computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

HIGH PERFORMANCE FAULT SIMULATION FOR DIGITAL SYSTEMS

Vladimir Hahanov, Gennadiy Krivoulya, Irina Hahanova, Olga Melnikova, Vladimir Obrizan

Professor, Ukraine, 61166, Kharkov, Lenin ave, 14,
E-mail: hahanov@kture.kharkov.ua

Abstract: *Fast backttraced deductive-parallel fault simulation method oriented on processing of complex digital devices containing hundreds of thousand equivalent gates is offered. Data structures and algorithms for method realization are described.*

Keywords: - *back traced simulation, ATPG, re-convergent fan-outs, fault analysis model, deductive, parallel simulation*

1. INTRODUCTION

The actuality of work is conditioned by necessity to increase considerably test generation speed for complex digital devices implemented in FPGA and ASIC. Automatic test generation systems of known firms(Cadence, Mentor Graphics, Synopsys, Logic Vision) are oriented on processing of chips, which contain at most hundred of thousand equivalent gates. The processing time is equal to several hours and is unacceptable if digital device contains millions of gates. Therefore, it is necessary to develop new approach to the problem, which enables to speed-up digital system analysis. For problem solving the new technology is used. It is the fast fault simulation method.

The testing object is a digital system, which can be implemented in chip. System should be described in VHDL in a form of Boolean equations. The purpose of investigation is to develop fault simulation method for quality assessment of tests generated for digital systems, which contains millions of gates and can be implemented in chips. Test is an input sequence, which detects all stuck-at-faults of the digital system.

The research tasks are:

1.The development of generalized model of digital circuit deductive-parallel analysis on the basis of superposition procedure.

2.The development of algorithms for structure functional analysis of the digital system. The purpose of analysis is to detect the set of reconvergent fan-outs.

3. The development of internal interpretative-compilative model of digital system. The model is used for effective fault-free and fault analysis of gates.

4. Algorithmic realization of the method. It is based on reconfiguration of digital device model during fault simulation. The purpose of reconfiguration is to decrease greatly the test assessment time.

Cubic fault simulation [1-3], deductive model of fault transportation [4,5], parallel method of fault lists processing [4,6], algorithm of primitives backtracing [7] form the basis of BDP(Backtraced Deductive-Parallel) fault simulation method.

2. MODEL OF DEDUCTIVE-PARALLEL FAULT ANALYSIS

The model of deductive-parallel analysis allows detecting all faults, which can be detected on the test-vector during the one iteration of circuit processing. The model is based on solving of following equation: [1]:

$$T \oplus F = L, \qquad (1)$$

where $F=(F_1,F_2,...,F_i,...F_n)$ – model of fault-free circuit behavior; n-number of lines; $T=(T_1,T_2,...,T_t,...,T_k)$– test, which consists of k binary patterns; $L=(L_1,L_2,...,L_t,...,L_k)$– set of deductive functions (DF) for parallel fault simulation on test T. The functions correspond to fault-free model F; $\oplus$ - is an exclusive OR(XOR) operation. . Test is an input sequence, which detects all stuck-at-faults. $F_i \in F$ is a Boolean function, which is used to define the value of line i

$$F_i = f_i(X_{i1}, X_{i2}, ..., X_{ij}, ...X_{in_i}). \quad (2)$$

Coordinate $T_{ti} \in T_t$ is a result of function f_i. $T_{ti} = F_i$ on test-vector t. Test is matrix of fault-free behavior of digital system:

$$T = [T_{ti}] = (T_{t1}, T_{t2}, ..., T_{ti}, ..., T_{tn}). \quad (3)$$

The equation (1) for obtaining DF for $T_t \in T$ is following: $L_t = T_t \oplus F$. If digital circuit is described with components, which define the values of all lines, the following equation can be used to transform fault-free primitive model into deductive function :

$$L_{ti} = T_t \oplus F_i = f_i[(X_{i1} \oplus T_{t1}), (X_{i2} \oplus T_{t2}), ... \\ .., (X_{ij} \oplus T_{tj}), ..., (X_{in_i} \oplus T_{tn_i})] \oplus T_{ti}, \quad (4)$$

Practical realization of expression (4) includes following steps:

1. Forming of interpretative models of digital system $W = \{F, L_0\}$. Initialization of test-vector number t=0. Initialization of vectors of fault, which are detected by

test $T = [T_{ti}]$: $\overset{n}{\underset{i=1}{\forall}}(D_i^0 = 0; D_i^1 = 0)$.

2. Definition of next input test pattern t=t+1 for $T_t \in T$. If there is no more input patterns (t > k), the simulation is finished.
3. Fault-free simulation of all primitives $F_i (i = \overline{1, n})$ of digital circuit on input test pattern $T_t^X \in T_t$ using model $F \in W$. The purpose of simulation is to define the values of internal and output lines $T_t^{\overline{X}} \in T_t$:

$$T_t^{\overline{X}} = f(T_t^X, F). \quad (5)$$

If the values of all lines, obtained in two contiguous iterations are equal $T_t^r = T_t^{r-1}$ then the step 4 can be realized .

The analysis of input vectors in two contiguous iterations: (T_{t-1}, T_t) is used for processing of sequential circuits Primitive $F_i (i = \overline{1, n})$ should be simulated if there are any differences in input test vectors $[T_{t-1}^X (F_i) \neq T_t^X (F_i)]$.

4. Initialization of the matrix of faults, which are detected by applied test-vector: $M = [M_{ij}]$, according to the expression:

$$[M_{ij}]\Big|_{(i,j=\overline{1,n})} = \begin{cases} 0 \Leftarrow (i \neq j); \\ 1 \Leftarrow (i = j). \end{cases} \quad (6)$$

Initialization of vectors of faults which can be detected on test -vector $\overset{n}{\underset{i=1}{\forall}}(S_i^0 = 0; S_i^1 = 0)$.

Reconfiguration of good interpretative primitive models $L_i (i = \overline{1, n})$, $L_i \in W$ by applying formula (4) for current test vector with defined good lines values:

$$T_t = (T_{t1}, T_{t2}, ..., T_{ti}, ..., T_{tn})$$

for achieving of modification $L_{ti} = T_t \oplus F_i$.

5. Forming of values of internal and output lines of matrix M by parallel simulation of primitives: $L_{ti} \in L_t$.
6. Forming of combined vector of faults S by formula:

$$S = \underset{\forall i \in Y}{\vee} M_i^r \quad (7)$$

This formula is applied to all matrix rows, which correspond to the primary output of the circuit.

7. When the condition $\overset{n}{\underset{i=1}{\forall}}(S_i = S_i^0 \vee S_i^1)$ is true, the evaluation of test pattern quality is executed by formula:

$$Q(T_t) = \frac{1}{2n}[\sum_{i=1}^{n}(S_i^0 + S_i^1)]. \quad (8)$$

then the transition to the next step is fulfilled, otherwise – pair $\{S^0, S^1\}$ is formed by equation

$$S^0 = S \wedge T, (S_i^0 = S_i \wedge T_i); \; S^1 = S \wedge \overline{T}, (S_i^1 = S_i \wedge \overline{T}_i) \quad (9)$$

If detected faults disappear from vector S = $\{S^0, S^1\}$: $\overset{n}{\underset{i=1}{\exists}}[(S_i = 0) \& (S_i^0 \vee S_i^1 = 1)]$, than it has to eliminate such defects from process of simulation by the rule:

$$(S_i^0 = S_i^1 = 0) \Leftarrow \overset{n}{\underset{i=1}{\forall}}[(S_i = 0) \& (S_i^0 \vee S_i^1 = 1)].$$

Fulfill transition to step 5.

8. Forming detectable fault vectors according to expression

$$D^0 = D^0 \vee S^0, \; D^1 = D^1 \vee S^1 \quad (10)$$

and evaluation of test quality by formula:

$$Q(T)=\frac{1}{2n}[\sum_{i=1}^{n}(D_i^0+D_i^1)]. \qquad (11)$$

Fulfill transition to step 2.

Represented algorithmic realization is oriented both on table description of complex primitives of RTL level and gate description of digital systems. Processing speed is invariant to the model type. Interpretative realization is more practically feasible from the view of program realization.

3. INTERPRETATIVE MODEL OF FAULT-FREE BEHAVIOR ANALYSIS

Structural model of primitive element good logic simulation for digital device is presented on fig. 1. The question about procedure of test vector T(Y) evaluation for definition of coordinate, which corresponds to output Y of logic element F is observed here. Its input values are represented by vector: $X=(X_1,X_2,...,X_k)$, and F_T is truth table of Boolean functions set, defined on concatenation of two vectors of binary variables $(F*T_X)$:

$$T(Y)=F_T(F*T_X)=F_T(F*T(X_1)*T(X_2)*...*T(X_k)).$$

Otherwise, for definition of coordinate T(Y) it is necessary to form binary vector of input variable values T_X on the basis of vector of line numbers X.

Then it is necessary to make concatenation of obtained vector with binary code of primitive kind F for $(F*T_X)$ row obtaining of truth table F_F, where in column Y, corresponding to the meaning of function, required value of coordinate T(Y) is present. Model of deductive-parallel fault analysis contains fault-free simulation structure that added by two modules (M,F_F). Analytic expression for calculating detected fault vectors, combined into matrix M with the helping of deductive function F_F which is transformed from F by expression (4), is represented by form:

$$M(Y)=F_F(F*T(X),M(X_1)\circ M(X_2)\circ...\circ M(X_k))=$$
$$=F_{(F*T(X))}(M(X_1)\circ M(X_2)\circ...\circ M(X_k)).$$

Here are operation, marked with the sign $\circ=\{\wedge,\vee\}$, may be presented by disjunction or conjunction; $F_{(F*T(X))}$ – deductive element, defined by binary address word $(F*T(X))$. For definition of vector-row M(Y) state it is necessary to find address (type) of deductive compiler-driven function, by using obtained concatenation of binary sequences $(F*T(X))$ during the fault-free simulation process. Input variables for element $F_{(F*T(X))}$ are register ones, where their theoretical length is equal to the number of lines in digital device. Then sequential execution of (k-1) register operations for all input vectors $M(X_i)\in M$ is realized. Result as a formed row M(Y) is written into matrix M. Vector-variable X_i can have inversion sign. In this case before execution of operations $\circ=\{\wedge,\vee\}$, inversion of register (variable) is performed.

$$M(\overline{X_i})=\overline{M}(X_i).$$

As an example let's consider the general truth table of four good behavior functions which contain also information for selection of binary address word of deductive compiler-driven functional element for fault analysis:

S	F	X_1X_2	Y	F_F
$\wedge$	00	00	0	00
	00	01	0	01
	00	10	0	10
	00	11	1	11
$\vee$	01	00	0	11
	01	01	1	10
	01	10	1	01
	01	11	1	00

S	F	X_1X_2	Y	F_F
$\overline{\wedge}$	10	00	1	00
	10	01	1	01
	10	10	1	10
	10	11	0	11
$\overline{\vee}$	11	00	1	11
	11	01	0	10
	11	10	0	01
	11	11	0	00

Here column F is identifier code of fault-free behavior function, (X_1,X_2) – set of binary input vectors of truth table for each of the four functions, Y – fault-free column value of function output, F_F – address word of deductive element compilative model that is represented by four primitives:

$$F_F=\begin{cases}00\rightarrow X_1\wedge X_2;\\01\rightarrow X_1\wedge \overline{X_2};\\10\rightarrow \overline{X_1}\wedge X_2;\\11\rightarrow X_1\vee X_2.\end{cases}$$

The computing complexity of digital circuit processing, which contains n two-input elements, is defined by following expression:

$$Q=[(2K+A)+A+(2n\tau)/W]=[2(K+A)+(2n\tau)/W]\times n,$$

where K – the bits concatenation time for obtaining address of function output value; A – the time of bit writing according to mentioned address; τ – the time of register operation (and, or, not) execution; W – the register length.

Because the first item 2(K+A) is unessential in comparison with the second one, computing complexity could be represented as following:

$$Q=(2n^2\tau)/W.$$

That is why the time expenses of digital circuit simulation are square proportional to the number of gates.

4. DEDUCTIVE METHOD OF STRUCTURAL ANALYSIS

Let's consider the current node (line) $V_j\in V$ for the oriented graph G of equipotent lines. The current

node $V_j \in V$ is so called original which can have corresponding predecessor nodes called image: $f^{-1}(V_j)$. The relations between mentioned subsets of the nodes are presented by formula:

$$V^i = f^{-1}(V_j), V^i \subseteq V, \qquad (12)$$

where $V^i \subseteq V$ – is subset of firsthand-predecessors for V_j.

The node model under consideration is a logical element OR, where the number of inputs is equal to its input edges. For example, if the or-graph contains 9 nodes (fig. 1) then it has 9 logical elements where edges are identified with input lines of primitive element OR.

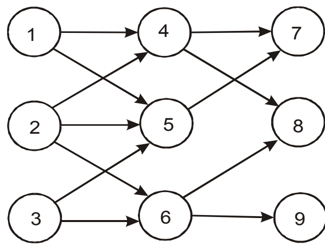


Fig. 1 – Re-convergent fan-outs graph.

Statement 1. Each node of feedback free graph has unique union of original and superposition of its images $[f_i^{-1}(V_j) \cup V_j]$ that so called extra-original.

Statement 2. All nodes in graph belonging to feedback loop have the equivalent extra-originals.

This statement is true because there is the same contrareachability of each graph point that belong to the feedback loop from the node that is predecessor for any feedback loop point. There are the nodes predecessors for RFO that are observed in fan-in line like RFO. Such predecessors have the same extra-originals as RFO. Node that has the single path without fan-outs till RFO is an example of such line.

Statement3. For removing all predecessors of RFO including in set V^j for node V_j and which are not RFO it is necessary and enough to deduct the intersections union of all pair of combination $C^2_{n_j}$ from set V^j.

The procedure of deductive analysis: for oriented graph that has not the global feedback loops the strategy it is necessary and enough to deduct the intersections union of all pair of combination $C^2_{n_j}$ from set V^j.

The procedure of deductive analysis: for oriented graph that has not the global feedback loops the strategy of RFO searching consists of the one pass around all nodes. This procedure based on the reach ability of deductive processing each of nodes includes three operations:

$$
\begin{aligned}
&1)\ V^j = \bigcup_{C^2_{n_j}} [f_p^{-1}(V_j) \bigcap_{\substack{p=i \\ q=\overline{i+1,n_j} \\ i=\overline{1,n_j-1}}} f_q^{-1}(V_j)];\\
&2)\ V^j = V^j \setminus \bigcup_{C^2_{m_j}} [V_j^p \bigcap_{\substack{p=i \\ q=\overline{i+1,m_j} \\ i=\overline{1,m_j-1}}} V_j^q];\\
&3)\ V_j = [\bigcup_{i=1}^{n_j} f_i^{-1}(V_j) \cup V_j] \setminus V^j,
\end{aligned}
\qquad (13)
$$

where $f_i^{-1}(V_j)$ – the element of original $f^{-1}(V_j)$, the total number of which for node V_j is equal to n_j.

The first equation is intended for RFO definition based on operation of union of all original combination $C^2_{n_j}$ pair intersection of considered node. The second one is intended for excluding from the list of RFO the nodes that are not RFO in according to the statement 3. The third one is intended for forming of predecessor list for each node. For the graph example (fig. 3) the sequential processing of all nodes by rules (13) is represented the following result of calculation:

$V^1 = \varnothing;\ V_1 = \{1\};\ V^2 = \varnothing;\ V_2 = \{2\};\ V^3 = \varnothing;\ V_3 = \{3\};$

$V^4 = (V_1 \cap V_2) = \varnothing;\ V_4 = \{1,2,4\};$

$V^5 = (V_1 \cap V_2) \cup (V_1 \cap V_3) \cup (V_2 \cap V_3) = \varnothing;\ V_5 = \{1,2,3,5\};$

$V^6 = (V_2 \cap V_3) = \varnothing;\ V_6 = \{2,3,6\};$

$V^7 = (V_4 \cap V_5) = \{1,2\};\ V_7 = \{1,2,3,4,5,7\} \setminus \{1,2\} = \{3,4,5,7\};$

$V^8 = (V_4 \cap V_6) = \{2\};\ V_8 = \{1,2,3,4,6,8\} \setminus \{2\} = \{1,3,4,6,8\};$

$V^9 = \varnothing;\ V_9 = \{2,3,6,9\}.$

The union of all subsets

$$V^{RC} = \bigcup_{j=1}^{n} V^j$$

represents possibility to obtain the full set of RFO (V^{RC}) of digital structure graph (see fig. 3):

$$V^R = V^1 \cup V^2 \cup V^3 \cup V^4 \cup V^5 \cup V^6 \cup V^7 \cup V^8 \cup V^9 =$$
$$= \varnothing \cup \varnothing \cup \varnothing \cup \varnothing \cup \varnothing \cup \varnothing \cup \{1,2\} \cup \{2\} \cup \varnothing = 1,2\}.$$

Because the lines defined as RFO are redundant for the continuing structural analysis, these lines can be excluded from the original list for each node. However the defined set of RFO after evaluation of every node has to be added to separate subset V^{RC}. Taking into account this statement it is necessary to modify equation 1 in (13). Such transformation leads to following equations:

$$1) V^j = \bigcup_{C^2_{n_j}} [f_p^{-1}(V_j) \bigcap_{i=1,n_j-1}^{\substack{p=i \\ q=\overline{i+1,n_j}}} f_q^{-1}(V_j)];$$

$$2) V^j = V^j \setminus \bigcup_{C^2_{m_j}} [V_j^p \bigcap_{i=1,m_j-1}^{\substack{p=i \\ q=\overline{i+1,m_j}}} V_j^q];$$

$$3) V^{RC} = V^{RC} \bigcup V^j;$$

$$4) V_j = [\bigcup_{i=1}^{n_j} f_i^{-1}(V_j) \cup V_j] \setminus V^{RC}. \qquad (14)$$

The differences between procedures (14), (13) are in RFO accumulation into set V^{RC} and its deduction from the list of predecessors for each node. Equation (14) is used for reduction of original line sub-sets for processed node.

5. DEDUCTIVE-PARALLEL ANALYSIS FOR GRAPH STRUCTURE

The vector $V_j \in V$ is as identifier of node V_j. The value j is position of vector coordinate that is equal to 1. Then the set of vectors (set of nodes in graph) is represented by unitary matrix:

$$V = \left\| V_{ij} \right\|_{(i,j=\overline{1,n})}; V_{ij} = \begin{cases} 1 \leftarrow i = j; \\ 0 \leftarrow i \neq j. \end{cases}$$

Because of isomorphism of Boolean- and set theory algebra, formula (13) can be modified to the algebraic-logical form:

$$1) V^j = \bigvee_{C^2_{n_j}} [f_p^{-1}(V_j) \bigwedge_{i=1,n_j-1}^{\substack{p=i \\ q=\overline{i+1,n_j}}} f_q^{-1}(V_j)];$$

$$2) V^j = V^j \wedge [\bigcup_{C^2_{m_j}} (V_j^p \bigwedge_{i=1,m_j-1}^{\substack{p=i \\ q=\overline{i+1,m_j}}} V_j^q)]; \qquad (15)$$

$$3) V^{RC} = V^{RC} \vee V^j$$

$$4) V_j = [\bigvee_{i=1}^{n_j} f_i^{-1}(V_j) \vee V_j] \wedge \overline{V}^{RC}.$$

where the conjunction and disjunction operations are executed for the vectors of matrix V. The same transformation for (14) brings to the result:

$$1) V^j = \bigvee_{C^2_{n_j}} [f_p^{-1}(V_j) \bigwedge_{i=1,n_j-1}^{\substack{p=i \\ q=\overline{i+1,n_j}}} f_q^{-1}(V_j)];$$

$$2) V^j = V^j \wedge \neg [\bigcup_{C^2_{m_j}} (V_j^p \bigwedge_{i=1,m_j-1}^{\substack{p=i \\ q=\overline{i+1,m_j}}} V_j^q)]; \qquad (16)$$

$$3) V^{RC} = V^{RC} \vee V^j;$$

$$4) V_j = [\bigvee_{i=1}^{n_j} f_i^{-1}(V_j) \vee V_j] \wedge \overline{V}^{RC}.$$

The advantages of formulas (15) and (16) are:
1) The simultaneous operations on matrix rows allow us to speed-up RFO analysis procedure in 10-100 times. 2)Speed-up of RFO list generation due to the operation on vectors corresponding to the identifiers of nodes. There is the sequence of calculation as an example based on formula (16) application (see fig. 1):

1. Matrix V^0 initialization:

V^0	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

and initialization of the RFO vector by zero values.
2. During the sequential nodes processing given by graph structure the following matrix V^1 is obtained:

V^1	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0
4	1	1	0	1	0	0	0	0	0
5	1	1	1	0	1	0	0	0	0
6	0	1	1	0	0	1	0	0	0
7	0	0	1	1	1	0	1	0	0
8	1	0	1	1	0	1	0	1	0
9	0	1	1	0	0	1	0	0	1

3. Accumulation of RFO during the all nodes evaluation leads to the final result represented by vector V^{RC}:

$$V^{RC} = \left| \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} \right|,$$

where meaning 1 identifies on line belonging to RFO list.

6. GRAPH STRUCTURE ANALYSIS FOR SEQUENTIAL CIRCUITS

Feature of sequential circuit analysis by structure simulation method connected with obtaining of identical originals for each node, which belongs to global feedback loop. First of all, the talk is about synchronous devices that have only local feedbacks in comparison with asynchronous one having global feedbacks. Because of delay hazard the last mentioned digital devices are not appeared by modern designing. That is why the circuits with flip-flops (FF) and latches having only local feedbacks are considered below. As an example of FF with 3 feedbacks it is shown on fig. 2.

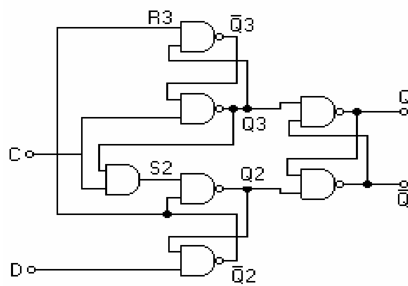


Fig. 2 – Gate level flip-flop model.

The graph model for FF contains 8 lines, 3 – RFO, 6 – lines including in local feedbacks (fig. 3).

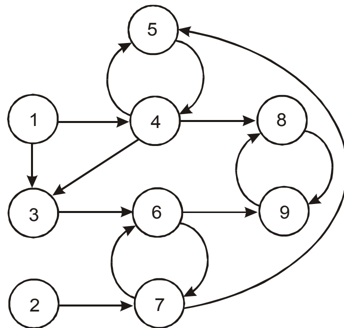


Fig. 3 – Graph model for FF. Nodes designation: 1 - C,2-D,3 - S2,4 - Q3, 5 - $\overline{Q3}$,6 -Q2,7 - $\overline{Q2}$,8 -Q,9 - $\overline{Q}$.

According to the statements 1,2 the criteria to recognize feedback from RFO is: existence of identical extra-originals for more then one node of digital device graph. The problem is in fact that the feedback nodes can be considered as RFO. There are two possible variants to solve this problem: 1) To add all feedback nodes to the set of RFO. 2) To mark feedback node as RFO if mentioned line has more then one outgoing arc. In the first case the number of RFO is increased, if the circuit contains a lot of FF. The second variant is more preferable because in the real digital structures consisting of FF this one decreases the RFO number via cutting of feedbacks. Thus there is the following definition for identification of RFO for digital circuits with feedbacks.

Definition: for the synchronous circuits containing FF feedbacks, the loop line is RFO if this one consists of more then one outgoing arcs and has the nearest line of convergence, which doesn't belong to this feedback loop.

Considering graph represented in fig. 3 it can be assumed that the RFO is the nodes $C, Q3, Q2$.

7. BDP FAULT SIMULATION METHOD

The offered interpretative-compilative model of deductive-parallel fault-free and fault analysis is a base for BDP-method and guarantees finding the correct solution in form of detected fault set on test-vector during the n^2 iterations. The main idea of speed-up is connected with RFO modification into primary pseudo-outputs for superposition procedure execution of tree-like structures and its un-processing in case of un-detectable of RFO presence. The program realization of BDP method consists of two basic components: 1) structure analyzer; 2) fault simulation on applied test-vector. Computational complexity of this procedure is $Q_r = n^2$, but this one is executed on the step of digital circuits pre-analysis and practically don't impact for the operation speed of test-vector simulation. For procedure of deductive parallel analysis the calculation of binary vector R of RFO lines using one of the considered methods of structural analysis is executed, where:

$$R = (R_1, R_2, ..., R_i, ..., R_n) ,$$

$$R_i = \begin{cases} 1 \leftarrow R_i \in R^{RC}; \\ 0 \leftarrow R_i \notin R^{RC}. \end{cases}$$

Test-vector simulation algorithm consists of:

1. Fault-free simulation of digital circuit. It is used to define reaction of all non input lines on applied test-vector $T_t \in T = [T_{ti}]$. All lines are divided into input, internal and output lines: (X,Y,Z). Test-row from matrix of fault-free behavior T presents as: $T_t = (T_t^X, T_t^Z, T_t^Y)$. Fault simulation vector $S = (S^X, S^Z, S^Y)$, is built every time for a new row T_t.

2. Fault simulation of RFO on applied test-vector. RFO reconfiguration into pseudo primary outputs. Vector initialization $S = (S_i^X = 0, S_i^Z = 0, S_i^Y = 1)$. If $S_i = 1$ then stuck-at-fault inversed to the fault-free value of line is detected.

Generation of initial list of RFO faults:

$$\{S_j\} = \begin{cases} j^{\overline{T}_{tj}} \leftarrow R_j = 1; \\ \varnothing \leftarrow R_j = 0, \end{cases}$$

where $\{S_j\} \subseteq S; j = \overline{1, n};$ S – fault list vector.

Fault simulation of RFO lines $L_R \subseteq L = (L_R, L_{\overline{R}})$, by deductive-parallel (deductive) method on reconfigurable model, which corresponds to the test-vector T_t. The method should be used if the number of RFO is non-significant (20%), because $L_R / L_{\overline{R}} << 1$.

3. Circuit model decomposition based on RFO fault simulation results. The tree-like sub-graphs with root node that is FRO with undetected fault, are excluded:

$$L_U = L_U \setminus L_R^0 \cup f^{-1}(L_R^0) .$$

4. Fault simulation of lines, complemented RFOs and sub-graphs with non detected root nodes till complete set $L_U = L_U \setminus [L_R^1 \cup L_R^0 \cup f^{-1}(L_R^0)]$.

The inputs fault analysis of every primitive is fulfilled via application deductive-parallel algorithm to the detected fault matrix of current element only, not circuit. Such analysis can be also executed deductively, by using input fault detected lists of primitive.

5. Superposition procedure for detected fault vectors on modified digital device model which is concluded in transformation of RFO into primary pseudo-outputs. The mentioned procedure is leaded to disjunction of input detected fault vector $S^i = (S_1^i, S_2^i, ..., S_j^i, ..., S_{n_i}^i)$ of i-primitive with fault simulation vector S on conditions that line, corresponding to the output of i-element, has 1-value ($s_i = 1$):

$$S(I_j^i) = S(I_j^i) \bigvee_{j=1}^{n_i} S_j^i \Leftarrow s_i = 1,$$

where $I^i = (I_1^i, I_2^i, ..., I_j^i, ..., I_{n_i}^i)$ – vector of input line numerical names of i primitive.

Consider the complete circuit superposition which is consequent unity of detected fault vectors for every primitive if its output line is predecessor for obtained set

$$L_Y = L_Y \cup L_R^1 .$$

The fault detecting analysis is used only for primary outputs complemented by RFO, which faults are detected on test-vector T_t:

$$L_Y = L_Y \cup L_R^1 \subseteq L_R = \{L_R^0, L_R^1\}; |L_R| = r,$$

where L_Y, L_R^0, L_R^1 - are primary output lines, RFO with undetected and detected faults correspondingly.

8. CONCLUSION

The offered fault simulation method is oriented on processing of complex digital devices implemented into PLD containing millions of gates. The program realization of the method was tested on several hundreds of combinational and sequential benchmarks and gave good speed-up results in comparison with classic parallel and deductive fault simulation algorithms.

The main result of given work is improving of deductive-parallel method [1-3] that consists in: 1) Creation of general deductive-parallel model of digital circuit analysis based on back-traced superposition procedure; 2) The development of algorithms for digital circuits structural and functional analysis with purpose of the RFO set definition and circuit structure reconfiguration for superposition procedure realization; 3) Creation of internal interpretative-compilative model of digital device.

9. REFERENCES

[1] V.I. Hahanov, A.V. Babich, S.M. Hyduke. *Test Generation and Fault Simulation Methods on the Basis of Cubic Algebra for Digital Devices. Proc. Euromicro Symposium on Digital Systems Design, Warsaw, Poland, 2001, 228-235.*

[2] V.I. Hahanov, HM. Jahirul, Haque Masyd M.D. Mehedi. *Fault analysis models of digital systems based on FPGA. Technologia i konsruirovanie v elektronnoi apparature, 2, 2001, 3-11.*

[3] V.I. Hahanov, I.Y. Sysenko, HM. Jahirul. *HaqueThe Cubic Fault Simulation of Digital Circuit based on FPGA. Radio Electronics, informatics, management, 1, 2001, 123-129.*

[4] Y.H. Levendel, P.R. Menon. *Comparison of fault simulation methods – Treatment of unknown signal values. Journal of Digital Systems. Vol. 4, 1980, 443-459.*

[5] M. Abramovici, M.A. Breuer, A.D. Friedman. *Digital System Testing and Testable Design (Computer Science Press, 1998)*

[6] V.I. Hahanov. *Technicheskaya diagnostika elementov i uzlov personalnih komputerov. Kiev: IZMN, 1997, 308p.*

[7] R. Ubar. *The analysis of diagnostic tests for combinational digital circuits by fault back tracing methods. Automatica and Telemechanica, 8, 1977, 168-176.*

Hahanov Vladimir Ivanovich, *Doctor of technical science.*

Area of interests: Technical diagnostics of computing devices, systems, networks, software products.

Education: 1978 Graduated from Kharkov National University of Radioelectronics. System analyst. Speciality: Automated control systems. Subject of graduate work: "Automatic system for digital device control"

Position: Professor of the Kharkov National University of Radioelectronics.

Krivoulya Gennady Fedorovich*, Doctor of technical science.*

Position: head of the Design Automation Department.

Scientific interests: Technical diagnostics of computing devices, systems, networks.

Education: 1966 Graduated from Kharkov National University of Radioelectronics.

Hahanova Irina Vitalievna, *PhD*

Position: professor assistant.

Scientific interests: Technical diagnostics of computing devices, systems, networks, software products.

Education: 1997 Graduated from Kharkov National University of Radioelectronics.

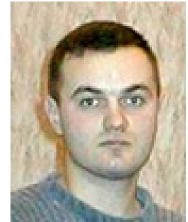
Melnikova Olga Vyacheslavovna

Area of interests: digital systems design and test; software quality assurance.

Education: 1999 Entering Kharkiv National University of Radioelectronics. Computer Science departament. Information Design Technologies speciality. Bachelor. Fifth year student at the moment.

Current position: researcher at the Design Automation Department of the Kharkiv National University of Radioelectronics.

Obrizan Volodymyr Igorovych



Area of interests: digital systems design and test; software quality assurance. Education: 2000 entering Kharkiv National University of Radioelectronics. Computer Engineering and Control departament. Computer Systems and Networks speciality. Fourth year student at the moment.

Current position: researcher at the Design Automation Department of the Kharkiv National University of Radioelectronics.