Estonian Journal of Engineering, 2011, **17**, 3, 185–200 doi: 10.3176/eng.2011.3.02

Hierarchical physical defect reasoning in digital circuits

Sergei Kostin, Raimund Ubar, Jaan Raik and Marina Brik

Department of Computer Engineering, Tallinn University of Technology, Ehitajate tee 5, 19086 Tallinn, Estonia; {skostin, raiub, jaan, brik}@pld.ttu.ee

Received 7 February 2011, in revised form 14 June 2011

Abstract. We propose a hierarchical physical defect-oriented approach for fault diagnosis in combinational digital circuits. We present the circuit as a network of modules. As modules we consider either library components (e.g. complex gates) of digital circuits or arbitrary subcircuits. The higher level fault diagnosis is carried out in two phases. In the first phase, faulty modules are located by cause–effect analysis using high-level faulty module dictionary. The size of the dictionary depends linearly on the number of modules in the circuit. In the second phase, the set of suspected faulty modules is pruned by reasoning of the defective behaviour. At the lower level, the physical defects are directly located in suspected faulty modules using defect libraries of the modules or by effect–cause reasoning inside the module. The proposed approach to fault diagnosis helps to cope with the growing complexities of digital circuits. The experimental results show high module-level diagnostic resolution of the proposed approach.

Key words: digital circuits, physical defects, fault models, fault diagnosis, diagnostic resolution.

1. INTRODUCTION

Rapid advances in the areas of nanoscale electron technology and design automation tools enable engineers to design larger and more complex integrated circuits. On the other hand, the increasing integration densities pose severe problems with respect to the quality assurance. The quality and reliability of microelectronic circuits depend essentially on the efficiency of debug and diagnosis of failures in circuits. Traditional approaches to diagnosis of digital circuits are based on gate-level models [1–4]. However, due to the continuous increasing of the gate count in circuits under diagnosis, the gate-level methods are becoming less efficient and obsolete.

Two main trends can be observed when searching solutions for the problems of testing and diagnosis: defect-orientation, and high-level modelling [5]. The trend towards high-level modelling helps us to cope with the complexity, but

moves us even more away from the real life of physical defects and, hence, from accuracy of diagnosis. To handle adequately defects in nanoscale technologies, new fault models and defect-oriented diagnosis methods should be used. But the defect-orientation is increasing again the complexity. To get out from the deadlock, these two opposite trends – high-level modelling and defect-orientation – should be combined into hierarchical approaches.

The classical fault diagnosis algorithms follow two different paradigms: cause–effect and effect–cause analysis [[6]]. The cause–effect approach begins with mapping the causes of failure to a specific fault type e.g. stuck-at fault (SAF) model. Then, the fault tables or fault dictionaries are built by fault simulation. Once the fault dictionary is available, the effect (or syndrome) of the failing chip is analysed using dictionary look-up. This approach is also called fault-dictionary based paradigm or combinational diagnosis approach [[7]]. However, as designs grow in complexity, dictionary-based fault diagnosis approaches are becoming infeasible due to prohibitively large dictionary sizes. Effect–cause approach looks at the failing outputs and starts reasoning on the logic or topological structure of the circuit to be diagnosed. Because of the sequential character of reasoning, this approach is called also sequential fault diagnosis.

Traditional approaches to the cause–effect fault diagnosis lay on the stuck-at fault model. Many researchers have focused on developing new fault models for particular types of failure mechanisms like signal line bridges [[6,8]], transistor stuck-opens [[9,10]] or failures due to changes in circuit delays [[11]]. Another trend has been to develop general fault modelling mechanisms and corresponding test tools that can effectively analyse arbitrary fault types like in [[12]] where D-cubes are used to model any arbitrary change in the logic function of a circuit block. A generalization of this approach has been found in the input pattern fault model [[13]] and in the pattern fault model [[14]], which can model any arbitrary change in the logic function of a circuit block, where a block is defined to be any combinational subcircuit, described at any level of the design hierarchy.

A similar pattern-related fault modelling approach, called *functional fault model*, was proposed earlier in [[15]] for the module-level fault diagnosis in combinational circuits based on solving systems of Boolean differential equations. The functional or pattern fault model allows an arbitrary set of signal lines to be grouped into activation conditions for a single fault site, allowing a variety of fault types to be modelled. The functional faults can be either static or dynamic [[16]].

In [[17]], a similar model, called *conditional faults*, was proposed for test generation purposes, and in [[18]] for diagnosis purposes. A conditional fault allows additional signal line objectives to be combined with the detection requirements of a particular fault or physical defect.

The main objective of this work is to carry out the fault diagnosis at two levels (Fig. 1). To reduce the complexity of the diagnosis problem we represent the circuit as a network of modules whereas the modules can be arbitrary gate subnetworks. The task of diagnosis is considered as a task to locate faulty modules in the network of modules by using module-level fault dictionaries. The

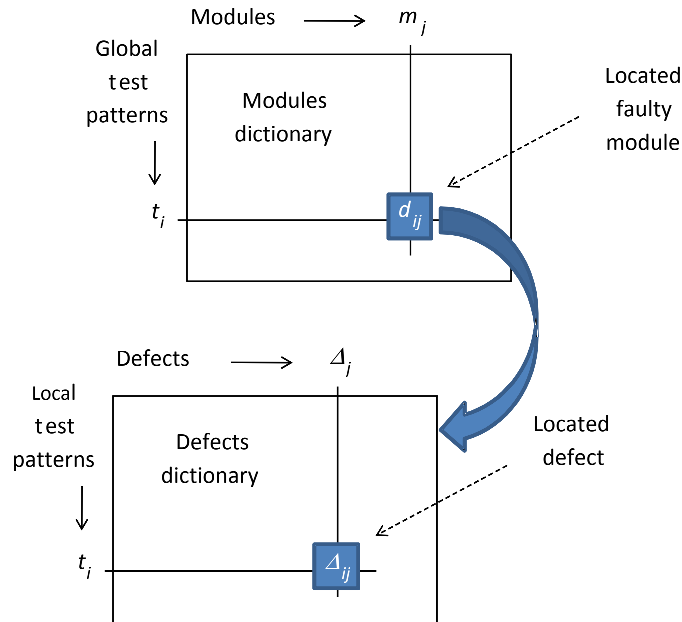

Fig. 1. Illustration of the hierarchical approach to fault diagnosis.

module size can be decided as the trade-off between the size of the faulty subcircuit to be located at the higher level and the size of the fault dictionary. Arbitrary and multiple faults in the module are allowed. However, it is assumed that no more than a single module is faulty at a time.

As the result of the fault diagnosis at the higher level, a faulty module or a subset of suspected faulty modules in the tested circuit is located. In the next step, this subset will be suppressed by a special defect reasoning based on using the functional fault (conditional SAF) model.

At the lower level, the defect location in the suspected faulty module will be determined. For this purpose we need the defects dictionaries for suspected modules. In the case when the modules represent library components used during design, we can precalculate the dictionaries for each library component beforehand and store the dictionaries in the library as well. Otherwise, the defects libraries should be created by simulation ad hoc.

The main objective of this work is to apply the cause–effect fault diagnosis hierarchically at two levels; first, in the module network to locate the faulty module, and thereafter, to locate a defect inside the faulty module by defect reasoning.

At the high level, we use the fault dictionary where instead of enumerating all the possible fault cases (or defects), we list only the modules as potential faulty items in the circuit. This helps to cope with the complexity of the fault dictionaries. At the lower level, two approaches may be used. In case when the

modules represent library complex gates, we use the low-level defect dictionaries to diagnose the suspected defect(s). In case when the modules represent logic subcircuits not described in the component libraries, either low-level effect–cause reasoning can be carried out to locate the defects, or the defects libraries should be created for the low-level diagnosis purpose.

The rest of the paper is organized as follows. In Section 2 we describe the general flow of fault diagnosis. Section 3 presents the method of high-level fault diagnosis based on modules dictionaries. In Section 4, we describe the functional fault model. A defect-reasoning method for pruning the set of suspected faulty modules is explained in Section 5. Section 6 presents the ideas of the low-level defect diagnosis, and in Section 7, we describe the experimental results. Section 8 concludes the paper.

2. GENERAL DESCRIPTION OF THE METHOD

In this paper, we propose a hierarchical approach for the fault diagnosis in combinational circuits. The high-level model of a circuit is presented as a network of modules. The modules may be library components (complex gates) and/or arbitrary single-output logic subcircuits. A high-level fault dictionary is generated for the given test where for each test pattern it will be shown in which modules the defects may be detected by this pattern. The task of the high-level diagnosis is to determine the faulty module or a subset of candidate faulty modules.

In the first step, by the procedure of the cause–effect diagnosis, based on using the high-level fault dictionary, a subset of candidate modules, suspected as faulty, is determined. On the next step, by using effect–cause high-level module reasoning, the initial subset of candidate faulty modules will be suppressed (Fig. 2). The high-level module reasoning is defect oriented, however the direct defect analysis is avoided. The defects are modelled indirectly by using the functional fault model, which is represented by pairs of SAF and defect conditions [15,18]. In the process of reasoning, only the defect conditions are considered, which allows to avoid dealing directly with the whole huge list of defects in the circuit.

Low-level fault diagnosis is carried out in the modules determined during the high-level reasoning as faulty. For low-level diagnosis purposes, the modules are represented by sets of local test patterns, which are treated as conditions for activating physical defects in modules. This diagnostic information is captured in the form of defect dictionaries and is stored in the component libraries. Using pre-generated defect dictionaries, low-level cause–effect reasoning is carried out to locate the defects in the modules. If the defect location with the help of defect dictionaries is not possible, or if no dictionaries for given modules are available, effect–cause low-level defect reasoning should be carried out to locate the defects in suspected modules.

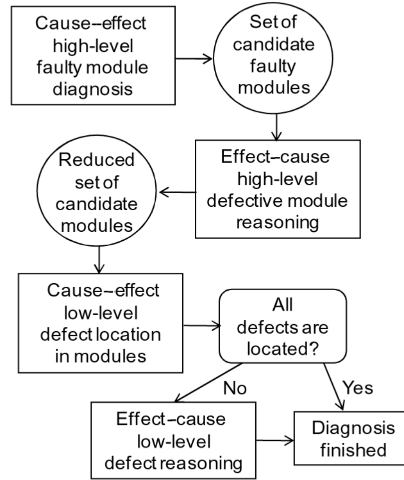

Fig. 2. The hierarchical fault diagnosis flow.

3. CAUSE–EFFECT HIGH-LEVEL FAULT DIAGNOSIS

Consider a combinational circuit C, synthesized as a network of modules with a single output (library complex gates or arbitrary sub-circuits) with a set of primary outputs *OUT*, a set of primary inputs IN, and a set of internal nodes *INT*. The network C consists of a set of modules M where each module $m_i \in M$ has an output node $y_i \in OUT \cup INT$ and a set of input nodes $X_i \in IN \cup INT$.

The circuit represents a graph $C = (M, \Gamma)$, where the nodes correspond to the modules $m \in M$, and Γ is a relation on M, where $\Gamma(m) \subset M$ denotes the successor nodes of *m*, and $\Gamma^{-1}(m) \subset M$ is the set of the predecessor nodes of *m*. For all the output nodes $m_j \in M$, $j: y_j \in OUT$, we have $\Gamma(m_j) = \varnothing$. For all the input nodes $m_j \in M$, which represent the modules with only primary inputs $j: X_j \subset IN$, we have $\Gamma^{-1}(m_j) = \varnothing$.

Example 1. An example of a combinational network C is presented in Fig. 3, where to each network component (functional block F) a module corresponds. The graph $C = (M, \Gamma)$ of the network is depicted in Fig. 4, where are shown modules that correspond to respective functional blocks (e.g. $F_1 \to 1$) and the connections between the modules (e.g. F_1 is connected to F_4 and F_7). The task of the high-level fault diagnosis is to locate the faulty module if the testing of the circuit has been failed.

In general case, there will be no restrictions to the fault types inside the modules (some restrictions considered later may be introduced to simplify the fault diagnosis at the cost of worse diagnostic resolution).

The first phase of the high-level fault diagnosis is carried out in the network of modules according to the cause–effect approach by using its pre-generated fault dictionary. The module-level fault dictionary is represented as the matrix $D = \| d_{ij} \|$, where i denotes th number of test pattern, and j denotes the module

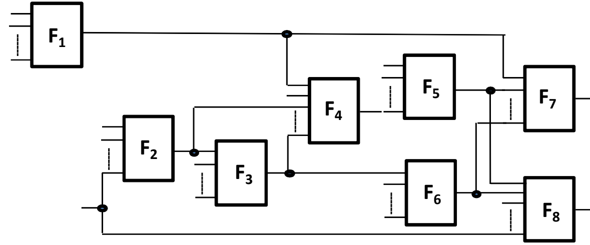

Fig. 3. A combinational circuit.

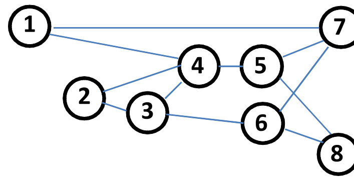

Fig. 4. Graph representing the circuit in Fig. 3.

$m_j \in M$ (a subcircuit or a library component in the network). We say $d_{ij} = 1$ if the test pattern t_i may detect a faulty signal on the output y_j of the module m_j.

Let $M_i = \{m_j\} \subset M$ be the subset of modules corresponding to the i-th row of the matrix D, so that $m_j \in M_i$ if $d_{ij} = 1$.

Let us have a test set T to be used for fault diagnosis. Partition the set T in accordance with the results of test experiment into two subsets: $T = T_F \cup T_P$, where $T_F \cap T_P = \varnothing$, T_F is the subset of test patterns, which failed during the test, and T_P is the subset of test patterns, which passed. Denote by the vector

$$\boldsymbol{M_F} = \bigcup_{i:t_i\ T_F} \boldsymbol{M_i}$$

the subset of all modules suspected as faulty on the basis of failing subset of test patterns T_F, and by the vector

$$\boldsymbol{M_P} = \bigcup_{i:t_i \in T_P} \boldsymbol{M_i}$$

the subset of tested modules, which have shown correct behaviour on the basis of the subset of passed test patterns T_P.

Consider now the subset of modules $M^* = M_F - M_P$. It is easy to understand that if $M^* \neq \varnothing$ then the modules in M^* should be suspected as faulty to explain the failing of test patterns T_F.

Consider another subset of modules determined as $M^1_{cond} = M_F \cap M_P$. All these modules in M^1_{cond} should be treated as ambiguous, because they may explain the failing of test patterns T_F, however, not necessarily. Let us call this

Table 1. Module level fault dictionary

Test	Modules							
	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
t_1	1		1			1	1	1
t_2	1						1	1
t_3			1			1	1	1
t_4		1	1			1	1	1
t_5				1	1		1	1

subset as conditionally suspected modules after the first phase of high-level fault diagnosis. The total subset of modules suspected as faulty after the first phase of high-level diagnosis is determined as $M^1 = M^* \cup M^1_{cond}$.

Example 2. Table 1 represents an example of the module-level fault dictionary D.

Assume the test patterns t_1 and t_2 failed during the test, e.g., $T_F = \{t_1, t_2\}$, and other three test patterns $T_P = \{t_3, t_4, t_5\}$ passed. Hence, $M_F = \{m_1, m_3, m_6, m_7, m_8\}$, and $M_P = \{m_2, m_3, m_4, m_5, m_6, m_7, m_8\}$. On this basis, as the result of the first phase of high-level fault diagnosis, we get: $M^* = \{m_1\}$ and $M^1_{cond} = \{m_3, m_6, m_7, m_8\}$. The modules m_3, m_6, m_7, m_8 remain conditionally suspected as faulty.

In this phase of high-level diagnosis we cannot use the information from the passed patterns $T_P = \{t_3, t_4, t_5\}$ to reduce the subset of suspected faulty modules in M^1_{cond}, because we do not know which defects have been tested in modules M^1_{cond} during the test sets T_F and T_P.

4. DEFECT REASONING IN MODULES BY USING THE CONDITIONAL SAF MODEL

In this Section we describe how the functional fault model, consisting of pairs of SAF and defect conditions, can be used for high-level defect reasoning in suspected faulty modules.

Consider a module $m \in M$, which is represented by a Boolean function $y = f(X)$, $X = (x_1, x_2, \ldots, x_n)$. Introduce a symbolic Boolean variable Δ for representing a given defect in the module, which converts the fault free function f into another faulty function f^{Δ}. Construct for this defect a generic parametric function

$$y^* = f^*(x_1, x_2, ..., x_n, \Delta) = \overline{\Delta} f \vee \Delta f^{\Delta}$$

to model the defect of the module m as a function of the defect variable Δ, which describes jointly the behaviour of the module for both, fault-free and faulty cases. For the faulty case, $\Delta = 1$, and for the fault-free case, $\Delta = 0$, i.e. $y^* = f^{\Delta}$ if $\Delta = 1$, and $y^* = f$ if $\Delta = 0$. The solutions $W_y(\Delta)$ of the Boolean differential equation

$$\frac{\partial f^*}{\partial \Delta} = 1 \tag{1}$$

describe the set of conditions (input signals of the module) which activate the defect Δ to produce an error on the output line y of the module. To find the conditions $W_y(\Delta)$ for a given defect Δ, we have to create the corresponding logic expression for the faulty function f^{Δ}, either by logical reasoning or by carrying out defect simulation directly, or by carrying out real experiments to learn the physical behaviour of different defects. The described method represents a general approach to map an arbitrary transistor level physical defect inside the module m to the higher logic (or module) level.

Example 3. As an example, assume there is a short inside the transistor circuit in Fig. 5, described by the function

$$y = f(X) = x_1 x_2 x_3 \vee x_4 x_5.$$

The short changes the function of the circuit as follows:

$$y = f^{\Delta}(X) = (x_1 \vee x_4)(x_2 x_3 \vee x_5).$$

Using the defect variable Δ for the short, we create a generic Boolean differential equation and solve it as follows:

$$\frac{\partial y^*}{\partial \Delta} = \frac{\partial((x_1 x_2 x_3 \vee x_4 x_5)\overline{\Delta} \vee (x_1 \vee x_4)(x_2 x_3 \vee x_5)\Delta)}{\partial \Delta}$$
$$= x_1 \overline{x_2 x_4} x_5 \vee x_1 \overline{x_3 x_4} x_5 \vee \overline{x_1} x_2 x_3 x_4 \overline{x_5} = 1.$$

From this equation three possible solutions follow: 10x01, 1x001, 01110, where x is don’t care value. Each of them can be used as a test pattern for the given short, or used as the defect condition for SAF at the output y. According to the definition of the functional fault model for the defect Δ, we have $W_y(\Delta) = \{10\text{x}01, 1\text{x}001, 01110\}$.

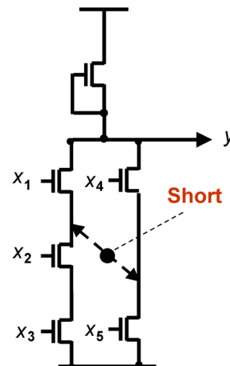

Fig. 5. Transistor circuit with a short.

We call the full set of conditions $W_y = \{W_y(\Delta)\}$ as the *functional fault model* to represent all the physical defects through functional deviations in the behaviour of the module m: a physical level defect Δ produces a higher logic level erroneous signal on the module output y if $W_y(\Delta) = 1$. By using the set of conditions W_y, it is possible to map the defects from lower physical level to higher logic level for fault simulation purposes or vice versa, to map the faulty logic signals from the module level to physical defects for fault diagnosis purposes. If the modules of the circuit represent standard library components (e.g. complex gates) the described analysis for finding conditions should be made once for all library components, and the sets of calculated conditions W_y will be included in the form of defect dictionaries into the library of components.

5. EFFECT–CAUSE HIGH-LEVEL FAULT REASONING

We define the effect–cause high-level fault reasoning as the second phase of high-level fault diagnosis as follows. We take the set of all conditional candidate faulty modules $M^1_{cond} = M_F \cap M_P$, found in the first high-level diagnosis phase, and create the sets of fault conditions $X_{F,i}$ for all modules $m_i \in M^1_{cond}$, suspected conditionally. The set M^1_{cond} can be interpreted as the effects evoked locally by the possible defects in the modules $m \in M_F \cap M_P$. In these suspected modules we have to carry out the indirect defect reasoning either to determine the possible defect causes or to remove the not faulty modules from suspicion. The task of the second phase of high-level fault diagnosis is to locate the faulty module or to reduce the set of suspected faulty modules as much as possible.

Denote by X_{ij} the local input condition of the module m_i at the test pattern t_j. Let us group all these input conditions for all the modules in such a way that $X_{F,i}$ contains all the local input conditions of the module m_i during the test patterns, which tested the module and failed, and $X_{P,i}$ contains all the local input conditions of the module m_i during the test patterns, which tested the module and passed.

Each condition $w_F \in X_{F,i}$, so that $w_F \in W_i(\Delta)$, refers to a defect Δ, which may have been detected by the test patterns in T_F, and therefore should be suspected because all the test patterns in T_F failed. On the other hand, each input condition $w_P \in X_{P,i}$, so that $w_P \in W_i(\Delta)$, refers to a defect Δ, which should have been detected by the test patterns in T_P. However, since all the test patterns in T_P passed, the defect Δ cannot be any more suspected.

From the above reasoning it results that only these defects in the module m_i, which are activated by the conditions $w \in (X_{F,i} - X_{P,i}) \cap W_i^f$, may be suspected. Hence, a module $m_i \in M^1_{cond}$ can be suspected as faulty only if

$$(X_{F,i} - X_{P,i}) \cap W_i^f \neq \varnothing. \quad (2)$$

The reasoning proceeds in the following way. We delete from all conditions $X_{F,i}$, where $m_i \in M^1_{cond}$, the conditions $X_{P,i}$, trying to suppress the set M^1_{cond}.

The result of the second phase of high-level fault diagnosis will be the set of conditionally suspected faulty modules $M^2 = M^* \cup M^2_{cond}$, where $M^2_{cond} \subseteq M^1_{cond}$, and for all $m_i \in M^2_{cond}$ the condition (2) is valid.

Example 4. Consider again the example in Table 1, where the first two test patterns failed during the test experiment, and the other three passed. Let the module m_3 had the local input vector $X_{13} = (1011)$ during the test pattern t_1, which tests the module according to Table 1. We assume here that the module has 4 input lines. Since t_1 failed, it means that in the module m_3, a defect Δ, activated by the conditions $w_F = (1011) \in W_3(\Delta)$ on the inputs of the module, should be suspected. On the other hand, let the same module had on its inputs the same condition $w_P = w_F = (1011)$ during the test pattern t_4, which tested the module m_3 as well, but passed. Since the test pattern t_4 activated the same defect Δ, which was under suspicion after the test pattern t_1, there is a contradiction, and the defect Δ cannot be present. So, we can remove the module m_3 from the set of suspected modules $M^1_{cond} = \{m_3, m_6, m_7, m_8\}$. After such a procedure of the indirect effect-cause reasoning of defects on the module level, we may get in ideal case $M^2_{cond} = \varnothing$. In this case, the result of the high-level fault diagnosis show that we should suspect a single module, $M^2 = M^* = \{m_1\}$, as faulty.

6. LOW-LEVEL FAULT DIAGNOSIS

Low-level fault diagnosis is carried out in the modules determined during the high-level reasoning as faulty. It may be carried out either by cause–effect diagnosis, based on the defect dictionaries pre-computed for the library components, or by effect–cause diagnosis inside the modules if the defect libraries are not available.

Consider as an example the fault dictionary in Table 2 for a library complex gate AND2,2/NOR2 [[19]] regarded as a module in the circuit under diagnosis. In the table, 25 different defects (shorts between inputs or internal nodes of the module) are considered with corresponding names in column 2 and erroneous functions in column 3. Other columns form the defect dictionary and correspond to the input patterns of the 4-input module (given with decimal numbers). The entries "1" in columns show which defects are tested by which patterns.

Assume that as the result of the test experiment, we have found for this particular module m_i that $X_{F,i} = \{w_3, w_{12}\}$, and $X_{P,i} = \{w_0, w_5, w_8, w_{10}, w_{13}, w_{15}\}$. The input patterns not exercised during the test are highlighted by grey colour. It is easy to see that the defects Δ_2 and Δ_8 should be suspected as present. For better diagnostic resolution, to distinguish these defects, additional input patterns should be used for testing, e.g. w_7 or w_{11}.

If no diagnosis is possible with using the defect dictionary, for example, when the real existing defect is not covered by the dictionary, or in the case when the defect dictionary for a module is missing, the low-level effect–cause defect reasoning inside the module is needed. This can be proceeded, for example, by the method of critical path tracing.

Table 2. Defect library for a gate AND2,2/NOR2 [19]

k	Fault Δ_k	Erroneous function $f^{\Delta k}$	Local input test patterns w_j															
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A/C	not((A*C)*(B+D))				1				1					1	1		
2	A/D	not((A*D)*(B+C))				1				1					1		1	
3	A/N1	not(B*(not(A)+C+D)+C*D)					1	1	1						1			
4	A/Q	A*(not(C*D))	1	1	1		1	1	1						1	1	1	
5	A/gnd	not(C*D)													1	1	1	
6	A/vdd	not(B+(C*D))					1	1	1									
7	B/C	not((B*C)*(A+D))				1								1	1	1		
8	B/D	not((B*D)*(A+C))				1								1	1		1	
9	B/N1	not(A+C*D)									1	1	1					
10	B/Q	B*(not(C*D))	1	1	1						1	1	1		1	1	1	
11	B/gnd	not(C*D)													1	1	1	
12	B/vdd	not(A+(C*D))									1	1	1					
13	C/N1	not((A+B+D)*(A*B+not(C)+D))		1			1	1			1	1						
14	C/Q	C*(not(A*B))	1	1		1	1	1		1	1	1		1				
15	C/gnd	not(A*B)				1				1				1				
16	C/vdd	not(D+A*B)		1				1				1						
17	D/N1	not(A*(B+C+notD)+B*notD +not((A+B)*D)			1		1		1		1		1					
18	D/Q	D*(not(A*B))	1		1	1	1		1	1	1		1	1				
19	D/gnd	not(A*B)				1				1				1				
20	D/vdd	not(C+A*B)			1				1				1					
21	N1/Q	not(A*B*C*D)				1				1				1				
22	N1/gnd	sa-0 for Q	1	1	1		1	1	1		1	1	1					
23	N1/vdd	not(C*D)													1	1	1	
24	Q/gnd	sa-0 at Q	1	1	1		1	1	1		1	1	1					
25	Q/vdd	sa-1 at Q				1				1				1	1	1	1	1

7. EXPERIMENTAL DATA

Experiments were carried out with ISCAS'85, ISCAS'89 and ITC'99 benchmark circuits (Table 3). The goal of the experimental research was to evaluate the achievable high-level module-based diagnostic resolution (Table 4) and to compare it with traditional stuck-at fault based diagnostic resolution (Table 5). All the circuits were presented as networks of modules, where as modules the fan-out free regions (FFR) were selected. Three different types of tests were used for evaluating the diagnostic resolution (Table 4): long pseudorandom test sequences, short deterministic test sequences, and combination of both sequences. The pseudorandom test sequences were generated by LFSR-based test generator and optimized by selecting proper seeds and polynomials. The LFSR was stopped when no useful test patterns were found in a reasonable time. The pseudorandom test was selected to have more test patterns for detecting the same fault with the goal to improve the distinguishability of the faults. The deterministic test patterns were synthesized by a genetic test generator. The

quality of test patterns, i.e. the maximum possible test coverage was not the target. The goal of the paper was not to generate the best test sequences, but rather to evaluate and compare the diagnosability, i.e. diagnostic resolutions for two methods and for a given test sequence. To generate diagnostic tests with as good as possible diagnostic resolution is the task not considered in this paper.

In Table 3, the characteristic data of the benchmark circuits are presented. In columns 2 to 4, the numbers of inputs, outputs and gates are presented, respectively. Column 5 lists the number of modules, and in column 6 the complexity reduction (Gates/Modules) in the module-level model is shown compared to the gate-level network. The reduction on average 5.7 times reflects the decrease of the complexity in fault dictionaries in case of using FFR-modules instead of gates and SAF lists.

In Table 4, the results of diagnostic resolution for three different test types are presented. In each section, the characteristics of tests are given (test length, and stuck-at fault coverage (SAF %)), and the diagnostic resolutions (the number of suspected modules) after the first and second phases of high-level diagnosis are depicted. The diagnostic resolution can be improved by generating additional tests.

Table 5 shows the comparison results of the proposed module-level diagnosis with traditional SAF-based diagnosis based on pseudorandom tests for ISCAS'85 benchmark circuits. The diagnostic resolution at the module level on average is similar to or even better than SAF-based diagnostic resolution.

Table 3. Characteristic data of benchmark circuits

Circuits	Inputs	Outputs	Gates	Modules	Complexity reduction
c1908	33	25	1 394	248	5.6
c2670	233	140	2 075	430	4.8
c3540	50	22	2 784	378	7.4
c5315	178	123	4 319	633	6.8
c7552	207	108	5 795	920	6.3
s9234	247	250	5 597	1 263	4.4
s13207	700	790	12 441	2 014	6.2
s15850	611	684	14 841	2 202	6.7
s35932	1 763	2 048	32 624	7 343	4.4
b12_C	126	127	1 000	512	2.0
b14_C	277	299	19 491	2 708	7.2
b15_C	485	519	18 248	2 872	6.5
Average					5.7

Table 4. Diagnostic resolution for high-level module-based diagnosis

Circuits	Pseudorandom test				Deterministic test				Both tests			
	Test length	SAF, %	Average resolution		Test length	SAF, %	Average resolution		Test length	SAF, %	Average resolution	
			Phase 1	Phase 2			Phase 1	Phase 2			Phase 1	Phase 2
c1908	4 420	99.48	86.61	1.42	121	99.48	110.67	2.08	4 541	99.48	86.17	1.41
c2670	22 682	88.65	223.86	3.45	72	88.46	235.82	4.39	22 754	90.36	221.05	3.24
c3540	9 631	95.54	69.63	2.53	155	95.54	85.77	3.89	9 786	95.54	69.62	2.51
c5315	1 793	98.89	213.23	1.85	119	98.89	238.46	3.45	1 912	98.89	212.00	1.84
c7552	24 337	94.08	287.55	2.60	188	95.18	304.08	2.90	24 525	95.52	285.78	2.58
s9234	29 873	86.21	271.31	6.32	365	92.19	398.04	6.75	30 238	92.23	205.01	4.50
s13207	29 694	96.81	781.34	10.53	434	98.19	1 035.53	9.98	30 128	98.19	701.69	7.60
s15850	29 360	90.99	669.44	10.50	370	94.19	886.01	10.45	29 730	94.22	557.32	4.12
s35932	168	90.81	3 748.42	63.85	55	88.49	4 391.17	17.87	223	90.81	3 554.58	6.45
b12_C	27 205	98.33	151.42	4.15	155	99.77	239.68	5.05	27 360	99.83	126.87	2.83
b14_C	29 388	88.86	249.80	18.29	848	92.76	655.79	6.15	30 236	95.69	531.72	2.69
b15_C	39 198	86.56	756.54	14.06	498	88.77	1 069.16	17.67	39 696	94.74	929.14	11.08

Table 5. Comparison of SAF- and module-based diagnosis

Circuits	Test characteristics		Diagnostic resolution	
	Length	SAF, %	SAF-based	Module-based
c1908	4 420	99.48	2.9	1.4
c2670	22 682	88.65	4.1	3.5
c3540	9 631	95.54	2.2	2.5
c5315	1 793	98.89	2.2	1.9
c7552	24 337	94.08	2.6	2.6

8. CONCLUSIONS

The main drawbacks of the traditional cause–effect diagnosis methods are the dependence on the fault model and poor scalability of fault dictionaries.

We consider here fault diagnosis as the following hierarchical procedure: first, to determine a set of suspected faulty modules in a circuit by using high-level module-based fault dictionary for the whole circuit, and second, to locate the physical defects in the suspected faulty modules either by low-level dictionaries for library components, or by defect reasoning in case of arbitrary subcircuits defined as modules.

The proposed high-level fault diagnosis is fault independent, since instead of specific faults, faulty structural components called modules are localized. No restrictions to fault multiplicity inside the modules are set.

On the other hand, the complexity of the proposed method compared to the flat-level fault model based diagnosis is reduced. The size of the high-level fault dictionary for the whole circuit depends linearly only on the number of modules to be determined as faulty or not faulty, and not on the number of possible faults or defects as traditionally. The size of the modules, however, will be the trade-off between the complexity of the high-level dictionary and the diagnostic resolution.

The complexity of the diagnosis problem was reduced on average 5.7 times for the case of FFR-based modules compared to gate-level diagnosis. The diagnostic resolution in the module-based diagnosis was similar to, or even better than in the case of SAF-based fault diagnosis.

The high diagnostic resolution at the module level was achieved by implementing a novel high-level effect–cause reasoning, based on the concept of functional fault model (conditional SAF model). This concept allowed to map physical defects from transistor level to the module level, and to carry out indirect defect-based reasoning of faulty modules at higher level. The same concept can be used to improve the diagnostic exactness by low-level defect reasoning in the faulty modules.

ACKNOWLEDGEMENTS

The work has been supported by Estonian Science Foundation (grants Nos 7068 and 7483), EC FP7 IST project DIAMOND, ELIKO Development Centre, Estonian IT Foundation “Tiger University", and Research Centre CEBE, funded by EU Structural Funds.

REFERENCES

1. Venkataraman, S. and Drummonds, S. B. Poirot: Applications of a logic fault diagnosis tool. *IEEE Trans. Design Test Computers*, 2001, **18**, 19–29.
2. Huisman, L. M. Diagnosing arbitrary defects in logic designs using single location at a time (SLAT). *IEEE Trans. CAD*, 2004, **23**, 91–101.
3. Holst, S. and Wunderlich, H.-J. Adaptive debug and diagnosis without fault dictionaries. In *Proc. 12th European Test Symposium*. Freiburg, 2007, 7–12.
4. Rousset, A., Bosio, A., Girard, P., Landrault, C., Pravossoudovitch, S. and Virazel, A. Derric: A tool for unified logic diagnosis. In *Proc. 12th European Test Symposium*. Freiburg, 2007, 13–20.
5. Wang, L. T., Wu, C. W. and Wen, X. VLSI test principles and architectures. In *Design for Testability*. Elsevier, 2006.
6. Zhuo, L., Lu, X., Qiu, W., Shi, W. and Walker, D. M. H. A circuit level fault model for resistive opens and bridges. In *Proc. VLSI Test Symp.* Napa, CA, 2003, 379–384.
7. Liu, C. Compact dictionaries for fault diagnosis in scan-BIST. *IEEE Trans. Computers*, 2004, **53**, 775–780.
8. Acken, J. M. and Millman, S. D. Accurate modeling and simulating of bridge faults. In *Proc. Custom Integrated Circuits Conference*. San Diego, CA, 1991, 17.4.1–17.4.4.
9. Jain, S. K. and Agrawal, V. D. Modeling and test generation algorithms for MOS circuits. *IEEE Trans. Computers*, 1985, **C-34**, 426–433.
10. Lee, H. K. and Ha, D. S. SOPRANO: An efficient automatic test pattern generator for stuck-open faults in CMOS combinational circuits. In *Proc. Design Automation Conference*. Orlando, FL, 1990, 660–666.
11. Kristic, A. and Cheng, K. T. *Delay Fault Testing for VLSI Circuits*. Kluwer, Boston, MA, 1998.
12. Roth, J. P. Diagnosis of automata failures: A calculus and a method. *IBM J. Res. Developm.*, 1966, **10**, 278–291.
13. Blanton, R. D. and Hayes, J. P. On the properties of the input pattern fault model. *ACM Trans. Des. Automat. Electron. Syst.*, 2003, **8**, 108–124.
14. Keller, K. B. Hierarchical pattern faults for describing logic circuit failure mechanisms. US Patent 5546408, Aug. 13, 1994.
15. Ubar, R. Detection of suspected faults in combinational circuits by solving boolean differential equations. *Automation and Remote Control*, 1980, **40**, 1693–1703 (Plenum Publ. Corp., USA).
16. Ubar, R., Devadze, S., Raik, J. and Jutman, A. Fast fault simulation for extended class of faults in scan-path circuits. In *Proc. 5th IEEE International Symposium DELTA*. Ho Chi Minh City, Vietnam, 2010, 14–19.
17. Mahlstedt, U., Alt, J. and Hollenbeck, I. Deterministic test generation for non-classical faults on the gate level. In *Proc. 4th Asian Test Symposium*. Bangalore, 1995, 244–251.
18. Holst, S. and Wunderlich, H.-J. Adaptive debug and diagnosis without fault dictionaries. In *Proc. 13th European Test Symposium*. Verbania, Italy, 2008, 199–204.
19. Ubar, R., Kuzmicz, W., Pleskacz, W. and Raik, J. Defect-oriented fault simulation and test generation in digital circuits. In *Proc. 2nd International Symposium on Quality of Electronic Design*. San Jose, CA, 2001, 365–371.

Hierarhiline füüsikaliste defektide analüüs digitaalskeemides

Sergei Kostin, Raimund Ubar, Jaan Raik ja Marina Brik

On kirjeldatud hierarhilist meetodit füüsikaliste defektide diagnoosiks kombinatoorsetes digitaalskeemides. Skeem esitatakse moodulite võrguna. Mooduliteks võivad olla nii skeemis kasutatavad teegi komponendid (näiteks keerukad loogikaelemendid) kui ka suvalised allskeemid. Kõrgtaseme rikete diagnoos viiakse läbi kahes faasis. Esimeses faasis leitakse kahtlustatavad vigased moodulid, kasutades kompaktseid kõrgtaseme diagnostika moodulsõnastikke. Selliste sõnastike suurus on lineaarses sõltuvuses moodulite arvust skeemis. Teises faasis kahtlustatavate vigaste moodulite arv "surutakse kokku" moodulite defektse käitumise analüüsi abil testeksperimendi käigus. Madalal loogikatasandil lokaliseeritakse füüsikalised defektid kahtlustatavates moodulites kas moodulite diagnostikasõnastike abil või tagajärg–põhjus-analüüsimeetodil otseselt moodulite sees. Väljatöötatud uus rikete diagnoosi meetod aitab üle saada keerukusprobleemidest suurte elementide arvu korral diagnoositavates skeemides. Eksperimentaaltulemused kinnitavad uue meetodi kõrget diagnoosiresolutsiooni.