of it's function, early damage detection may be very important. However the increasing complexity of circuits complicates testing and test of complex circuit becomes a complex problem. For the test

*Recommended by thesis supervisor: Assoc. Prof. Zdeněk Kotásek. Defended at Faculty of Information Technology, Brno University of Technology in Brno on June 26, 2012.

Stareček, L. Digital circuits test optimization by multifunctional components. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 4, No. 3 (2012) 54-62

of a circuit it is important that the test is effective enough with reasonable expense. Therefore new techniques and procedures to test circuits are developed and their quality is measured by metrics eg. fault coverage, test price, test application time, the power required for the test etc. The importance of each metric may vary depending on the use of the tested circuit. It turns out that the test application time is becoming a very important parameter for more complex circuits produced in a larger series. The time for circuit tests may take one third of the whole production process. Shortening the test time can speed up production and save expenses.

The main objective was to develop a new methodology for reducing the time of circuit test application. The focus was on simplicity and universality of the methodology and its applicability to complex circuits. Application of new methodology should not have a negative impact on other metrics of the test. A new generic methodology for speeding up the production and reduce test costs are expected.

2. Multifunctional electronics

The term "multifunctional electronics" is used for such electronics, which is able to change its function in a predictable, controllable and desirable way. Since the work deals with the circuits described at gate level, multifunctional logic gates further are discussed.

Multifunctional logic gates can change logic function depending on the control condition. The control condition can generally be anything that is able to change the gate function. Notation of the function of the multifunctional logic gate with n functions is $X_1/X_2/\ldots/X_n$, where each X_i is a standart logic function and is called i-th logic function. Sometimes the i-th logic function is referred as function in i-th mode of multifunctional logic gate.The values âĂŃâĂŃof the control variable can be noted similarly as $Y_1/Y_2/\ldots/Y_n$ where each Y_i is the value of control variable (or control variables). For proper function of the gate must be true that at one time only one Y_i is valid and at that time the gate performs function X_i. An example can be gate AND/OR controlled by a supply voltage 1.5/3.3 V. It is a multifunctional logic gate with two functions. With a power supply voltage of 1.5V the gate performs logical AND and with 3.3V it performs logical OR.

Implementation of multifunctional logic gates may be different. Gates can be implemented like conventional gates or may be implemented using new methods such as polymorphic electronics or electronics based on graphene.

2.1 Conventional gates

Implementation of conventional multifunctional logic gates use standard technologies and procedures for logic gate design. Gate function control is performed by control inputs that have the same behavior and properties as gate function inputs. So they are controlled by standard values of logic levels and cause a similar load to previous elements etc. In terms of function and control inputs gates behave like conventional gates described by a truth table, where one or more inputs are used as control inputs and others as functional inputs. By changing the logic level on the control inputs, the gate changes logic function between the functional inputs and outputs. Because control inputs have no physical difference from functional inputs, we can generally say that any conventional gate with more then one input can be seen as multifunctional gate. In such gate are some inputs considered as control inputs and others as functional inputs. So conventional multifunctional logic gates are obtained only by a way of looking at classical gates and not by special properties of such gates.

The simplest multifunctional gates can be two-input gates. As an example, consider the XOR gate. If we look at this gate as a multifunctional gate where input B is the control input, then in the case of $B = 0$ gate acts as function BUF between input A and the output (transfers same value from input A to output). In the case of $B = 1$ gate acts as function INV (inversion of value from input A). From a multifunctional point of view the XOR gate can act as a controlled inverter, which can be controlled whether output value is inverted or not. Therefore XOR gate can be seen as a multifunctional logic gate BUF/INV.

With the growing number of gate inputs increases the number of options how to use the gate as multifunctional. As an example, consider now a gate described by table 1. If we select inputs A and B as the control inputs and the rest as functional, then we get multifunctional logic gate AND/OR/NAND/NOR. This gate act as function AND in the case $AB = 00$, function OR in the case $AB = 01$, function NAND in the case $AB = 10$ and function NOR in the case $AB = 11$. Another option is to choose only input A as control and the rest of the inputs as functional. Then in the case $A = 0$ the gate serves as the majority and in the case $A = 1$ the gate serves as negation of the majority. Other posibilities control inputs selection can be described similarly.

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Table 1: Truth table of multifunctional logic gate AND/OR/NAND/NOR

2.2 Polymorphic gates

Polymorphic electronics (polytronics) is a type of electronics that is able to change their function according to environmental conditions (eg heat, light, power supply, control input, radiation, etc.) [16]. This principle was introduced in [13], patented in 2000 and studied at NASA Propulsion Laboratory in Pasadena. The basic idea of polytronics is to create electronic circuits that would be able to react to external stimuli by changing their functions. The important feature is that this change is based only on the physical properties of the technology which is used to implement such electronics and not on special sensors.

Most of the published polymorphic gates are shown in Table 2. Implementation of NAND/NOR gate controlled by V_{DD} is shown in Figure 1. Polymorphic gates introduced by Adrian Stoica et al. presented in [15, 16, 17, 14] was obtained by searching for solutions using evolutionary techniques. Gate presented in [11] was invented by direct design by Roman Prokop from UMEL FIT VUT Brno.

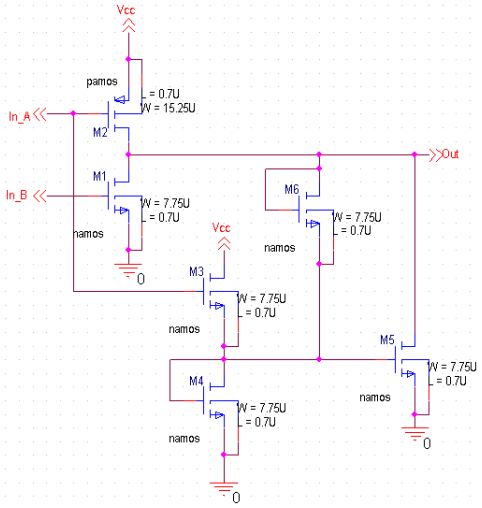


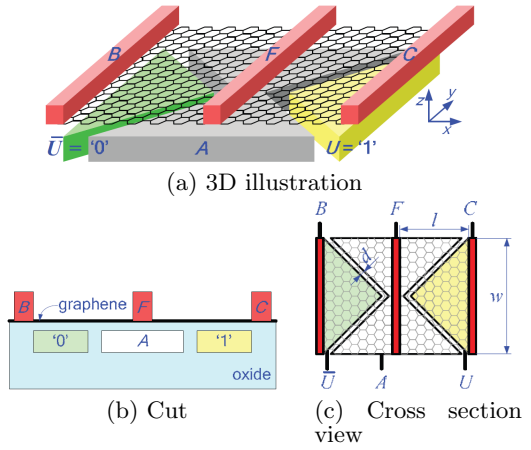
Figure 1: NAND/NOR controlled by power supply voltage (A. Stoica)

2.3 Graphene gates

Graphene is a great hope for electronics and digital circuits. It was shown that graphene can act as a FET transistor and hence can be used as an underlying technology for electronic circuits as well as today's technology, largely based on silicon [9, 6, 1]. In 2006, IBM created a complete integrated logic circuit consisting of transistors implemented on carbon nanotubes [2]. Among scientists there is a presumption that the graphene technology will gradually replace today's silicon [2, 4, 1]. But it is estimated that this will not happen in less than 20 years.

One of the first implementation of multifunctional graphene gate was introduced in [18] and is shown in figure 2. The gate is created on the semiconductor substrate. In this substrate three areas $\overline{U}$, A and U in the triangular shape at an angle of 45° are formed. On the smooth top of the substrate is applied a graphene layer on which electrodes B, F and C are placed. Contacts connected to A, B and C are gate inputs, contact connected to F is the gate output and contacts U and $\overline{U}$ are used for power supply. In basic connection $\overline{U}$ is connected to a low potential (ground) and U to the high potential (power supply). In this mode labeled $U = 1$ gate performs a logic function described by the equation $Y = AC + \overline{A}B$. However, the gate also allows the opposite connection, where $\overline{U}$ is connected to a high potential (power supply) and U to a low potential (ground). In this case, the mode is labeled $U = 0$ and the gate function is described by the equation $Y = \overline{A}C + AB$. Functions for both modes are described in the truth table 3a. If we use one input as a control input, it is possible to implement multifunctional gates listed in table 3b.

One of the advantages of this gate is that the base substrate technology for its implementation is the same as



(a) 3D illustration

(b) Cut

(c) Cross section view

Figure 2: Structure of graphene logic gate [18]

Inputs			Outputs U_1	Outputs U_0
A	B	C	Y	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	1

(a) Truth table

Control	function U_1	function U_0
A	B/C	C/B
B	AND/⟶	⟵/OR
C	⟵/OR	AND/⟶

(b) Achievable multifunctional gates

Table 3: Graphene gate function

for the CMOS technology. The main technological difference is only the graphene layer which is applied between the substrate and gate electrodes. According to [18] similar technologies should ensure integration of conventional CMOS technology with the graphene technology and it would be possible to create new hybrid graphene-CMOS circuits.

3. Thesis objectives

3.1 Motivation

In the last few years discussions started about the technology and principles of creating multifunctional logic gates. Similar technologies also bring the possibility to change the function of internal circuit elements without modifying circuit structure at the gate level. Changing the gate function inside circuits causes a change in transparent paths, the equivalent faults and last but not least changes in faults, which are tested by currently assembled activation and detection path. These features lead to considerations about the possibilities of using multifunctional gates for various diagnostic characteristics modifications of the circuit, which would lead to parameters optimization of the resulting tests. Optimized parameters can be a number of test vectors and fault coverage of the final test.

3.2 Thesis objectives

The main objective was to verify the assumption that the multifunctional elements can affect the diagnostic characteristics of the circuit in such a way that it is possible to achieve the desired changes in the parameters of the final circuit test. Based on this assumption, the work aimes to

Function	Control levels	Control value	Tran.	Source
AND/OR	27/125°C	temperature	6	[16]
AND/OR/XOR	3.3/0.0/1.5V	input voltage	10	[16]
AND/OR	3.3/0.0V	input voltage	6	[16]
NAND/NOR/XOR/AND	0.0/0.9/1.1/1.8V	input voltage	11	[14]
AND/OR	1.2/3.3V	supply voltage	8	[16]
NAND/NOR	3.3/1.8V	supply voltage	6	[15]
NAND/NOR	5/3.3V	supply voltage	8	[11]

Table 2: Published polymorphic gates

design and implement a methodology for optimizing the parameters of the test. Using this methodology then verify the assumption on the task of minimizing the number of test vectors needed to test the circuit while maintaining other test quality parameters. The solution should comply with the following specific objectives:

1. Propose and describe the principles of using multifunctional gates in circuits for optimizing the tests.
2. Formaly describe these principles and using this formalism define methodology for test optimization.
3. Implement the proposed methodology.
4. With the implementation verify the methodology on the task of minimizing the number of test vectors for various circuits, including the ISCAS 85 test kit.
5. Analyze the known technologies for creating multifunctional gates. Find out their properties and assess their suitability for the proposed methodology.

The focus is put on simplicity and universality of the methodology. Furthermore, the methodology should be usable for complex circuits. At the same time proposed methodology could not have a greater negative impact on quality of other test parameters. Created methodology and its implementation for its verification should also be easily embeddable into todays standard manufacturing process of circuits and their tests, and should be easy to use with standard design systems.

4. Multifunctional logic gates

4.1 Conventional gates

Conventional multifunctional logic gates are in fact conventional logic gates described by the standard truth table and implemented using standard technologies. As part of my work I invented a gate based on CMOS technology shown in Figure 3, its function is described in truth table 4. This gate has three inputs A, B and $Vsel$ and an output Out. In fact, it is a conventional three-input gate with a function of inverted majority of the three. When anyone is choosen out of its three inputs as control input, the gate is always multifunctional logic gate NAND/NOR. I consider the NAND/NOR function as one of the most important because the NAND and NOR functions are logically complete and it is possible with their help to create any logical function or complex digital circuit.

Because it is in fact a classical CMOS logic gate, its electrical properties are similar to other CMOS gates. Gate has a good levels of output voltages for both logic levels and greater current consumption (and hence power consumption) only when switching states. Neither the other parameters such as noise immunity, delay or gain in

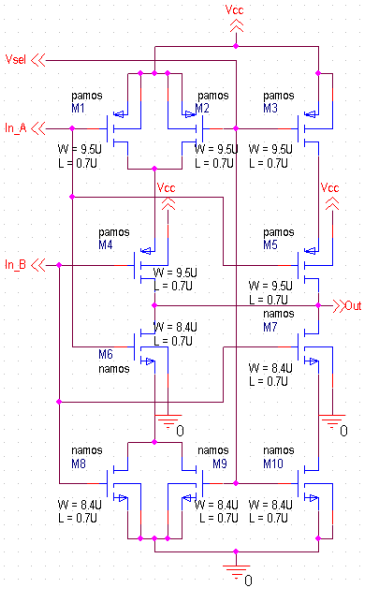


Figure 3: Conventional multifunctional CMOS NAND/NOR gate

Vsel	B	A	Out
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Table 4: Truth table of conventional multifunctional CMOS NAND/NOR gate

simulations were different from other conventional CMOS gates.

4.2 Polymorphic gates

Most of the published polymorphic gates are listed in Table 2. For each gate except the AND/OR gate controlled by temperature model in OrCAD PSpice was created and the simulations of logical and electrical properties were performed. It is important to note that Adrian Stoica et al. invented and simulated gates using HP 0.35 μm technology. I used transistors from AMI 0.7 μm technology.

Simulations have shown a fundamental problem with all gates controlled by the voltage on the control input. None of them were functional. All my attempts to make them working were unsuccessful. Simulations of the remaining polymorphic gates found problematic features that sig-

nificantly reduce the possibility of their extensive use in complex circuits. Generally speaking the gates had a large current consumption from the power supply, some of them excessively overloaded the previous stage, most of them had an incorrect voltage level at the output and with an increasing load at the output some of the gates rapidly decrease maximum operating frequency. In the case of connection of other (even conventional) gates to the gate output, the incorrect output voltage levels leads to another increase of current consumption from the power supply. Polymorphic gates also usually have a small noise immunity and are very sensitive to the exact input voltage levels. This behavior together with the inaccuracy of the output voltage levels often causes not to be possible to connect more similar gates directly to other.

5. Digital circuits test optimization

When creating a test sequence for logic circuit, the test vectors are gradually created to cover failures in the analyzed circuit. The resulting sequence of test vectors then has certain quality parameters described by various properties such as the number of test vectors, fault coverage, etc. Multifunctional gates offer the possibility to modify the function of the internal circuit elements and thus affect behavior of the circuit also in test. The proposed method is based on the assumption that changing some functions of internal elements will alter the diagnostic characteristics of the whole circuit. We can therefore talk about a specific hypothesis, which is subject to verification in my work. In my work is a formal model proposed for this purpose. This formal model defines the concepts used in the methodology, which is named as "digital circuits test optimization". Used procedures and algorithms then operate on this formal model.

5.1 Principle of the method

The method is built on the possibility to change the function of some internal gates in such a way that the required parameters of the test are improved. Because it is necessary for the operation of the circuit to maintain its original function, it is required that modified gates can perform the orginal logic function. So modified gates in *operation* mode must fulfill the function as is required for the function of the circuit and in *test* mode which is more suitable for the test. This feature can be achieved using multifunctional gates discussed in chapter 2 and 4.

On the basis of the problem, test requirements and limiting conditions, this method allows the choice whether it will be possible to switch gate functions during test application. The simplest way is when the gate function does not change during the test. All gate function control inputs are then connected together and accessible by one circuit input. Prior to circuit test, the function of gates switches to test mode, then the test is applicated and after its completion, gates switch back to operation mode. Another option is to allow gate function switch during the test.

The whole method thus consists of three basic sub-tasks. Identification of the gates in the circuit where a function change will improve the test, the selection of their test function and the selection of how to control the test function of these gates.

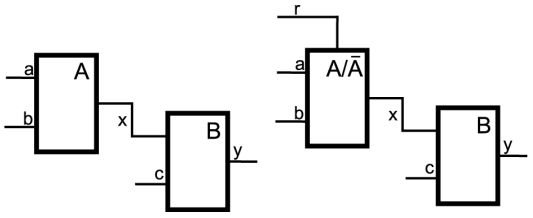
5.2 Optimization of testability

Testability optimization using multifunctional elements is based on the insertion of test points to improve the controllability or observability of the selected point. The basic difference is that the proposed methodology does not add a new gates to the circuit, but only replaces the existing ones by multifunctional gates. Selection gates, its functions in test mode and control principle must be implemented according to the requirements to improve the controllability.

5.2.1 Direct control

Traditional implementation of direct control of a certain point by inserting the test point is achieved by adding a new logic gate before this specified point. Inserted gate function depend on whether it is needed to manage only one particular logical level or it is needed to have complete control over the point. A new gate in the circuit results in a change of dynamic properties around this gate, or even change of the dynamic properties of the whole circuit.

The proposed methodology does not add any new gates to the circuit. The requirement for direct control can be solved by replacing the gate (whose output goes to the specified point) by a multifunctional gate. Consider the situation in figure 4a and the requirement for direct control of the logical level at point x. If we replace gate A by a multifunctional gate (figure 4b), we can directly control the logical value at point x.



(a) Initial circuit (b) Modified circuit

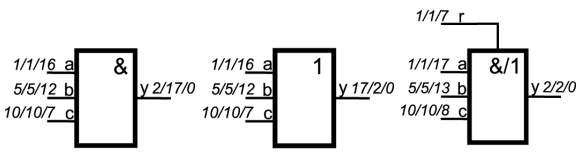
Figure 4: A circuit with the requirement for direct control of the logical level at point x

Test function of multifunctional gate depend on the requirement which logical levels at the point x is needed to control. If the full control of point x is required, then the function in test mode must be a complement of function in operation mode.When the logic level at the point x is different then is requested, the requested logic level can be achieved by the change of the gate function.

5.2.2 Improving testability by SCOAP

Another problem to be solved is a general improvement of circuit testability based on SCOAP method. The solution is based on the reduction of controllability values of selected points in a circuit.

As an example, let us look at a three-input AND gate rated with SCOAP as shown in Figure 5a. If we allow the gate function to change from AND to OR during the test, then it is possible to change the function of the gate to achieve good controllability value for the currently controlled logical level. This principle is shown in Figure 5c.



(a) AND (b) OR (c) AND/OR

Figure 5: Three-input AND, OR and AND/OR gate rated with SCOAP method

5.3 Optimization of the test

The principle of test optimization by multifunctional elements is based on assumptions specified in chapter 3. If these assumptions are valid, then it is possible to achieve the desired test parameter optimization by appropriate modifications of some internal circuit elements. The desired optimization can be various test aspects such as the number of test vectors, required power, fault coverage etc. Although the proposed methodology is generally applicable to more problems, it was used to reduce the number of test vectors.

5.3.1 Principle of the method

As stated in chapter 5.1, it is necessary to solve three main problems. Selection of gates, selection of their test function and selection how to control them. The proposed method selects a method of control when the function of gates can not be changed during the test. So the method consists only of identifying gates, that would be appropriate to change, and selection of their functions.

The input of the method is circuit C described at the gate level, which test should be optimized. The output of the method is structurally same circuit C', which has changed the function of certain gates and has better optimized test parameters. In the final circuit are altered gates represented as multifunctional, when in functional mode have function as gates in original circuit and in test mode have function as the gates in the optimized circuit. To optimize circuit C and obtain the circuit C' with better optimized parameters, the method of combinatorial optimization is used.

5.3.2 Objective function and constraints

For optimization of the number of test vectors were used parameters:

- Number of test vectors.
- Fault coverage.
- Number of multifunctional gates.
- Number of CMOS transistors for implementation.

The solution is based on the search for C' for circuit C as defined in the thesis [12]. Because this definition ensures consistency of the circuit, it is not necessary to address this problem. Any solution found under this definition is a valid circuit.

Objective function was implemented as a comparative function of the two circuits. This function returns, whether the tested circuit is better or not. Comparative function considers the circuit as better, if the number of test vectors was reduced and fault coverage was not reduced. If the number of test vectors was same, then the circuit was better when there was an fault coverage increased. If this parameter was the same, then the circuit was better if it had fewer multifunctional gates. If a number of gates were identical, then the circuit was better when it needs fewer CMOS transistors to implement

5.3.3 Optimization algorithms

The first algorithm successfully applied to the problem was exhaustive search. The number of candidate solutions to the problem can be simply approximated by exponential equation c^g, where c is the average size of the set of replaceable gates and g is the gate count of the circuit. As the complexity of exhaustive search grows in proportion to the complexity of the problem, the complexity of finding a solution by this algorithm is exponential. The algorithm was only applicable to relatively small circuits.

The second algorithm successfully applied to the problem is described by algorighm 1. This algorithm is based on depth-first search and was inspired by a hill climbing and backtracking algorithm. The algorithm starts with the circuit for optimization and looks in his 1-neighborhoods (circuits with only one gate changed) for a better solution. If any better solution is found, then the algorihm runs recursively on this solution. After returning from the recursion, the algorithm continues to the next circuit from the 1-neighborhood. Each recursive run keeps the best solution and by comparing best solutions of each run the overal best solution is found.

```
procedure dhc(circuit C) {
 bestsolution = C;
 Cn = getNextSolution(C);
 while Cn != NULL do {
  if compare(C, Cn) {
   solution = dhc(Cn);
   if compare(bestsolution, solution) {
    bestsolution = solution;
   }
  }
  Cn = getNextSolution(C, Cn);
 }
 return bestsolution;
}
```

Algorithm 1: A recursive algorithm

Since the algorithm searches only 1-neighborhood, it may get stuck in a local extreme. On the other hand, the algorithm is able to quickly converge to better solutions and is thus ideal for confirming assumptions.

6. Results

6.1 Exhaustive search

Exhaustive search was used for finding a solution in six small circuits. Summary results are shown in table 5. Since initial circuits fault coverage is 100 % and optimization was restricted to not reduce it, the results are interesting only in terms of reducing the number of test vectors. For all circuits the number of test vectors were reduced from 9,09 % to 50 %. The circuits "prices" increased from 13 % to 60 %. "Price" was expressed by estimation of the number of CMOS transistors required for circuit implementation. The most significant reduction of test vectors was for the dec3to8 circuit. The reduction was 50 % with an estimated "price" increase by 42,85 %.

Circuit	Gates			Vectors			Fault coverage			CMOS transistors			
	C	C'	%	$vc(C)$	$vc(C')$	%	$fc(C)$	$fc(C')$	%	$tc(C)$	$tc(C')$	P	%
fulladd1	4	1	25,00	6	5	83,33	100,00	100,00	100,00	32	28	38	118,75
fulladd2	6	3	50,00	6	4	66,67	100,00	100,00	100,00	40	50	64	160,00
fulladd3	5	2	40,00	6	4	66,67	100,00	100,00	100,00	42	38	56	133,33
comp3bit	11	2	18,18	11	9	81,81	100,00	100,00	100,00	62	70	74	119,35
enc8to3	13	1	7,69	11	10	90,91	100,00	100,00	100,00	74	72	84	113,51
dec3to8	11	4	36,36	8	4	50,00	100,00	100,00	100,00	56	66	80	142,85

Gates C	The number of gates in circuit
Gates C'	The number of modified gates in circuit
$vc(C)$	The number of test vectors of original circuit
$vc(C')$	The number of test vectors of modified circuit
Vectors %	$vc(C')$ in percent versus $vc(C)$
$fc(C)$	Fault coverage of the original circuit
$fc(C')$	Fault coverage of the modified circuit
Fault coverate %	$fc(C')$ in percent versus $fc(C')$
$tc(C)$	The number of CMOS transistors of original circuit
$tc(C')$	The number of CMOS transistors of modified circuit
P	Estimated number of CMOS transistors of circuit with multifunctional gates
CMOS transistors %	P in percent versus $tc(C)$

Table 5: Exhaustive search method results

6.2 A recursive algorithm

A recursive algorithm was used for optimization of more complex circuits from the ISCAS 85 test set and other similar circuits. Summary results are shown in table 6.

For circuits from the ISCAS 85, the algorithm finished only on circuits c17 and c1355. Algorithm finished on all other circuits except mul8 circuit. For the remaining circuits the algorithm was terminated prematurely, because it did not finish within the desired time. Later it was found that the implementation of optimization tool was not optimal and the performance of that tool can be significantly optimized.

Although the algorithm in some circuits did not finish and therefore probably did not find the best achievable solution, the number of test vectors was reduced for all circuits except circuit c1355. Reduction of the number of test vectors was between 12,67 % and 58,21 % with the estimated "price" increase between 0,73 % and 75 %. For circuits from ISCAS 85, the average reduction of the number of test vectors was 27,98 %. There were 2,79 % of changed gates, fault coverage was increased by 0,51 % and "price" rose by 5,29 %.

The most significant reduction of test vectors was for the c499 circuit. The reduction was 58,21 % from 67 to 28 with estimated "price" increase by 13,86 % from 1764 to 2062. An interesting result was also achieved for example with circuit c6288, which has decreased the number of test vectors by 21,74 % from 46 to 36 at an estimated transitors count increase only by 0,73 % from 10112 to 10186 with only ten changed gates.

Some circuits have improved fault coverage. Imporevemts were between 0,16 % and 1,76 %. An interesting result is for example on circuit comp8, where fault coverage have been increased by 1,76 % from 98,27 % to 100 %.

7. Discussion

7.1 The validity of the test

For the entire methodology it is critical to answer the question, what the test of the circuit (in test mode) realy says about the function of the circuit in operation mode. Can we trust to this test? The answer is not simple and we should consider several factors.

The first is to consider that the methodology is built for structural testing. So for the test is not necessary, that the circuits under test have functions, for which they were designed. Furthermore, the methodology is designed for tests at gate level with *t0*/*t1* fault model. In this type of test it is generally assumed that any fault within the gates appear as a fault of type *t0* or *t1* at one of the pins of the gate. Gate implementation is not tested. The proposed methodology did not change the circuit structure at gate level, so the modified circuit can have the same faults and modified circuit test have the same validity as test of original circuit.

The most problematic is the assumption that any fault inside the gate appear as *t0* or *t1* at least on one pin of the gate. This is not always true and tests based on this assumption may not cover these faults [12]. Now we can ask how this will affect the proposed methodology. If we simplify this, we can say that some faults are not detected even inside of the conventional gates, so the incidence of these faults inside multifunctional gates does not significantly affect the test.

Looking at the problem in depth, an important measure is how much and how likely the internal faults may be undetected. Although I did not research a similar topic, I believe that with the increasing complexity of the gates also increases the number of faults, which can be undetected by tests. So for the proposed methodology it will also be important to have gates with an internal structure as simple as possible.

The last important issue relating to the validity of the test is the possibility of failure of gate function control. In terms of the circuit test, the worst case situation is when the gate is in test mode and can not be switched to operation mode. The test will not find any fault, but the circuit would probably not be functional.

Circuit	Gates			Vectors			Fault coverage			CMOS transistors			
	C	C'	%	$vc(C)$	$vc(C')$	%	$fc(C)$	$fc(C')$	%	$tc(C)$	$tc(C')$	P	%
c17	6	3	50,00	9	5	55,56	100,00	100,00	100,00	24	30	42	175,00
c432	160	34	21,25	102	54	52,94	99,24	99,83	100,59	824	1028	1168	141,75
c499	202	35	17,33	67	28	41,79	98,94	100,00	101,07	1764	1818	2062	116,89
c880a	383	34	8,88	104	63	60,58	100,00	100,00	100,00	1802	1902	2092	116,09
c1355	506	0	0,00	108	108	100,00	99,49	99,49	100,00	2244	2244	2244	100,00
c1908	880	34	3,86	163	112	68,71	99,52	99,79	100,27	3446	3568	3762	109,17
c2670	1269	59	4,65	189	109	57,67	95,74	96,73	101,03	5668	5872	6240	110,09
c3540	1669	64	3,83	252	190	75,40	96,00	97,29	101,34	7504	7702	8034	107,06
c5315	2307	62	2,69	190	124	65,26	98,88	99,04	100,16	11262	11404	11790	104,69
c6288	2416	10	0,41	46	36	78,26	99,56	99,56	100,00	10112	10126	10186	100,73
c7552	3513	36	1,02	371	324	87,33	98,26	99,41	101,17	15400	15392	15608	101,35
add8	32	5	15,63	17	10	58,82	100,00	100,00	100,00	250	248	290	116,00
addsub	43	3	6,98	20	17	85,00	100,00	100,00	100,00	274	272	296	108,03
comp8	39	6	15,38	26	19	73,08	98,27	100,00	101,76	244	250	280	114,75
mul8	356	14	3,93	47	31	65,96	100,00	100,00	100,00	2362	2360	2458	104,06
mux16	34	9	26,47	21	11	52,38	100,00	100,00	100,00	292	320	370	126,71
mux8	17	2	11,76	15	13	86,67	100,00	100,00	100,00	146	150	162	110,96
shifter	24	2	8,33	58	49	84,48	88,33	88,33	100,00	96	104	108	112,50
sub8	35	2	5,71	13	9	69,23	100,00	100,00	100,00	256	246	266	103,91

Table 6: A recursive algorithm results (legend is the same as in table 5)

There are several options to solve this problem. The first and simplest can be based on a base structural test assumption, that all faults will occur as *t0* or *t1* on any input of the gate. We can slightly modify this assumption that all faults will occur as *t0* or *t1* on any *functional* input of the gate. Another option is the solution in the form of simple supplementary test in the functional mode. This test should aim to test the potential gate failure occurring only in functional mode and whether gates are switched correctly. The last option I will mention is the possibility of switching the gates during the test. This approach could solve all mentioned problems.

7.2 The use of multifunctional gates

An important part of test optimization is also a selection of multifunctional gates that will be used. One possibility is the use of polymorphic gates. An interesting alternative would be the polymorphic gates, where the function is dependent on the power supply voltage. At some specific supply voltage this gates would be in test mode. In this case a special pin and distribution inside the circuit would not be needed. Unfortunately from the simulations it is evident that the polymorphic gates suffer from various problems and, unlike classical CMOS gates can not be understood as general building blocks of more complex digital circuits. In this current state polymorphic gates are not good candidates for similar applications. Improvements could be achieved by new polymorphic gates, which would not suffer from similar problems.

Graphene multifunctional gates seem to be very promising for the proposed method. For example, the multifunctional gate presented in 2.3 have only three semiconductor areas in the semiconductor substrate, one graphene layer and three electrodes. Gate with similar function in conventional CMOS technology have four times the number of semiconductor areas and twice the number of electrodes [12]. It can be assumed that graphene multifunctional gates will have less potential defects and therefore less of possible faults. Due to the factors mentioned in chapter 7.1 a similar gate seems to be suitable for the presented methodology.

On the other hand, it should be noted that the graphene technology is at the beginning and further research on its properties and usability is necessary. In 2007 it was not expected that over the next twenty years this technology would appear in complex digital circuits [4]. Usability of graphene gates in presented methodology can not be determined until the technology is examined better.

As the previous technologies are not easily applicable, the only technology of conventional gates remain. Conventional gates are designed the same as other parts of the circuit and therefore they does not negatively affect the behavior of the circuit. Since the complexity of this gates can be in some cases significant, it is necessary to have caution with this complexity and influence on the properties of the test.

8. Conclusion

My thesis and this article show that it is possible with multifunctional gates to optimize the diagnostic characteristics of circuits in a desired way. As a first was shown that it is possible with the proposed methodology to achieve similar features as with DfT ad-hoc method of inserting of test control points. Unlike conventional DfT method, the proposed one does not necessarily change the dynamic behavior of the circuit. As a second simple principle based on SCOAP was shown. This principle can reduce the value of controllability and partially observability, so in overall it can improve the testability of the circuit.

However the main part of the work was the creation, implementation and verification of the methodology for the test optimization. On the basis of this methodology was described and implemented new software tool that was designed to reduce the number of test vectors while maintaining fault coverage and keeping circuit complexity as little as possible. This tool was then tested on various circuits, including the ISCAS 85 test set. For all tested circuits except c1355 from ISCAS 85 there was a noticeable reduction in the number of test vectors (usually in the tens of percent) while fault coverage was maintained

or slightly increased. So the tool was able to find high quality solutions and is able to significatnly reduce the number of test vectors.

At the end of this article the problem of validity of modified circuit test in the scope of presented technologies of multifunctional gates was discussed. At first sight graphene technology seems to be very suitable for the method. However this technology is in the research stage and is not applicable today. Its suitability to proposed methodology can be realy confirmed after some reseearch and practical experience. Currently known polymorphic gates are not very suitable with the proposed methodology because of their non-optimal electronic properties. From this section we can then make a simplified conclusion that today the only usable tehchnology for proposed methodology is technology of conventional gates. However, care should be taken to the complexity of such gates.

Acknowledgements. This work was partially supported by the Grant Agency of the Czech Republic under contract No. 102/06/0599 *Methods of polymorphic digital circuit design* and the Research Plan No. MSM 0021630528 *Security-Oriented Research in Information Technology.*

References

[1] K. Bullis. Graphene Transistors [online]. Jan. 2008 [cit. 2011-09-03].

[2] Z. Chen, J. Appenzeller, Y.-M. Lin, J. Sippel-Oakley, A. G. Rinzler, J. Tang, S. J. Wind, P. M. Solomon, and P. Avouris. An integrated logic circuit assembled on a single carbon nanotube. *Science*, 311(5768):1735, 2006.

[3] L. D. Forest. Space telegraphy. u.s. patent 879,532, 1908-02-18 (filled 1907-01-29).

[4] A. K. Geim and K. S. Novoselov. The rise of graphene. *Nature Materials*, 6:183–191, Mar. 2007.

[5] M. Kanellos. New life for moore's law [online], 2005 [cit. 2011-09-04].

[6] M. C. Lemme, T. J. Echtermeyer, M. Baus, and H. Kurz. A Graphene Field-Effect Device. *IEEE Electron Device Letters*, 28:282–284, Apr. 2007.

[7] E. Mollick. Establishing moore's law. *IEEE Annals of the History of Computing*, 28:62–75, 2006.

[8] G. E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, 1965.

[9] K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov. Electric Field Effect in Atomically Thin Carbon Films. *Science*, 306:666–669, Oct. 2004.

[10] Nvidia's next generation cuda compute architecture: Fermi [online], 2009 [cit. 2010-01-21].

[11] R. Ruzicka, L. Sekanina, and R. Prokop. Physical demonstration of polymorphic self-checking circuits. In *Proc. of the 14th IEEE Int. On-Line Testing Symposium*, pages 31–36. IEEE Computer Society, 2008.

[12] L. Stareček. *Digital circuits test optimization by multifunctional components*. PhD thesis, Brno University of Technology, Brno, 2012.

[13] A. Stoica. Polymorphic electronics - a novel type of circuits with multiple functionality. New Technology Report NPO-21213, NASA, 2000.

[14] A. Stoica and R. Zebulum. Four-function logic gate controlled by analog voltage. New Technology Report NPO-40772, NASA, 2006.

[15] A. Stoica, R. Zebulum, X. Guo, D. Keymeulen, M. Ferguson, and V. Duong. Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration. In *IEE Proc.-Comp. Digit. Tech.*, volume 151(4), pages 295–300, 2004.

[16] A. Stoica, R. Zebulum, and D. Keymeulen. Polymorphic electronics. In *In Proc. of International Conference on Evolvable Systems: From Biology to Hardware, LNCS 2210*, pages 291–302. Springer, 2001.

[17] A. Stoica, R. Zebulum, D. Keymeulen, and J. Lohn. On polymorphic circuits and their design using evolutionary algorithms. In *Proc. of IASTED International Conference on Applied Informatics AI2002, Insbruck*, 2002.

[18] S. Tanachutiwat, J. U. Lee, W. Wang, and C. Y. Sung. Reconfigurable multi-function logic based on graphene p-n junctions. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 883–888, New York, NY, USA, 2010. ACM.

Selected Papers by the Author

L. Stareček. Polymorphic gates fault modeling. In *Proceedings of Czech-Slovak seminar Computer architectures and diagnostics for students of doctoral studies*, Bratislava, SK, SAV, 2006, pages 67–72, ISBN 80-969202-2-7.

L. Sekanina, L. Stareček, Z. Kotásek. Novel Logic Circuits Controlled by Vdd. In *Proc. of 2006 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, Prague, CZ, IEEE CS, 2006, pages 85–86, ISBN 1424401844.

L. Sekanina, L. Stareček, Z. Gajda, Z. Kotásek. Evolution of Multifunctional Combinational Modules Controlled by the Power Supply Voltage. In *Proc. of the 1st NASA/ESA Conference on Adaptive Hardware and Systems*, Piscataway, US, IEEE CS, 2006, pages 186–193, ISBN 0-7695-2614-4.

L. Stareček. Polymorphic gates for circuit test optimization. In *Proceedings of Czech-Slovak seminar Computer architectures and diagnostics for students of doctoral studies*, Pilsen, CZ, ZU in Pilsen, 2007, pages 41–46, ISBN 978-80-7043-605-9.

L. Stareček, L. Sekanina, Z. Gajda, Z. Kotásek, R. Prokop, V. Musil. On Properties and Utilization of Some Polymorphic Gates. In *6th Electronic Circuits and Systems Conference (ECS 2007)*, Bratislava, SK, FIIT STU, 2007, pages 77–81, ISBN 978-80-227-2697-9.

R. Růžička, L. Stareček. Development of Building Blocks for Polymorphic Digital Circuits. In *Proceedings of the Work in Progress Session of 10th Euromicro DSD 2007*, Linz, AT, JKUL, 2007, pages 33–34, ISBN 978-3-902457-16-5.

L. Stareček, L. Sekanina, Z. Kotásek. Reduction of Test Vectors Volume by Means of Gate-Level Reconfiguration. In *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, Bratislava, SK, IEEE CS, 2008, pages 255–258, ISBN 978-1-4244-2276-0.

L. Sekanina, L. Stareček, Z. Kotásek, Z. Gajda. Polymorphic Gates in Design and Test of Digital Circuits. In *International Journal of Unconventional Computing*, vol. 4, num. 2, 2008, Philadelphia, US, pages 125–142, ISSN 1548-7199.

J. Šimáček, L. Sekanina, L. Stareček. Evolutionary Design of Reconfiguration Strategies to Reduce the Test Application Time. In *Evolvable Systems: From Biology to Hardware*, Berlin, DE, Springer, 2010, pages 214–225, ISBN 978-3-642-15322-8.