Fault Collapsing in Digital Circuits Using Fast Fault Dominance and Equivalence Analysis with SSBDDs

Raimund Ubar, Lembit Jürimägi, Elmet Orasson, Jaan Raik

Fault Collapsing in Digital Circuits using Fast Fault Dominance and Equivalence Analysis with SSBDDs

Raimund Ubar, Lembit Jürimägi, Elmet Orasson, Jaan Raik

Department of Computer Engineering, TTU, Ehitajate tee 5, 19086 Tallinn, Estonia

E-mails: raiub@pld.ttu.ee, lembit.jyrimagi@gmail.com, elmet@pld.ttu.ee, jaan@pld.ttu.ee

Abstract. The paper presents a new method and an algorithm for structural fault collapsing to reduce the search space for test generation, to speed up fault simulation and to make the fault diagnosis easier in digital circuits. The proposed method is based on hierarchical topology analysis of the circuit description at two levels. First, the gate-level circuit will be converted into a macro-level network of Fan-out Free Regions (FFR) each of them represented as a special type of structural BDD. This conversion procedure represents as a side-effect the first step of fault collapsing, resulting in a compressed Structurally Synthesized BDD (SSBDD) model explicitly representing the collapsed set of representative fault sites. The paper presents an algorithm which implements a complementary step of further fault collapsing. This algorithm is carried out at the macro-level FFR-network by topological reasoning of equivalence and dominance relations between the nodes of the SSBDDs. The algorithm has linear complexity and is implemented as a continuous scalable fault eliminating procedure. We introduce higher and lower bounds for fault collapsing and provide statistics of distribution of fault collapsing results over a broad set of benchmark circuits. Experimental research has demonstrated considerably better results of structural fault collapsing in comparison with state-of-the-art.

Keywords: combinational circuits, fault collapsing, fault equivalence and dominance, Binary Decision Diagrams, lower and higher bounds

1 Introduction

Fault collapsing is a procedure which is applied to reduce the number of faults of a given circuit to be targeted for testing purposes. Using a reduced set of only representative faults instead of a full set of faults has the goal to minimize the efforts in many test related tasks like test pattern generation, fault simulation for test quality evaluation, fault diagnosis, circuit testability evaluation etc.

The methods of fault collapsing are classified as structural and functional. Structural fault collapsing uses only the topology of the circuit whereas functional fault collapsing uses the circuit functional properties inherent in the circuit.

There are two classical ways used for structural fault collapsing: *fault equivalence* based and *fault dominance* based collapsing [1]. A fault f_j is said to *dominate* a fault f_i if every test that detects f_i also detects f_j. If f_j dominates f_i, only f_i needs to be considered during test generation. When two faults dominate each other, they are called *equivalent*. If two faults are equivalent, only one of them needs to be considered during test gene-ration or fault diagnosis. *Structural fault collapsing* uses the topology of the circuit structure. For example, a *stuck-at* 0 fault (SAF y/0) at the output y of AND gate is equivalent to all of the SAF x/0 faults at its inputs x_i. In a similar way, SAF y/1 at the output of AND gate dominates all the input SAF x/1 faults. The classical structural approaches to fault collapsing are based on gate-level circuit processing. An approach based on *fault-folding* was introduced in [2] for structural collapsing faults, using the iterative analysis of gate fault equivalence and dominance relations. Since structural fault collapsing is very fast, it is employed in many Automated Test Pattern Generators (ATPG) [3,4].

Functional fault collapsing uses the circuit's functional information to establish equivalence and dominance relations. Two faults are functionally equivalent if they produce identical faulty functions [5] or we can say, two faults are functionally equivalent if we cannot distinguish them at the Primary Outputs (PO) with any input test vector [6]. Functional fault collapsing is generally regarded as very difficult to compute because it deals with the whole function of the circuit under test. In [7] it has been shown that the algorithmic complexity for identifying functionally equivalent faults is similar to that of ATPG.

Approximate fault collapsing via simulation has been proposed in [8]. In [9], a metric called *level of similarity* has been introduced and is efficiently used to improve the level of approximation. The fault collapsing suffers from the danger that if a fault in the collapsed fault set remains undetected then all other faults equivalent or dominating this fault removed from the collapsed fault set remain undetected as well. In [10], a safety parameter s to restrict the use of the dominance relation is introduced, and a safe fault collapsing method with a level of safety s is proposed.

The potentials of hierarchical fault collapsing were discussed in [11]. It was shown that hierarchical approach to fault collapsing gives more possibilities to increase the efficiency compared to the non-hierarchical one. An algorithm based on *transitive closures* on the *dominance graphs* has been proposed [12, 13], which enables more efficient hierarchical fault collapsing. It is a graph theoretic, fault independent and polynomial technique for functional fault collapsing.

In [14], functional dominance has been used to collapse the fault sets. However, this technique requires quadratic number of ATPG runs to obtain the collapsed fault set. An improvement was proposed in [15], which has the linear complexity regarding the number of ATPG runs. Since ATPG itself is used for learning functional dominance relations, both these techniques are suitable for small circuits only, but they can be helpful when combined with hierarchical fault collapsing. In [7] two theorems were introduced based on unique requirements and D-Frontiers of faults to extract equivalence and dominance relations. Similar approach was used in [16] based on the dominator theory for identifying more functionally equivalent fault pairs. In [17] a generalized dominance approach requires similar or lower run-times than that of [7].

A collapsed fault set helps generating smaller test sets for achieving the desired fault coverage, and it contributes to fault diagnosis as well. Since fault diagnosis deals with fault pairs, a linear reduction of the number of faults would result in a quadratic reduction of the target pairs.

In [5,15], a novel diagnostic fault equivalence and dominance technique was proposed. A new method for fault collapsing for diagnosis called dominance with subfaults was proposed in [18]. The method allows reducing the diagnosis search space. A framework where equivalence and dominance relations are defined for fault pairs is introduced in [19]. A fault pair collapsing is described, where fault pairs are removed from consideration under diagnostic fault simulation and test generation, since they are guaranteed to be distinguished when other pairs are distinguished. A technique to speed-up diagnosis via dominance relations between sets of faults using function-based techniques was proposed in [20]. Due to the high memory and time complexity this approach is applicable for small circuits only. All the listed techniques are fault oriented approaches, i.e. they consider a fault-pair at a time and use ATPG for identification of equivalence or dominance relations. In [21], a dynamic fault collapsing procedure is presented for fault diagnosis, where the faults are collapsed during the diagnostic test pattern generation contrary to the traditional static approaches described above where the faults are collapsed before test generation.

One of the main limitations of the described methods is that there is no evidence that investing more effort in fault collapsing reduces the total test generation time [10]. The reason is that most of the methods are using ATPG itself as a tool for fault collapsing, or they are usable only for small circuits because of the high computing complexity.

In this paper we concentrate on the structural fault-independent fault collapsing based on the topology analysis of the circuit. We target the minimal necessary set of representative faults as objectives for both, test generation and fault simulation. To cope with the complexity problem in case of big circuits, we use a hierarchical approach to structural fault collapsing, which is based on the topology analysis of the circuit at two levels – gate- and macro-levels, where the Fan-out-Free Regions (FFR) are regarded as macros. The proposed method is characterized at both levels by linear complexity which allows achieving high speed in fault collapsing, and provides smaller collapsed representative fault sets compared to other known structural methods. Due to low complexity, the method is well scalable and is therefore usable for large circuits where the functional fault collapsing methods give up because of the complexity.

The approach we propose consists of two consecutive procedures. During the first procedure, fault collapsing is carried out at the gate level by superposition of Binary Decision Diagrams (BDD) [22] of logic gates with the main goal of constructing a higher macro-level model of the circuit in form of Structurally Synthesized BDDs (SSBDD) [23,24] where to each FFR an SSBDD corresponds. The fault collapsing can be regarded here as a side-effect (byproduct) of the SSBDD model synthesis. The second procedure, complementary part of the approach, is carried out at the higher macro-level by topological analysis of SSBDDs. Both parts of the fault collapsing procedure have linear complexity. It has been shown that SSBDDs can be efficiently

used for fault simulation, outperforming in the speed state-of-the-art fault simulators [25, 26]. In this paper we show the possibility of additional fault collapsing using SSBDDs, which in turn can lead to further speed-up of fault simulation.

The paper is organized as follows. In Section 2 we give an overview of SSBDDs and in Section 3 we describe the synthesis of SSBDDs as the first step of gate-level fault collapsing. Section 4 presents the main theoretical concepts for the analysis of equivalence and dominance relations between the faults in the higher level FFR-networks modeled with SSBDDs, and Section 5 describes the algorithm of fault collapsing with SSBDDs. In Section 6, lower and higher bounds for fault collapsing are given. Section 7 presents experimental data, and Section 8 concludes the paper.

2 Structurally Synthesized BDD

Binary Decision Diagrams (BDD) have become by today a state-of-the-art data structure in VLSI CAD for representation and manipulation of Boolean functions. BDDs were first introduced for logic simulation in [27], and for test generation in [28,29]. In 1986, Bryant proposed a new data structure called *Reduced Ordered BDDs* (ROBDDs) [22]. He showed simplicity of the graph manipulation and proved the model canonicity that made BDDs one of the most popular representations of Boolean functions. This model, however, suffers from the memory explosion problem, which limits its usability for large designs. Moreover, it cannot be used as a model for representing structural information about the design like representation of faults directly in the model. In [23, 28, 30], *Structurally Synthesized BDDs* (SSBDDs) were proposed with the goal to represent the structural features of circuits. The most significant difference between the function-based BDDs [22] and SSBDDs [23] is the method how they are generated. While BDDs are generated on the functional basis by *Shannon's expansions*, which handle only the Boolean function of the logic circuit, the SSBDD models are generated by a *superposition procedure* that extracts both, functions and data about structural signal paths of the circuit. The linear complexity of the SSBDD model results from the fact that a digital circuit is represented as a system of SSBDDs, where for each FFR a separate SSBDD is generated.

SSBDDs are generated by iterative superposition of library BDDs for simple or complex gates, guided by the structure of the given circuit. To avoid the explosion of the complexity of the SSBDD model, and to keep its size as minimal as possible, the superposition of BDDs is stopped at fan-out stems of the circuit. Using this restriction, to each FFR in the circuit an SSBDD will be created where a signal path in the FFR corresponds to each node in an SSBDD.

Example 1. An example of a combinational circuit and its SSBDD is depicted in Fig.1. The SSBDD represents an FFR of the circuit obtained after cutting all the input fan-out branches of the circuit. This FFR can be described by the following Boolean expression:

$$y = f(X) = \left(x_1 x_{21} \vee \left(x_{22} x_3 \vee x_4(\overline{x_5} \vee \overline{x_{61}})\right)\overline{x_{71}}\right)x_{81} \vee x_{82} x_9 (x_{72} \vee \overline{x_{62}})\overline{x_{10}}$$

The non-terminal (internal) nodes of the SSBDD are labeled by the input variables of the FFR. To differentiate the fan-out branch variables from the fan-out stem variable

we introduce for each of them a second subscript. The node variables may be inverted.

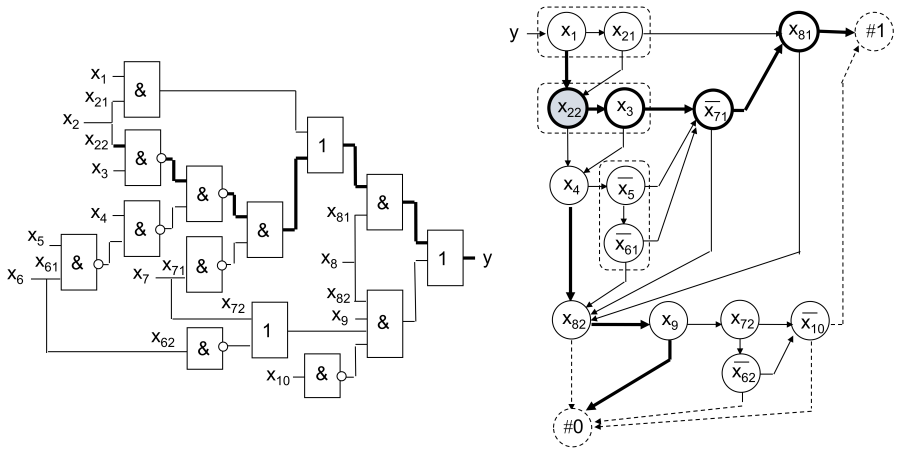

Fig. 1. Combinational circuit with a single output and its representation as an SSBDD

When using SSBDDs for calculating the output signals at given test patterns, we have to traverse the graph starting from the root node up to a terminal node guided by the input pattern. Let us agree that we exit each node during simulation to the right if the node variable has value 1, and downwards if the value is 0. In this case we don't need to label the edges in the graph by the values of the node variables on Figures. Entering the terminal node #1 as the outcome of graph traversing will mean the result of simulation $y = 1$, and entering the terminal node #0 will mean $y = 0$.

X	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y
X^t	0	1	1	0	-	-	0	1	0	-	1
Tested: $x_{22} \equiv 0$, $x_3 \equiv 0$, $x_{71} \equiv 1$, $x_{81} \equiv 0$											

Fig. 2. Test pattern for detecting selected faults SAF/0 or SAF/1 for the circuit in Fig.1

Example 2. For the circuit in Fig.1 with function $y = f(X)$ the output signal y for the given input pattern X^t in Fig.2 will be $y = 1$. During simulation of this pattern on the SSBDD, the following nodes are traversed: x_1, x_{22}, x_3, $\neg x_{71}$, x_{81}, #1 (shown by bold lines).

SSBDD model has several features that make it attractive compared to other commonly used mathematical models, such as conventional BDDs or gate-level netlists [31,32]. The worst-case complexity (time) of generating SSBDD model from a circuit's netlist is linear in respect to the number of gates, while it is exponential for common BDDs. The size of the SSBDD model is always linear in respect to the circuit size (BDDs can be of exponential size). Compared to the gate-level representa-

tion, SSBDDs help to reduce the complexity of the circuit by representing them as FFR (SSBDD) networks whereas the algorithms of processing the network components do not need dedicated treatment of the components described usually by design libraries. Moreover, instead of considering each gate separately, it deals with *macros* – FFRs represented by SSBDDs.

The most important feature of the SSBDD model is that it preserves structural information about the circuit while traditional BDDs do not. This is why differently from traditional BDDs, SSBDDs support structural test generation [23, 28] and fault simulation [25, 26, 30] for gate-level structural faults in terms of faulty signal paths with representing the faulty paths explicitly in the model. Each node in the SSBDD represents a signal path in the corresponding circuit, and the faults of the nodes represent the faults in signal paths.

For example, the SSBDD in Fig.1 consists of 14 internal nodes where each of them represents a corresponding signal path of the total 14 paths in the circuit in Fig.1 (the correspondence is shown by the variables x where x_i denote input signals, and x_{ij} denote the signals at the fan-out branches). The one-to-one mapping between the nodes in SSBDD and the paths in the circuit is the result of the SSBDD synthesis from the netlist of the given circuit. The synthesis process is presented in Section 3.

Note, that the SSBDD model in Fig.1 represents only the FFR of the circuit. The faults of the input fan-out stems x_2, x_6, x_7 and x_8 should be handled separately, either by introducing trivial single-node BDDs to represent the input fan-out stems, or by modeling the stem faults as multiple faults in the nodes which represent the fan-out branches.

3 Synthesis of SSBDDs

Consider first, the following graph theory related definitions of the BDDs (SSBDDs). We use the graph theory notations instead of traditional *ite* expressions [22] because all the test related procedures based on SSBDDs are based on the topological reasoning rather than on symbolic manipulations as is traditionally the case for BDDs.

Definition 1. A BDD that represents a Boolean function $y=f(X)$, $X = (x_1,x_2, \ldots , x_n)$, is a directed acyclic graph $G_y = (y,M,\Gamma,X)$ with a set of nodes M and mapping Γ from M to M. $\Gamma(m) \subset M$ denotes the set of *successor* nodes of $m \in M$, and $\Gamma^{-1}(m) \subset M$ denotes the set of *predecessor* nodes of m. M consists of two types of nodes: *internal* (non-terminal) M^N and *terminal* M^T. For terminal nodes m^T we have $\Gamma(m^T) = \varnothing$. There is a single node $m_0 \in M$ where $\Gamma^{-1}(m) = \varnothing$ called the *root* node. A terminal node $m^T \in \{m^{T.0}, m^{T.1}\}$ is labeled by a constant $x(m^T) \in \{0,1\}$ and is called *leaf*, while all the nodes $m \in M^N$ are labeled by Boolean variables $x(m) \in X$, and have exactly two successor nodes $\Gamma(m) = \{m^0, m^1\}$.

Definition 2. We say, the edge $l(m, m^e)$ between nodes m and $m^e \in \Gamma(m)$ is *activated* when the node variable $x(m)$ is assigned to one of the values $e \in \{0,1\}$. We say, a path $l(m_i, m_j)$ between the nodes m_i and m_j is activated if all the edges which form the path are activated.

Definition 3. We say that a BDD $G_y = (y,M,\Gamma,X)$ represents a Boolean function $y=f(X)$, iff for every possible vector $X^t \in \{0,1\}^n$, a path $l(m_0, m^T)$ is activated so that $y = f(X^t) = x(m^T)$.

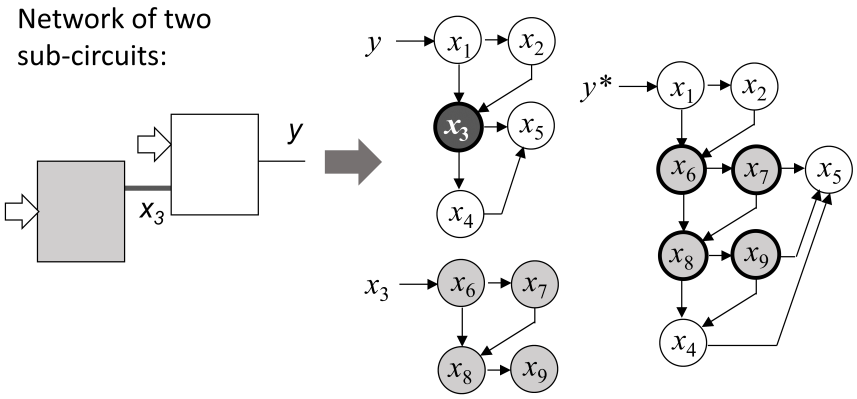

Fig. 3. Superposition of two SSBDDs

The main idea of superposition of BDDs as the basis procedure of SSBDDs proposed first in [23,28], is illustrated in Example 3.

Example 3. Let us have in Fig.3 a network of two components y and x_3 in Fig3, connected by the wire x_3. The components implement the following functions:

$$y = x_1x_2 \vee (x_3 \vee x_4)x_5, \;\; x_3 = x_6x_7 \vee x_8x_9$$

The components y and x_3 are represented by the SSBDDs y and x_3, respectively. For simplicity, we have omitted in SSBDDs the terminal nodes, with introducing the agreement that leaving the graph to the right means entering the terminal node #1, and leaving the graph down means entering the terminal node #0. Superposition of the two graphs y and x_3 is equivalent of merging the two components y and x_3 into a single component y* which implements the function:

$$y* = x_1x_2 \vee (x_6x_7 \vee x_8x_9 \vee x_4)x_5$$

To carry out this operation we have to substitute the node x_3 in the graph y with the graph x_3. To do that, we:

(1) connect the incoming edges of the node x_3 in graph y with the root node x_6 of graph x_3;
(2) connect all the nodes in the graph x_3 , which enter into #1, with the right-hand neighbor of x_3 in graph y and
(3) connect all the nodes in the graph x_3 , which enter into #0, with the down-hand neighbor of x_3 in graph y.

The new SSBDD y* represents the function of the network with two merged components y and x_3.

Let us have, in general case, a gate-level circuit C where each gate is represented by an elementary BDD. The procedure of generating the SSBDD model $G(C)$ for C starts from the BDD of an output gate, and uses iteratively the superposition procedure where a node in a BDD is replaced by another BDD [23].

Consider two BDDs, G_y for the output gate $g_y \in C$ with output y, and G_x for the gate g_x connected to the input x of g_y. Let us call further, for simplicity, a node in a BDD labeled by a variable z as a "node z". By substitution of the node x in G_y with the BDD G_x we create from G_y a new SSBDD G_y' which represents now the extended network consisting of g_y and g_x. We call the new graph as SSBDD because the new nodes z in G_y' which belonged to G_x represent the signal paths from the inputs z of the gate g_x via the connection line x between the two gates up to the output y of the gate g_y.

The procedure of the *superposition* of a node m labeled by x in BDD G_y with BDD G_x can be presented as follows.

Procedure 1. Superposition of BDDs

(1) The node m labeled by x is removed from G_y.
(2) All the edges in G_x connected to terminal nodes $m^{T,e}$ in G_x will be cut and then connected, respectively, to the successors m^e of the node m in G_y.
(3) All the incoming edges of m in G_y will be now incoming edges for the root node m_0 in G_x.

By applying *Procedure 1* for two BDDs, we reduce the current model by one node and by one BDD. Suppose, the label variable x of a node m in a BDD G_y corresponds to the output of the gate g_x with k output branches. This means that the variable x is used as a label for k different nodes in the initial model as a set of BDDs. If we would proceed the superposition of graphs beyond the fan-out stem x, and would try to replace all the k nodes labeled by x with the BDD G_x, the complexity of the model would increase instead of reduction, i.e. the k nodes will be replaced by k BDDs G_x. Therefore, to keep the complexity of the final SSBDD model linear with the size of the circuit, and to reach the maximum compression of the initial model given as a set of elementary BDDs, we generate SSBDDs only for FFRs. Hence, at each fan-out stem we start a new superposition procedure for the next FFR.

Definition 4. A BDD which is constructed for a given FFR by Procedure 1 is called *structurally synthesized* BDD (SSBDD).

Corollary 1. It is easy to conclude from Procedure 1 that in the SSBDD G_y generated for the given FFR C_y with a function $y = f(x_1, x_2, \ldots, x_n)$, there are exactly n nodes with labels $x_1, x_2, \ldots, x_n$, and the node m with label x_i represents a unique signal path in C_y from the input x_i to the output y.

Corollary 2. Since all the SAF faults at the inputs of FFR according to the approach of fault folding [2] form the collapsed fault set of the FFR, and since all these faults are represented by the faults at the nodes of the corresponding SSBDD, then the creation of the SSBDD is equivalent to the fault collapsing procedure similar to fault folding.

Theorem 1. Let $G(C)$ be the SSBDD model generated for the combinational circuit C by Procedure 1. Then, any set of tests that checks all the SAF faults at the nodes of $G(C)$ checks all the SAF faults in C.

Proof. The proof follows from Corollaries 1 and 2, and from Theorem 5 in [2].

Unlike the traditional gate level approaches to test generation and fault simulation that use the collapsed fault list apart from the simulation model, the SSBDD based test generation and fault simulation are carried out on the macro-level (FFRs as macros) with direct representation of the faults in the model. Therefore there is no need for separate fault list to be used during test generation and fault simulation.

Example 4. The node x_{22} in the SSBDD represents the path from x_{22} to y in the circuit shown by bold lines in Fig.1. On the other hand, the stuck-at faults SAF $y/0$ and SAF $y/1$ dominate the faults $x_{22}/0$ and $x_{22}/1$, respectively. The same dominance relation stands for all the faults along the bold path from x_{22} to y, regarding to the faults at x_{22}.

From this dominance relation, it results that all the faults along the signal path from x_{22} to y, except $x_{22}/0$ and $x_{22}/1$, can be collapsed. The two faults at x_{22} will form the representative fault subset for the full signal path from x_{22} to y. But, exactly these faults are represented in the SSBDD as the faults of the node x_{22}.

From above it follows that the SSBDD model can be regarded as the model where all the collapsed faults are removed and the fault sites are not visible either. This fault collapsing result is similar to that of fault folding method presented in [2].

The minimum size of SSBDDs, generated with Procedure 1 is always fixed and determined by the circuit structure. Let us denote N_{SSBDD} as the number of nodes in the SSBDD model, as the size characteristic of SSBDDs. Let $N_{Signals}$ be the number of lines, and N_G is the number of gates in the circuit represented by SSBDD. In [16] it has been shown that the number of nodes in SSBDDs can be calculated as

$$N_{SSBDD} = N_{Signals} - N_G$$

Since a digital circuit can be represented both by gate-level and by FFR-level SSBDDs, then in order to compare the gain in fault collapsing we get from translating the gate level SSBDD into FFR-level SSBDD, let us use the same units for measuring the size of SSBDDs in both cases – the number of SSBDD nodes in the model.

Denote s – as the number of inputs of the circuit, s_0 – as the number of inputs with no fan-outs , s_1 – as the number of internal lines with no fan-outs, s_k – as the number of nets in the circuit with k fan-outs (k >1), n – as the number of outputs, and m – as the maximum number of fan-out branches over all fan-out stems in the circuit.

In [33] we have developed the following estimations for the sizes of SSBDDS for the gate-level N_{gate} and for FFR-level N_{SSBDD} cases:

$$N_{Gate} = s_0 + s_1 + \sum_{k=2}^{m} s_k(k+1)$$

$$N_{SSBDD} = s + \sum_{k=2}^{m} s_k(k+0)$$

Since in both cases for the stuck-at fault model, the number of nodes must be doubled to get the number of faults, then the ratio N_{gate} / N_{SSBDD} will characterize the gain in fault collapsing as the side-effect of FFR-level SSBDD synthesis from the initial gate-level SSBDD model, and the subtraction N_{gate} - N_{SSBDD} gives the exact number of collapsed faults thanks to the SSBDD synthesis.

Example 5. For the FFR of the circuit and its SSBDD in Fig.1 we get N_{gate} = 30 and N_{SSBDD} = 18. The values of the arguments of the formulas for N_{gate} and N_{SSBDD} are depicted in Table 1. Hence, the gain in the SSBDD sizes, in this example, is 1.7, and the number of collapsed faults is 12. Note, the SSBDD with 14 nodes in Fig. 1 represents only the FFR part of the circuit. To get the full FFR-level SSBDD model, we have to include 4 single node SSBDDs for representing the 4 fan-out inputs in the circuit.

Table 1. Calculation of the number of nodes for 2 types of SSBDDs

	s_0	s_1	s_k	m	s	N
N_{gate}	6	12	4	2	-	N_{gate} = 6 + 12 +4*3 = 30
N_{SSBDD}	-	-	4	2	10	N_{SSBDD} = 10 + 4*2 = 18

To summarize, the procedure of SSBDD synthesis can be regarded as the first part of fault collapsing for the given circuit. In the next section we will discuss the possibility of additional fault collapsing directly on the SSBDD model.

4 Fault Equivalence and Fault Dominance on the SSBDD Model

The second part of fault collapsing will consist of the processing of the SSBDD model with the goal to find additional set of faults which may be collapsed using the equivalence and dominance relationship on the SSBDD level. Since the nodes of SSBDDs represent signal paths on the gate-level circuit then each node related fault on the SSBDD to be collapsed is equivalent to all the related gate-level faults on the signal path represented by the node. Whereas the first part of fault collapsing was carried out by tracing the signal paths in the gate-level circuit level, then the second part concentrates on the path analysis at the higher FFR-level by tracing the paths on the SSBDDs

Definition 5. Let us call a path $L(a, b)$ in the SSBDD between two nodes a and b, *activated* by a given input pattern X^t, if by traversing the graph under guidance of X^t, the node b will be reached from a.

In SSBDD-based test generation the targets are node related faults. As explained in [23], to test a node m in an SSBDD we have to activate three paths in it: (1) $L(m_0, m)$ from the root node m_0 to m, (2) $L(m^1, \#1)$ from the neighbor m^1 of m to the terminal node #1, and (3) $L(m^0, \#0)$ from m^0 to #0.

Example 6. To test the node x_{22} in the SSBDD in Fig.1 we have to activate three paths in it: (1) $L(x_1, x_{22})$ from the root node x_1 to x_{22}, (2) $L(x_3, \#1)$ from x_3 to the terminal node #1, and (3) $L(x_4, \#0)$ from x_4 to #0. When we assign x_{22} = 1 then the activa-

tion of the listed paths produce a test pattern X^t which detects the fault SAF $x_{22} \equiv 0$. The pattern X^t which activates these paths (bold lines in Fig.1) is depicted in Fig.2.

Definition 6. Let us call the path which is activated from the root node up to the one of the terminal nodes, the *full activated path* in SSBDD. The full activated path which terminates in the node #1 (#0) is called 1-*path* (0-*path*). The nodes traversed along the 1-path (0-path) in direction to 1 (0), are called 1-*nodes* (0-*nodes*).

Example 7. The path $L(x_1, \#1) = (x_1, x_{22}, x_3, \neg x_7, x_{81}, \#1)$ in Fig.1, activated by the pattern in Fig.2, is 1-path, the node x_1 on this path is 0-node, and all other nodes are 1-nodes.

Property 1. If a test vector X^t activates in SSBDD a 0-path (1-path), then only 0-nodes (1-nodes) have to be considered as *candidate fault sites* [32].

The Property 1 can be taken into account to speed-up fault simulation. According to Property 1, the analysis of the 1-path in Fig.2 shows us that all the nodes, except x_1, may be qualified as candidate fault sites. However, further analysis is needed to confirm which of the candidate nodes are in fact detectable by the pattern. Since the faults at all 1-nodes for X^t (in Fig.2), will cause the direction change during graph traversing, then the faults at all 1-nodes are detectable by X^t.

Example 8. In the path $L(x_1, \#1)$ activated by the test in Fig2, according to Property 1, the nodes x_{22}, x_3, $\neg x_7$, and x_{81} are the candidates of fault sites. By additional simulation – by inverting the values of these variables, and by tracing the related paths $L(x_{22}, \#0)$, $L(x_3, \#0)$, $L(\neg x_7, \#0)$, $L(x_{81}, \#0)$ for each of these nodes, we can find that the test pattern in Fig.2 detects the faults: $x_{22} \equiv 0$, $x_3 \equiv 0$, $\neg x_7 \equiv 0$, and $x_{81} \equiv 0$, respectively.

Theorem 2. The faults at two connected SSBDD nodes a and b are *equivalent* iff the following two conditions are satisfied: (1) the nodes have the same neighbor c, and (2) the node b has a single incoming edge from a.

Proof. The first condition refers to the fact that both nodes can be tested by the same test pattern which activates the paths $L(\text{Root},a)$, $L(a,\#e)$ where $e \in \{0,1\}$, and the path $L(c,\#(\neg e))$. The second condition refers to that this test pattern is the only one which can test both of the node faults a/$\neg e$ and b/$\neg e$.

Example 9. For example the faults $x_{22}/0$ and $x_3/0$ are equivalent, because the related nodes x_{22} and x_3 have the same neighbor node x_4, and a single entry edge into x_{22}, hence, one of these faults can be collapsed. In a similar way, using Theorem 2, it is easy to find in the SSBDD in Fig.1 other equivalent faults: $x_1/0 \equiv x_{21}/0$, $x_5/0 \equiv x_{61}/0$ (or $\neg x_5/1 \equiv \neg x_{61}/1$, according to the notation in the SSBDD), $x_{8,2}/0 \equiv x_9/0$, and $x_{72}/1 \equiv x_{62}/0$. On the other hand, the faults $\neg x_{71}/0$ and $x_{81}/0$ are not equivalent. Despite of having the same neighbor x_{82}, the node $\neg x_{71}$ has three entry edges, and the single entry requirement of Theorem 2 is not satisfied.

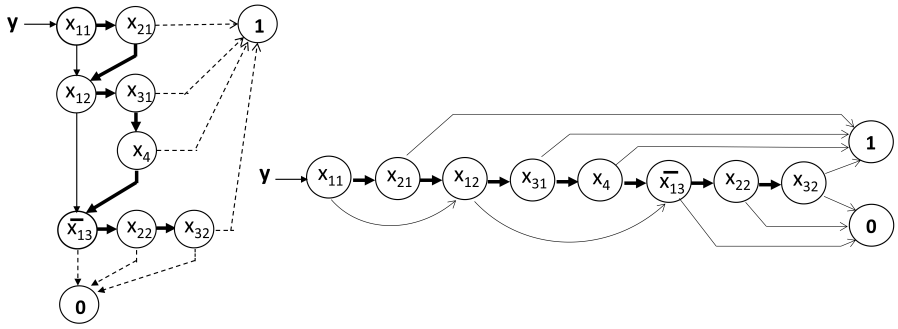

Fig. 4. Hamiltonian path in two presentations of the same SSBDDs

Property 2. SSBDDs have always a single *Hamiltonian path* that visits all the nodes (except #0 and #1), and which determines a unique ranking of the nodes. The nodes a and b are in the relationship $a < b$ if the node a will be traversed before b along the Hamiltonian path [32].

Fig.4. depicts an example of two possible presentations of the same SSBDD which represents the following Boolean expression:

$$y = x_{11}x_{21} \vee x_{12}(x_{31} \vee x_4) \vee \overline{x_{13}}\, x_{22}x_{32}$$

Theorem 3. The fault $b/0$ *dominates* $a/0$ (or $b/1$ *dominates* $a/1$), iff the following conditions are satisfied: (1) there exists a single 1-path (or a single 0-path) through the nodes for detecting both of these faults, (2) $a < b$, and (3) the node b has more than 1 incoming edges. **Proof.** The first condition demands that these faults can be detected by a single test pattern (the condition of the equivalency). The second condition demands that there will be no other path for testing a and not testing b. The third condition is needed to give the possibility to test b and not to test a. From satisfying these conditions, it follows that any test for a must detect the related fault as well at b. Hence, the fault at a is dominated by b. If the third condition is not fulfilled, the related node faults at the nodes a and b are equivalent.

Example 10. In Fig.1, the faults $\neg x_{71}/0$ and $x_{81}/0$ dominate $x_{22}/0$, and, according to Theorem 3, can be collapsed. Based on this result and taking into account Example 7, we can collapse 3 faults on the activated path $L(x_1,\#1)$: $x_3/0$, $\neg x_{71}/0$ and $x_{81}/0$.

Corollary 3. The fault $a/0$ dominates $b/0$ ($a/1$ dominates $b/1$) iff the following conditions are satisfied: (1) there exists a single 1-path (0-path) through the nodes for detecting both of these faults, (2) $a < b$, and (3) the node a can be tested by activating another path where b is not tested.

Proof. The proof results directly from Theorem 3 after transforming the SSBDD, so that the ranking of nodes a and b involved in the dominance relation will be swapped (see Fig.5 and Example 10).

Example 11. In Fig.5 two different SSBDDs are shown which represent the same digital circuit, and correspond to the following two Boolean expressions:

$$y = x_{11}x_{21} \vee x_{12}(x_{31} \vee x_4) = x_{11}x_{21} \vee (x_{31} \vee x_4)x_{12}$$

The graphs represent the following rankings R1: $x_{12} < x_{31} < x_4$ and R2: $x_{31} < x_4 < x_{12}$, respectively, according to the Hamiltonian paths in the SSBDDs. In the SSBDD with node ranking R2, we determine that the node x_{12} dominates both, x_{31} and x_4, according to Theorem 3, and the same result we get for the SSBDD with node ranking R1, according to Corollary 3.

Using Theorems 2, 3 and Corollary 3 may directly lead to a simple algorithm of fault collapsing by systematic pairwise analysis of the equivalence and dominance relationships. However, in the worst case, such a pairwise analysis may lead to a quadratic complexity of SSBDD tracing.

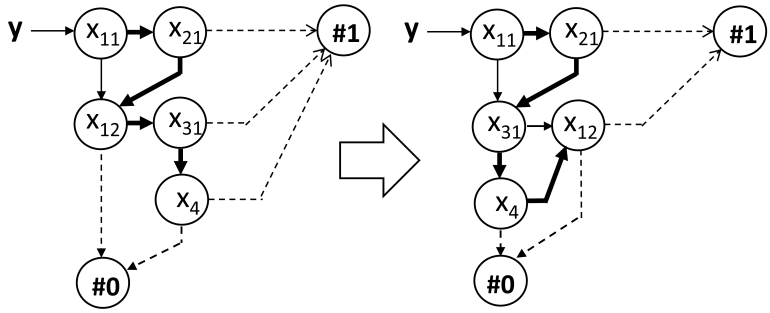

Fig. 5. Transformation of SSBDDs by swapping the nodes or subgraphs

On the other hand, taking into account the possibility of mapping sub-graphs in the SSBDD into the sub-circuits of the gate network, it would be possible to develop an algorithm of fault collapsing which will use only a single trace through the SSBDD with local analysis of proximate node pairs, and which would provide linear complexity of the algorithm.

5 Fault Equivalence and Fault Dominance Fast Reasoning on the SSBDD Model

From the definition of the SSBDDs [32] we can derive the following rules for recognition of gates and sub-circuits in the SSBDD model, which will help us to develop a fault collapsing algorithm on SSBDDs with linear complexity.

Definition 7. Let us call the consecutive nodes on the *Hamiltonian path* of SSBDD as a *group* if they all have the same neighbor node, and all these nodes except the first one have a single incoming edge.

Example 12. Consider a circuit and its SSBDD model in Fig.1. The two consecutive nodes x_{22} and x_3, and the nodes $\neg x_5$ and $\neg x_{61}$ form two groups in the SSBDD in Fig.3. No more groups exist in this graph. The nodes $\neg x_{61}$ and $\neg x_{71}$ don't form a group.

Rule 1. A group of two nodes connected by horizontal edges (vertical edges) represents AND (OR) gate, and due to the fault equivalence, a fault at one of the inputs

can be collapsed. The Rule 1 results directly from the method of synthesis SSBDDs by superposition of BDDs of gates [23].

Example 13. The nodes x_{22} and x_3 in the SSBDD in Fig.6 represent AND gate, and $\neg x_5$ with $\neg x_{61}$ represent OR gate. These gates can be recognized in the circuit. According to Rule 1, the faults $x_{22}/0$ (or $x_3/0$) and $\neg x_5/1$ (or $\neg x_{61}/1$) can be collapsed.

Rule 2. If a node b in SSBDD has at least two or more incoming edges, it represents a path to a gate G where all the paths, represented by a subset of nodes $S(b) = \{a \mid a < b\}$, are joining. The fault of b dominates over the related faults of the nodes $a \in S(b)$, since the conditions of Theorem 3 are satisfied.

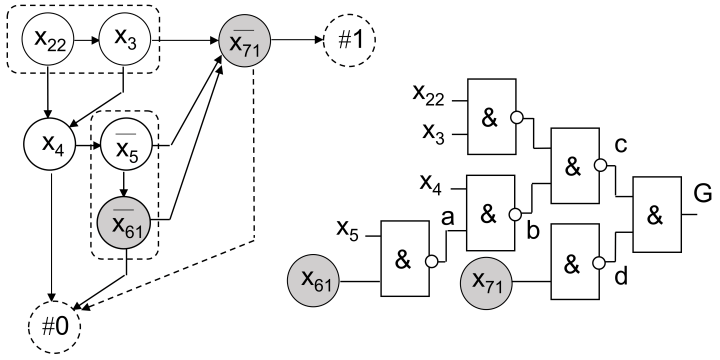

Fig. 6. Mapping SSBDD subgraphs into the circuit

Example 14. The node $\neg x_{71}$ in SSBDD in Fig.6 has three incoming edges. It represents a path to the gate G joining with the paths represented by all other nodes a, $a < \neg x_{71}$, in this SSBDD. The nodes $\neg x_{61}$ and$\neg x_{71}$ don't form a group according to Definition 7 and don't represent AND.

The Rules 1 and 2 help to understand, how the fault equivalence and dominance relations in SSBDDs can be related to the similar equivalence and dominance relations in the gate-level circuit. If we have recognized a gate in SSBDD, the equivalence relations overlap for SSBDD and the circuit. The dominance relation in an SSBDD for a node with several incoming edges can be explained by transitive closure of dominance relations. For example, the dominance $\neg x_{71}/0 \rightarrow \neg x_{61}/0$ (or $x_{71}/1 \rightarrow x_{61}/1$ in the circuit) in the SSBDD in Fig.6 can be explained by the following transitive closures in the circuit: $x_{71}/0 \equiv$ d/0 $\equiv$ c/0, and c/0 $\rightarrow$ b/1 $\rightarrow$ a/0 $\rightarrow x_{61}/1$, from which $x_{71}/1 \rightarrow x_{61}/1$ results ($x_{71}/1$ dominates $x_{61}/1$).

Algorithm 1 presents a procedure for fault collapsing in circuits which are represented by the SSBDD model. The algorithm is based on pairwise checking of Rule 1 (for equivalence) and Rule 2 (for dominance) by traversing along the Hamiltonian path in SSBDD. The algorithm has linear complexity.

Example 15. Consider the fault collapsing in the SSBDD in Fig.1 according to Algorithm 1. The SSBDD represents the FFR region of the circuit in Fig.1 where the

fan-out inputs are not included. The initial number of the gate level SAF faults in the FFR in Fig.1 is 52 (2 faults per each of 26 lines). By synthesizing the SSBDD for the FFR of the circuit according to Procedure 1 we reduce the number of representative faults from all 52 faults to 28 faults (2 faults per each of 14 nodes in the SSBDD). By using Algorithm 1, we further collapse 10 faults ($x_1/0$, $x_{22}/0$, $x_4/0$, $x_5/0$, $x_{71}/1$, $x_{81}/0$, $x_{82}/0$, $x_9/0$, $x_{72}/1$, $x_{10}/1$) which results in the total number of remaining 18 representative faults, i.e. 3 times reduction compared to the initial number of faults.

Table 2 shows which node faults in the SSBDD in Fig.1 are collapsed on the basis of equivalence relation and which faults on the basis of dominance relation.

ALGORITHM 1: Fault collapsing on SSBDDs

Input: SSBDD model for a given circuit
Output: Set of collapsed faults C
Notations:
M – the number of all nodes
m – the number of the first node of the current node pair
n – the number of the second node, $n = m + 1$, $n^* = n+1$
$d(m)$ – direction from node m to n
$n(d)$ – the neighbor of the node n in direction $d(m)$
$C(m)$ – the type of the collapsed fault $C(m) \in \{0,1\}$
$IN(m)$ / $OUT(m)$ – directions of incoming/outgoing edges
$FI(m)$ – flag to remember the multiple fan-in for m
$D(m)$ – flag to remember the direction of multiple fan-in

```
1:  for all SSBDDs in the model
2:    for all nodes m in the current SSBDD
3:      if m < M then go to 6 end if
4:      if m = M then C(m) = ¬IN(m) end if
5:      go to 28
6:      if m(¬d) = n(¬d) then                        (checking of Rule 1)
7:        if FI(n) ≠ 1 then                          (checking of Rule 2)
8:          if [(IN(m)>1) & (IN(m) = OUT(m))] or D(m) = 1 then
9:            D(n) = 1
            end if
10:         if n = M then
11:           if IN(m)>1 or D(m) = 1 then C(n) = ¬d(m), go to 13 end if
12:           if OUT(m) ≠ OUT (n) then               (checking of Rule 1)
13:             C(m) = ¬d(m)
14:             if n(¬d) ≠ ∅ then FI(n(¬d)) = 1 end if
15:             m = m + 2, go to 3
              else
16:             if FI(n*) = 1 then go to 13 end if
17:             if n(¬d) = n*(¬d) then               (checking of Rule 1)
18:               C(m) = ¬d(m), m = m + 1, go to 3
19:             else C(n) = ¬d(m), go to 15 end if
              end if
            end if
20:       else go to 23 end if
21:     else if FI(m) = 1 then                       (checking of Rule 2)
22:           if FI(n) = 1 then                      (checking of Rule 2)
23:             C(m) = ¬IN(m)
24:             if m(¬d) ≠ ∅ then FI(m(¬d)) = 1 end if
25:             m = m + 1, go to 3
26:           else C(m) = ¬d(m), go to 24 end if
27:         else go to 26 end if
28:     end if
29:   end for
30: end for
```

Table 2. Fault collapsing results for the SSBDD in Fig.1

Node	Collapsed fault	Comments
x_1	SAF $x_1/0$	Equivalent with $x_{21}/0$
x_{21}	No collapse	
x_{22}	SAF $x_{22}/0$	Equivalent with $x_3/0$
x_3	No collapse	
x_4	SAF $x_4/0$	Dominates $x_5/1$, $x_{61}/1$,
$\neg x_5$	SAF $x_5/0$	Equivalent with $x_{61}/0$
$\neg x_{61}$	No collapse	
$\neg x_{71}$	SAF $x_{71}/1$	Dominates $x_{22}/0$, $x_3/0$, $x_{22}/0$, $x_5/1$, $x_{61}/1$
x_{81}	SAF $x_{81}/0$	Dominates $x_{22}/0$, $x_3/0$, $x_{22}/0$, $x_5/1$, $x_{61}/1$, $x_1/0$, $x_{21}/0$, $x_{71}/1$
x_{82}	SAF $_{82}/0$	Equivalent with $x_9/0$
x_9	SAF $_9/0$	Dominates $x_{82}/0$
x_{72}	SAF $x_{72}/1$	Equivalent with $x_{62}/0$
$\neg x_{62}$	No collapse	
$\neg x_{10}$	SAF $x_{10}/1$	Dominates $x_{72}/0$, $x_{62}/1$, $x_9/0$, $x_{82}/0$

6 Lower and Higher Bounds for Fault Collapsing

Denote by N the number of all nodes in the SSBDD model of a circuit and by C the number of collapsed faults. The number of all SAF faults is $2N$, the number of representative faults after fault collapsing will be $R = 2N - C$, and the effect from fault collapsing can be expressed by the ratio $R/2N$.

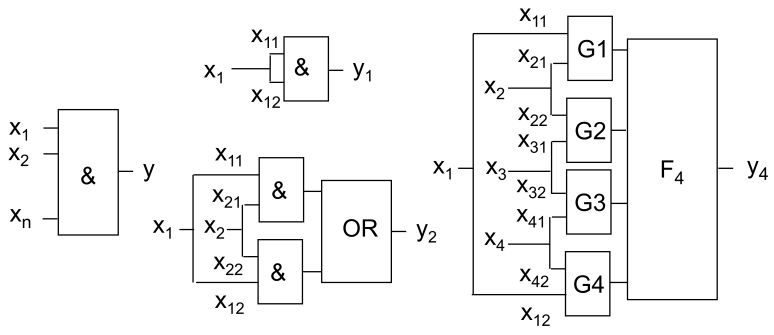

Fig. 7. Tree-like circuits with fan-out inputs of increasing complexity

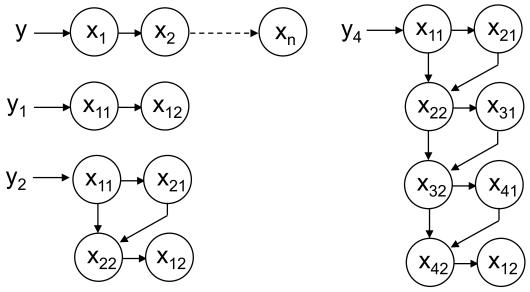

Fig. 8. SSBDD models for FFRs in the circuits in Fig.4

Theorem 4. The effect of fault collapsing in the SSBDD model of the given digital circuit will be always in the boundary $1/2 < R/2N \leq 5/6$. For a single FFR of any given digital circuit the effect of fault collapsing in the related single SSBDD will be always in the boundary $1/2 < R/2N \leq 3/4$.

Proof. Any tree-like circuit with N inputs can be represented by SSBDD with N nodes. Examples of such circuits, with fan-outs only in inputs, and the related SSBDDs for the FFRs are depicted in Fig.7 and Fig.8, respectively.

In the simplest tree, a single gate with N inputs (gate y in Fig.7 and SSBDD y in Fig.8), we can collapse N-1 faults. Hence, $R = 2N - (N - 1) = N + 1$. Any partitioning of the set of inputs for more than one gate in this tree will reduce the total C by one fault per added gate and, hence, increase R. When increasing N, the lower bound for $R/2N$ is:

$$\lim_{n\to\infty} \frac{R+n}{2N+2n} = \lim_{n\to\infty} \frac{N+n+1}{2N+2n} = \lim_{n\to\infty} \frac{n}{2n} = \frac{1}{2}$$

On the other hand, consider formally (neglecting the redundancy) a single-input logic gate y_1 in Fig. 7. The SSBDD model of the gate has N=3 nodes representing the fan-out stem with 2 branches. The SSBDD for only the FFR of this gate has 2 nodes. There are two equivalent faults at the gate inputs where one of them can be collapsed. Hence, the number of representative faults in this circuit will be R=2N-1=5, and $R/2N$ = 5/6. Similarly, the SSBDD for the FFR has only R=2N-1=3, and $R/2N$ = 3/4.

Consider now the tree-like circuit y_2 in Fig.7 with two 2-input gates and two fan-out nodes. The SSBDD model of the circuit has N=6 nodes. There are again two equivalent faults at the gate inputs where one of them can be collapsed. Hence, the number of representative faults after fault collapsing will be $R = 2N - 2 = 10$, and again we get $R/2N = 5/6$. Similarly, for the SSBDD representing only the FFR in this circuit, we get N = 4, R = $2N - 2 = 6$, and $R/2N = 3/4$.

The circuit y_4 in Fig.7 illustrates how we can generalize the series of two circuits y_1 and y_2 into a series of expanding circuits y_n , n = 1,2,3,4…, where each circuit will consist of an input sub-circuit IN_n as a chain of n 2-input gates, and a tree-like sub-

circuit F_n. In each such a circuit, the ratio $R/2N = 5/6$ remains constant. In IN_n for each gate, only a single fault can be collapsed resulting in total in n collapsed faults.

It is easy to realize that any structural change inside the sub-circuit F_n will not change the ratio $R/2N = 5/6$. The reason is that all the faults in F_n will dominate the faults in IN_n. On the other hand, by adding $n = 1, 2,\ldots$ non-fan-out inputs to the sub-circuit IN_n we get $R/2N^* = (R+n)/(2N+2n)$, and by adding n fan-out inputs with 2 branches to IN_n we will get $R/2N^{**} = (R+2n)/(2N+6n)$. Each addition of a fan-out branch is equivalent to the case of adding a single input node where no faults can be collapsed.

From above it follows that for the case of digital circuits as networks of FFRs the higher bound (the worst case of remaining representative faults) for the ratio $R/2N$ will be:

$$R/2N^{**} < R/2N^{*} < R/2N \leq 5/6$$

Hence, the range between lower and higher bounds for the ratio $R/2N$ characterizing the number of remaining representative faults after using Algorithm 1 for fault collapsing will be:

$$1/2 < R/2N \leq 5/6$$

Extending the same analysis for a single sub-circuit as an FFR, we will have the range between lower and higher bounds for the ratio $R/2N$ as:

$$1/2 < R/2N \leq 3/4$$

Corollary 4. From Theorem 4, it directly follows that for the SSBDD model of the given digital circuit with N nodes in the model, the number of collapsed faults $C = 2N - R$ belong will always to the interval $N/3 \leq C < N$. Hence, $N/3$ will serve as the lower bound for the number of collapsed faults achievable in the SSBDD model.

Corollary 5. From Theorem 4, it directly follows that for the SSBDD with N nodes which represents any FFR in digital circuits, the number of collapsed faults $C = 2N - R$ will always belong to the interval $N/2 \leq C < N$. Hence, $N/2$ will serve as the lower bound for the number of collapsed faults achievable in any single SSBDD created for the given FFR.

Example 16. Consider again the digital circuit and the FFR of the circuit in Fig.1. In Example 15 we found the number of collapsed faults $C = 10$, and the numbers of remaining representative faults for the case of FFR $R_{FFR} = 18$, and for the full circuits (with the SSBDD for FFR and additional 4 single-node SSBDDs for 4 inputs with fan-outs) $R_{circuit} = 26$. For FFR we get $N/2 = 7 \leq \mathbf{C = 10} < N = 14$, whereas for the full model we have $N/3 = 6 \leq \mathbf{C = 10} < N = 18$.

The result shows that the interval between lower and upper bounds for a single FFR is smaller compared to the interval for whole circuit. More discussion in that topic follows in the experimental part of the paper.

7 Experimental Data

The fault collapsing experiments were carried out with Intel Core i5 3570 Quad Core 3.4 GHz, 8 GB RAM, using ISCAS'85, ISCAS'89 and ITC'99 benchmark circuits. The experimental results are presented in Tables 3 and 4.

Table 3. Comparison with other methods

Circuit	# Faults	Fault set size					CPU time, s	
		[2]	[14]	[34]	[18]	New	[18]	New
c1355	2710	1234	1210	1100	808	1210	46	0.003
c1908	3816	1568	1566	1286	753	1243	14	0.008
c2670	5340	2324	2317	2046	1853	1989	110	0.009
c3540	7080	2882	2786	2584	2092	2340	831	0.010
c5315	10630	4530	4492	4404	3443	3900	72	0.012
c6288	12576	5840	5824	4832	5824	5824	4	0.019
c7552	15104	6163	6132	5480	4707	5156	232	0.016

Table 4. Fault collapsing for ISCAS'89 and ITC'99 circuits

Circ	# *Gates*	R^* [35,36]	$2N$	R (New)	$R/2N$ %	Gain R^*/R	Time s
s13207	24882	9815	10456	7933	75.9	1.24	0.04
s15850	29682	11727	12150	9178	75.5	1.28	0.04
s35932	65248	39094	39094	29797	76.2	1.31	0.26
s38417	69662	31180	32320	25162	77.9	1.24	0.20
s38584	72346	36305	38358	28016	73.0	1.30	0.18
b15	47414	21072	23498	17439	74.2	1.21	0.04
b17	154220	68037	81330	60684	74.6	1.12	0.12
b18	463570	206736	277978	205866	74.1	1.00	0.42
b18_1	453088	202812	264244	196179	74.2	1.03	0.40
b19	1345442	533142	560704	415251	74.1	1.28	0.84
b19_1	1275720	507476	534184	396151	74.2	1.28	0.80
b21	79556	35994	48182	35169	73.0	1.02	0.08
b21_1	63732	29091	34510	25359	73.5	1.15	0.06
b22	113308	51277	70464	51511	73.1	1.00	0.11
b22_1	98006	44771	52172	38359	73.5	1.17	0.08
Aver	290392	121902	138643	102804	74.1	1.2	0.24

In Table 3, the sizes of fault sets after fault collapsing for the proposed method (New) with previous structural [2,14,34] and functional [18] methods are compared. The new proposed method has better results in fault collapsing than the previous structural methods. The functional method [18] is very slow and not scalable due to high computational cost of calculating transitive closures on dominance graphs

whereas the proposed method has a very high speed due to the linear complexity and is well scalable. As an example, the difference in time costs for c3540 and c6288 in case of [18] is 200 times whereas for the proposed method the difference is 2 times.

Due to different computing frameworks the speeds of the algorithms [18] and developed in this paper cannot be directly compared. On the other hand, the time cost needed for the first part of fault collapsing as a side effect of SSBDD synthesis was not included into the CPU time data in Table 3.

The experimental results for larger ISCAS'89 and ITC'99 circuits (R^* is the number of remaining faults after collapsing) are depicted in Table 4. The column R^*/R shows the gain (1.2 times in average) of the achieved fault collapse (in the column R(New)) compared to the results in [35,36]. The last column shows that Algorithm 1 has linear complexity, is well scalable and can be efficiently used for large circuits. The linear complexity of the method is explained by the fact that the fault equivalence and dominance reasoning is reduced only to the local pairwise analysis of the neighbor nodes during traversing the Hamiltonian path of the SSBDD. The number of pairs to be analyzed, as it results from Algorithm 1, is in the interval (N-1, N/2+1) where the lower bound refers to the extreme case of the logic gate with N inputs, and the higher bound refers to the extreme case of the two-level AND-OR (OR-AND) circuits with 2 inputs for the 1st level AND (OR) gates, plus one additional input for the 2-nd level gate.

Note, that according to Theorem 4, the higher bound for R/2N is 83% and the lower bound is 50%. The best result of fault collapsing – 73.0 % of remaining faults, the worst result – 77.9% and the average of 74.1% all fit well into the interval between the bounds. However the results are considerably closer to the higher bound of 83.3% of remaining representative faults than the lower bound of 50%.

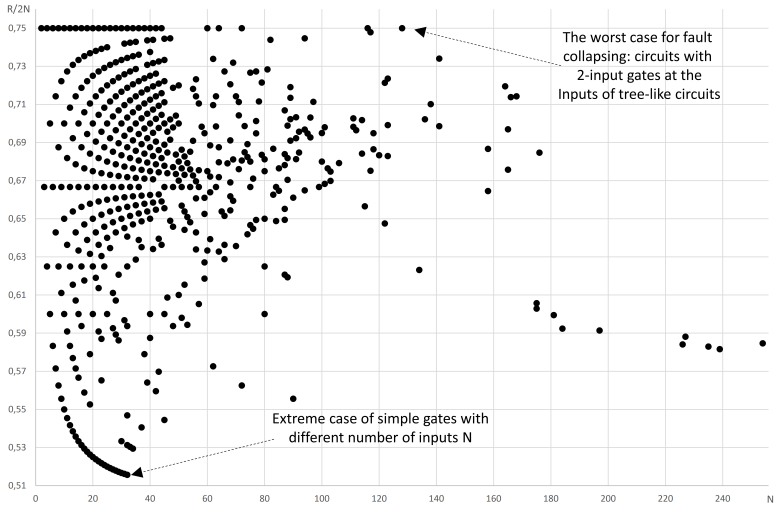

Fig. 9. Distribution of SSBDD cases with different characteristics (N, R/2N)

In Fig.9 we show statistical data collected from the fault collapse experiments with SSBDDs in 0.5 million tree-like sub-circuits (FFRs) in 111 different circuits of ISCAS'85, ISCAS'89 and ITC'99 families. Fig.6 presents a plot of different sub-circuit cases characterized by the number of nodes N in SSBDDs and the results of fault collapsing $R/2N$. Two extreme cases are highlighted: single-gate circuits (the best fault collapsing case) and the circuits with 2-input gates at the first level of tree-like circuits – the worst fault collapsing case where the higher bound for the remaining representative faults $R/2N = 3/4$ was reached, respectively the lower bound (the minimum number) of faults collapsing $C = N/2$.

8 Conclusions

In this paper we proposed a new structural fault collapsing method and an algorithm with linear complexity. The method is based on using SSBDD model for representing gate-level circuits as higher FFR-level networks. The synthesis of SSBDDs presents the first step of fault collapsing in FFR-s, and the resulting collapsed fault set can be regarded as a side-effect of SSBDD synthesis. We have introduced the concepts of fault equivalence and dominance relations between the faults on the SSBDD model, and present an algorithm for systematic fault collapsing in SSBDDs as a process of creating of the representative fault set defined at the higher level communication network of FFRs.

We developed the lower and higher bounds of the SSBDD based fault collapsing, and showed that the number of collapsed faults C in the SSBDD model of an arbitrary digital circuits belongs to the interval $N/3 \leq C < N$ where N is the number of nodes in the SSBDD model.

Experiments showed that the proposed method is more efficient than the previous structural fault collapsing methods and due to high scalability makes it very promising for large circuits.

ACKNOWLEDGMENTS

The work has been supported by EU FP7 STREP project BASTION, and HORIZON 2020 RIA project IMMORTAL.

References

1. G.M.L.Bushnell, V.D.Agrawal. Essentials of Electronic Testing. Springer, Boston, 2000.
2. K.To. Fault Folding for Irredundant and Redundant Combinational Circuits. IEEE Trans. on Comp, Vol. C-22, No. 11, 1973.
3. T.Niermann, J.H.Patel. HITEC: A Test Generation Package for Sequential Circuits. EDAC, pp.214-218, Feb. 1991.
4. T.P.Kelsey, K.K.Saluja, S.Y.Lee. An Efficient Algorithm for Sequential Circuit Test Generation. IEEE Trans. on Computers, Vol.42, No.11, pp.1361-1371, Nov. 1993.

5. Raja K. K. R. Sandireddy and Vishwani D. Agrawal. Using Hierarchy in Design Automation:The Fault Collapsing Problem. Proc. of the 11th VLSI Design and Test Symposium Kolkata, Aug. 8–11, 2007.
6. A.Veneris, R.Chang, M.S.Abadir, S.Seyedi, Functional Fault Equivalence and Diagnostic Test Generation in Combinational Logic Circuits Using Conventional ATPG. JETTA. 21(5):495–502, 2005.
7. A.Lioy. Advanced Fault Collapsing. IEEE Design and Test of Computers, Vol.9, No.1, pp. 64-71, Jan. 1992.
8. H.Al-Assad, R.Lee. Simulation Based Approximate Global Fault Collapsing. Proc. of Int. Conf. on VLSI, pp.72-77, 2002.
9. I.Pomeranz, M.Reddy. Level of Similarity: a Metric for Fault Collapsing. Proc. of DATE, pp.56-61, Feb. 2004.
10. I.Pomeranz, S.Reddy. Safe Fault Collapsing Based on Dominance Relations. Proc. of ETC, pp. 7-8, 2008.
11. R.Hahn, R.Krieger, B.Becker. A Hierarchical Approach to Fault Collapsing. Proc. of EDTC, pp.171-176, 1994.
12. A.V.S.S.Prasad, V.D.Agrawal, M.V.Atre. A New Algorithm for Global Fault Collapsing into Equivalence and Dominance Sets. Proc. of ITC, pp.391-397, Oct.2002.
13. R.Sethuram, M.L.Bushnell, V.D.Agrawal. Fault Nodes in Implication Graph for Equivalence Dominance Collapsing, and Identifying Untestable and Independent Faults. Proc. of VLSI Test Symposium, pp.329-335, 2008.
14. V.D.Agrawal, A.V.S.S.Prasad, M.V.Atre. Fault Collapsing via Functional Dominance. Int. Test Conf., pp.274-280, 2003
15. R.K.K.R.Sandireddy, V.D.Agrawal. Diagnostic and Detection Fault Collapsing for Multiple OUtput Circuits. Proc. of DATE, pp.1014-1019, 2005.
16. M.E.Amyeen, W.K.Fuchs, I.Pomeranz, V.Boppana. Fault Equivalent Identification in Combinational Circuits Using Implication and Evaluation Techniques. IEEE Trans. on CAD, Vol.22, No.7, pp.922-936, July 2003.
17. V.C.Vimjam, M.S.Hsiao. Efficient Fault Collapsing via Generalized Dominance Relations. Proc. of VLSI Test Symposium, pp.258-265, 2006.
18. R.Adapa, S.Tragoudas, M.K.Michael. Sub-Faults Identification for Collapsing in Diagnosis. Int. Conf. ISCAS, pp.815-818, 2006.
19. I.Pomeranz, S.Reddy. Equivalence and Dominance Relations Between Fault Pairs and Their Use in Fault Pair Collapsing for Fault Diagnosis. Int. Conf. on VLSI Design, pp. 1-6, 2007.
20. R.Adapa, S.Tragoudas, M.K.Michael. Accelerating Diagnosis via Dominance Relations between Sets of Faults. Proc. of the VLSI Test Symposium, pp.219-224, 2007.
21. I.Pomeranz, S.Reddy. Safe Fault Collapsing Based on Dominance Relations. Proc. of ETC, pp. 7-8, 2008.
22. R.Bryant. Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans on Comp, 1986, vol. C-35, 677-691.
23. R.Ubar. Test Synthesis with Alternative Graphs. IEEE Design & Test of Computers. Spring 1996, pp. 48-59.
24. R.Ubar, J.Raik, H.-T.Vierhaus. Design and Test Technology for Dependable Systems-on-Chip. IGI Global, 2011, 550 p
25. R.Ubar, S.Devadze, J.Raik, A.Jutman. Parallel X-Fault Simulation with Critical Path Tracing Technique. Proc. of DATE, 2010.
26. M.Gorev, R.Ubar, S.Devadze. Fault Simulation with Parallel Exact Critical Path Tracing in Multiple Core Environment. DATE, 2015.

27. C.Y. Lee. Representation of Switching Circuits by Binary Decision Diagrams. Bell System Techn. J., 1959, v.38, No7.
28. R.Ubar. Test Generation for Digital Circuits Using Alternative Graphs. Proc. Tallinn Technical University, 1976, No.409, Tallinn, pp.75-81.
29. S.Akers. Binary Decision Diagrams," IEEE Trans. on Comp., Vol.27, 1978, pp.509-516.
30. R.Ubar Multi-Valued Simulation of Digital Circuits with Structurally Synthesized BDDs. OPA, Gordon and Breach Publishers, Multiple Valued Logic, 1998, Vol.4.
31. A.Jutman, J.Raik, R.Ubar. SSBDDs: Advantageous Model and Efficient Algorithms for Digital Circuit Modeling, Simulation & Test. 5th Int. Workshop on Boolean Problems. Freiberg, Germany, Sept. 19-20, 2002, pp.157-166.
32. R.Ubar. Overview about Low-Level and High-Level Decision Diagrams for Diagnostic Modeling of Digital Systems. Facta Universitatis (Nis) Ser.: Elec. Energ. vol.24, no.3, Dec. 2011, 303-324.
33. D.Mironov, R.Ubar. Lower Bounds of the Size of Shared Structurally Synthesized BDDs. IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). Warsaw, April, 23-25, 2014, pp. 77-82.
34. R.Ubar, D.Mironov, J.Raik, A.Jutman. Structural Fault Collapsing by Superposition of BDDs for Test Generation in Digital Circuits. IEEE ISQED, 2010, San Jose, CA - USA, pp. 250-257.
35. F.Brglez et al. Combinational Profiles of Sequential Benchmark Circuits. ISCAS'89, 1989.
36. ITC'99. http://www.cad.polito.it/downloads/tools/itc99.html