ELECTRONICS AND ELECTRICAL ENGINEERING

ISSN 1392 – 1215 *2008. No. 8(88)*

ELEKTRONIKA IR ELEKTROTECHNIKA

SYSTEM ENGINEERING, COMPUTER TECHNOLOGY

T 120

SISTEMŲ INŽINERIJA, KOMPIUTERINĖS TECHNOLOGIJOS

Embedded Method of SoC Diagnosis

V. Hahanov, E. Litvinova, V. Obrizan, W. Gharibi

Computer Engineering Faculty, Kharkov National University of Radioelectronics, Lenin av. 14, Kharkov, Ukraine, 61166, phone: (057) 70-21-421, (057) 70-21-326, e-mail: hahanov@kture.kharkov.ua; kiu@kture.kharkov.ua

I-IP Infrastructure

Computational and hardware complexity of modern digital systems on a chip (SoC) is characterized by millions of equivalent gates and requires making and implementation of new high-level design technologies: Electronic System Level (ESL) Design, Transaction Level Modelling (TLM) and embedded service – Infrastructure Intellectual Property (I-IP). It means that search for high-performance methods and facilities [1-7] reduces all researchers to necessity to increase an abstraction level of Functional Intellectual Property (F-IP) models, which are created and embedded into a chip. EDA market suggests facilities for computer-aided modelling and verification of system level devices, beginning with HDL- compilers (C++, SystemC, SystemVerilog, UML, SDL) up to graphics environments (Simulink, LabView, Xilinx EDK). These facilities enable to create projects using existing library components by means of ESL-mapping and creation of TLM-interfaces [5]. Market appeal of the implementation of a digital system to FPGA is determined by the followings: application of relatively cheap chips instead of the universal processors, low power consumption, small overall size, qualitative and reliable realization of the main functions due to on-chip I-IP-infrastructure that is urgent in the century of mobile computers.

The research aim is the development of an algebra-logical method of SoC Functional Intellectual Property Infrastructure that is intended for the diagnosis of SoC components in real time. The problems: 1) State of the market of SoC Infrastructure Intellectual Property technologies; 2) Algebra-logical (AL) method of Infrastructure Intellectual Property on basis of the cover matrix; 3) Application of the AL-method to diagnosis of SoC components; 4) Practical results.

Modern design technologies of digital systems on chips propose along with creation of functional blocks F-IP development of service modules I-IP, which are oriented on complex solving of the project quality problem and yield increase in manufacturing that is determined by implementation of the following services into a chip: 1) Observation for state of input and output lines in functioning, verification and testing of standard blocks on the basis of utilization of the boundary scan standard IEEE 1500 [7]; 2) Testing of functional modules by means of input of the fault detection patterns from different test generators, which are oriented on the verification of faults or fault-free state; 3) Fault diagnosis by means of analysis of an information obtained on the testing stage and utilization of special methods of embedded fault search on the basis of the standard IEEE 1500 [7]; 4) Repair of functional modules and memory after fixation of negative testing result, fault location and its type on diagnosis stage; 5) Measurement of the general characteristics and parameters of a device operation on basis of on-chip facilities, which enable to make time and volt-ampere measurements; 6) Reliability and fault tolerance of a device operation in working that is obtained by diversification of functional blocks, redundancy of them and repair of SoC in real time.

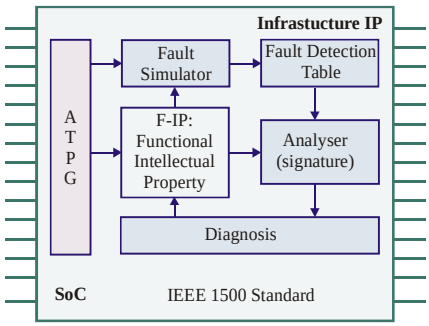

Fig. 1. Infrastructure Intellectual Property of SoC DSP

In Fig. 1 the reduced structure it is represented [5], oriented on solving the following problem: 1) testing of the functionalities on the basis of generalizable input patterns (Automated Test Pattern Generator) and analysis of output reactions; 2) Fault simulation [4] to ensure the diagnosis and repair on the basis of the fault detection table; 3) Fault diagnosis with given resolution of fault location by means of utilization of the IEEE 1500 multiprobe.

Automated Test Pattern Generator

Automated Test Pattern Generator for verification of functionalities and single faults consists of a set of input patterns generators for creation of the following tests [1,2]: pseudo-random generator (PRTG) of input stimuli with uniform distribution law of zero and unit signals at input

variables; test generator of hexadecimal codes (SATG) on the basis of the signature analysis; algorithmic generator of the test patterns, which activate one-dimensional logical paths (SPTG), oriented on verification of given single faults; test generator for verification of the summatory ALU circuits (ADTG); test generator for the bus organized structure of data transceiving (BSTG); test generator, based on the matrix memory verification (METG); test generator for automata, specified in the form of algorithm flow graphs (DFTG); test generator for sequential arithmetic-register structures and trigger circuits (RCTG).

Generator module analyzes the structural-functional model of a tested block and assigns a subset of such synthesizers, which provide given fault cover quality (F^c) and functional modes (P^c):

$$\begin{cases} F^c(\bigcup_{i=1}^{n_{\min}} T_i) \ge F^c_{\min};\ P^c(\bigcup_{i=1}^{n_{\min}} T_i) \ge P^c_{\min}, \\ T = \{T_1^{PR}, T_2^{SA}, T_3^{SP}, T_4^{AD}, T_5^{BS}, T_6^{ME}, T_7^{DF}, T_8^{RC}\}. \end{cases} \quad (1)$$

Generalized structure of Testbench synthesis [1] (Fig. 2) includes HDL-code generator for functional testing and verification on the stage of project development.

A number of test generators on the SoC development stage can be considerably greater than a subset of ones that is embedded into a chip later. So, on the simulation and verification stage the analysis of covering features of every test generator is performed to search for the minimal aggregate configuration of them that is satisfied expression (1). It is important to say that within the next 5 years the test synthesis ideology for SoC will borrow the best traditions of ESL-, TLM-design [6].

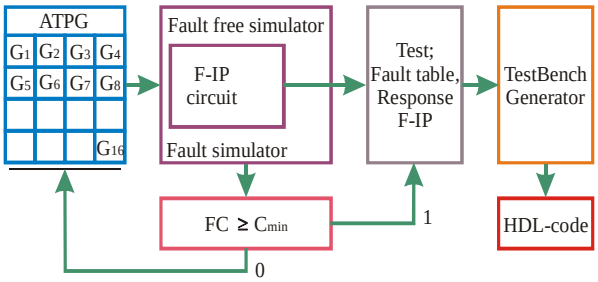

Fig. 2. Structure of the Testbench synthesis process for F-IP

It means: 1) Use of the Testbench libraries of the lead companies in the world for testing and verification of standardize functionalities, which are designated as F-IP. 2) Application of I-IP standard solution for on-chip testing of SoC components. 3) Creation of own test libraries for newly developed functionalities. 4) Adoption of new technology of the test synthesis for a digital system, based on the discrete mapping [6] of covering of functionalities and faults of the initial specification by means of minimal Testbench set from a test library. 5) Application of the on-chip testability facilities, such as IEEE boundary scan and six I-IP components, to increase of the technological effectiveness of test synthesis procedures.

Fault analysis module

Fault analysis module uses the deductive algorithm that is oriented on verification of single faults, which are generated on the basis of analytical or tabular definitions of SoC functionalities. It means that deductive simulation can be applied for projects, represented on gate abstraction level or on some higher one (register or system). The main idea of this method is making deductive functionality model on the basis of the known expression using [4]:

$$F = f[(X_1 \oplus T_1),(X_2 \oplus T_2),...,(X_j \oplus T_j),...,(X_{n_i} \oplus T_{n_i})] \oplus T_i, \quad (2)$$

where deductive function F on test-vector T is the modified definition of fault-free behaviour that enables to determine an input fault lists, which are transported on a circuit output under the influence of input signals. The model synthesis strategy, proposed in the paper, is based on the creation of a deductive element library that covers all standardize functional elements, which are used by a designer in the process of computer-aided project creation in a SoC form. In this case the matter is the synthesis of a deductive structure on basis of mapping [6].

The proposed approach to the deductive analysis provides creation one more embedded model on a chip, which must ensure practically all six services, specified by the I-IP infrastructure standard. “Pay” for quality of diagnostic and test assistance is a sufficiently large cost in addition to the hardware costs, which is greater than nominal functionality in 10-15 times. With that gain in the performance in comparison with the external software realization of the deductive analysis is 2-3 orders that practically provides the service in real time. Another more economical solution of the problem is the interactive modification of the deductive model circuit structure for every test-vector. For that the internal memory of a chip is used, where the model is formed in compliance with the rules, defined in (2). Mapping gives a deductive function, where the hardware costs is equal to the cost of F-IP functionality.

Algebra-logical method of the fault diagnosis

The general role, assigned to the boundary scan technology [7] that is implemented into a chip now, has to simplify solution of practically all problems of SoC Functional Intellectual Property Infrastructure. The access controller to internal lines and ports of the boundary scan register uses a cell or a stage of the register. In the aggregate a number of such cells, which provide monitoring in this case, must be equal to the quantity of problem observable lines of a project, which are necessary for exact establishment of a diagnosis. Diagnosis procedure, based on the boundary scan register uses information from the fault detection table that is a fault set, which are covered by test patterns. Using result information of a diagnostic experiment that is represented in an experimental validation vector form $V = (V_1,V_2,...,V_i,...,V_n)$ and fault detection table F [5], the diagnosis procedure is carried out in compliance with the expression, written in the product of disjunctions form for all faults, which can give an experimental reaction in the form of V that determined by unit and zero values:

$$F = \bigwedge_{\forall V_i=1}^{i=\overline{1,n}} \left(\bigvee_{\forall M_{ij}=1}^{j=\overline{1,m}} F_j \right). \quad (3)$$

The conjunctive normal form, derived from the fault detection table, is transformed to the disjunctive normal

form by means of equivalent transformations (conjunction, minimization and absorption). Therefore we have the Boolean function, where terms are the logical product, which represent full solution set in the fault combination form (they give the experimental validation vector V at SoC outputs or its component):

$$F = \bigwedge_{\forall V_i=1}^{i=\overline{1,n}} \left(\bigvee_{\forall M_{ij}=1}^{j=\overline{1,m}} F_j \right) = \left| \begin{matrix} a \vee ab = b \\ a \vee a = a \end{matrix} \right| = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^{m} k_j F_j \right), k_j = \{0,1\}. \quad (4)$$

Represented procedure in general case diagnoses some fault subset that later needs a refinement by application of additional flexing of internal points by means of the boundary scan register. An example of defect finding is considered on basis of the following fault detection table (columns are faults, rows are test patterns) that is product of the deductive fault analysis and the experimental validation vector:

M =

T_i / F_j	F_1	F_2	F_3	F_4	F_5	F_6	V
T_1	1			1			1
T_2		1			1		1
T_3			1	1	1		0
T_4	1		1				1
T_5		1			1	1	1

A number of units in the experimental validation vector V forms quantity of CNF disjunctive terms (4). Every term is line-by-line writing of faults (by logic operation OR), which influence on functional outputs. Table representation in the analytical form (conjunctive normal form) makes possible to reduce the volume of diagnostic information for defect finding essentially. Subsequent transformation of CNF to DNF on the basis of the Boolean algebra identities enables to reduce the Boolean function that is illustrated by the following result:

$$\begin{cases} F = (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ = (F_1 \vee F_4)(F_2 \vee F_5)(F_3 \vee F_4 \vee F_5)(F_1 \vee F_3) = \\ = (F_1F_2 \vee F_2F_4 \vee F_1F_5 \vee F_4F_5)(F_1F_3 \vee F_1F_4 \vee F_1F_5 \vee \\ F_3F_3 \vee F_3F_4 \vee F_3F_5) = (F_1F_2 \vee F_2F_4 \vee F_1F_5 \vee F_4F_5) \\ (F_1F_4 \vee F_1F_5 \vee F_3) = (F_1F_2F_1F_4 \vee F_2F_4F_1F_4 \vee F_1F_5F_1F_4 \vee F_4F_5F_1F_4) \\ (F_1F_2F_1F_5 \vee F_2F_4F_1F_5 \vee F_1F_5F_1F_5 \vee F_4F_5F_1F_5) \\ (F_1F_2F_3 \vee F_2F_4F_3 \vee F_1F_5F_3 \vee F_4F_5F_3) = \\ = F_1F_2F_3 \vee F_2F_3F_4 \vee F_1F_3F_5 \vee F_3F_4F_5 \vee F_1F_2F_4 \vee F_1F_4F_5 \vee \\ \vee F_1F_5 = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4. \end{cases} \quad (5)$$

To decrease the number of computing while carrying out of conjunction in the first line (5), the initial notation can be simplified in accordance with the Boolean algebra laws:

$$(F_2 \vee F_5) \wedge (F_2 \vee F_5 \vee F_6) = (F_2 \vee F_5). \quad (6)$$

The derived result

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4 \quad (7)$$

represents all possible solutions (fault covering of the fault detection table rows of SoC functionality on condition that the experimental validation vector has all unit coordinates V = (11111). Taking into account the actual value of the experimental validation vector V=(11011), the simulation of function F by substitution of zero fault values, is carried out which are verified theoretically, but they give zero coordinate in the vector V. Such fault is: $\bar{F} = F_3 \vee F_4 \vee F_5$. Final result is determined by the next function:

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee$$
$$\vee F_1F_2F_4 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = 0. \quad (8)$$

Whatever combination is DNF conjunctive term, presented in solution

$$F = F_1F_5 \vee F_1F_2F_3 \vee F_2F_3F_4 \vee F_3F_4F_5 \vee F_1F_2F_4 \quad (9)$$

covers all rows of the fault detection table according to the definition, so addition of any zero row transforms function F to zero without fail. So, correct solution that corresponds to the experimental validation vector must take into account zero coordinates of the vector V. Subject to the stated above it is necessary to eliminate the term $(F_3 \vee F_4 \vee F_5)$ from expression (7) on CNF forming stage

$$\begin{aligned} F &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3)(F_2 \vee F_5 \vee F_6) = \\ &= (F_1 \vee F_4)(F_2 \vee F_5)(F_1 \vee F_3) = \\ &= (F_1F_2 \vee F_2F_4 \vee F_1F_5 \vee F_4F_5)(F_1 \vee F_3) = \\ &= F_1F_2 \vee F_1F_2F_4 \vee F_1F_5 \vee F_1F_4F_5 \vee F_1F_2F_3 \vee \\ &\vee F_2F_3F_4 \vee F_1F_3F_5 \vee F_3F_4F_5 = \\ &= F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5. \end{aligned} \quad (10)$$

The result represents all possible solutions, which make a device reaction, determined by given experimental validation vector:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5. \quad (11)$$

Additional simulation of last Boolean function gives final solution in the form of two faults combination:

$$F = F_1F_2 \vee F_1F_5 \vee F_2F_3F_4 \vee F_3F_4F_5 \Big|_{F_3 \vee F_4 \vee F_5 = 0} = F_1F_2. \quad (12)$$

Further refinement of a diagnosis is possible by application of the multiprobe on the basis of the boundary scan register.

Algebra-logical diagnosis model

The structure of I-IP service modules for fault diagnosis in F-IP functional blocks is represented in Fig. 3. Comparator ($\oplus$) analyses output reactions of a model and a real device on input test vectors, entering from a test generator. Discrepancy between model and experimental reactions on a test forms unit coordinates of the experimental validation vector $V(T) = (V_1, V_2, ,..., V_i, ,..., V_n)$ for every input pattern. Communication between the vector V and the fault detection table ($T = [T_{tr}]$, $t = \overline{1,p}$; $r = \overline{1,q+n}$ of dimension $p \times n$, p is a number of test-vectors, n is a number of stages of the boundary scan register) and circuit structure gives a set of lines and elements, which are suspected as faulty on a current test-vector. To organize computational processes, which result in exact diagnosis, it is an important metrics or initial information representation form.

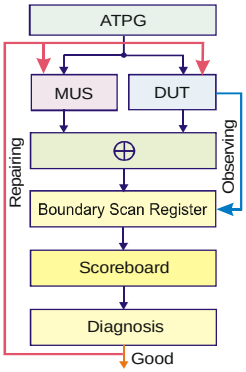

Fig. 3. Diagnosis process model for F-IP

An interesting solution of the diagnosis problem can be obtained by application of the Boolean algebra and the fault detection table M that is the Cartesian product of the test T on the set of given faults F, in the aggregate with the experimental validation vector V, where realization of the covering task gives maximally exact result in the DNF form and every term is a possible variant of presence of faults in a device. Thereby, the diagnosis process model is represented by components:

$$\begin{cases} A =< T, F, M, V >, \\ T = (T_1, T_2, ..., T_i, ..., T_n); \\ F = (F_1, F_2, ..., F_j, ..., F_m); \\ M = \left| M_{ij} \right|, i = \overline{1,n};\ j = \overline{1,m}; \\ V = (V_1, V_2, ..., V_i, ..., V_n); \\ \{V_i, T_i, M_{ij}, F_j\} \in \{0,1\}. \end{cases} \quad (13)$$

The diagnosis problem solution consists of analysis of the fault detection table, formed at fault simulation, by writing logical product of disjunctions (CNF), which are formed by unit values of the fault detection table rows (3). Then CNF is transformed to DNF (4) by means of equivalent transformations. Therefore Boolean function is turned out, where terms (logical products) are full solution set that is fault combinations, giving the experimental validation vector, formed in the process of diagnosis experiment, at functional outputs.

Sumulation for F-IP diagnosis refinement

Obtained disjunctive form (4) is basic model for defect finding. It does not always identify a functional fault definitely, so it needs procedures which improve diagnosis. First of all it should be noted that all rows of the matrix $M = T \times F$, which were marked by zero values of the experimental validation vector, can be joined in a disjunction of faults (4), which can not be present in a circuit.

The creation of form (3) from concerned fault detection table enables to determine all faults, which can not be present in a circuit:

$$\begin{aligned} \overline{F} &= (F_2 \vee F_7) \vee (F_3 \vee F_6 \vee F_9) \vee (F_1) \vee (F_1 \vee F_2) \vee \\ &\vee (F_3) \vee (F_5 \vee F_6) \vee (F_7) = \\ &= (F_2 \vee F_7 \vee F_3 \vee F_6 \vee F_9 \vee F_1 \vee F_5) = \\ &= (F_1 \vee F_2 \vee F_3 \vee F_5 \vee F_6 \vee F_7 \vee F_9). \end{aligned} \quad (14)$$

Analysis of the expressions, represented by formulas (13) and (14) results in interesting conclusions: 1) Faults, which can not be present in a circuit, are determined in the DNF terms, obtained by zero rows concerning the experimental validation vector; 2) Faults, which are in DNF, must be removed from function (14); 3) In this case removal of the fault F_5 results in breakup of two terms $F_4F_5F_9 \vee F_4F_5F_{10}$, as far as without the fault F_5, every one of them separately can not form given experimental validation vector; 4) So, it makes the sole conclusion – double fault that is determined by the term $F = (F_4F_8)$ is present in a circuit; 5) Computational complexity of gaining exact and full solution set is determined by expression $Q = 2^{m+1}(2m+1)$, m is a number of faults.

If to designate absence of the concrete fault $F_i = 0$, it can to form input conditions for DNF (11) for subsequent simulation of the function on the following initial conditions:

$$(F_1, F_2, F_3, F_5, F_6, F_7, F_9) = (0000000). \quad (15)$$

Then simulation result of the function $F = (F_4F_5F_9 \vee F_4F_5F_{10} \vee F_4F_8)$ is equal to $F = (F_4 0F_9 \vee F_4 0F_{10} \vee F_4F_8) = F_4F_8$.

Actually, if the faults $(F_1, F_2, F_3, F_5, F_6, F_7, F_9)$, which are verified on the test patterns theoretically, give the negative result (don't distort the output states), it means they are absent in a circuit. Support of this fact is corroborated by the following proof.

Lemma 1. Full set of all possible fault combinations, which are verified by the test T, is determined as DNF, and obtained by transformation of a conjunctive form

$$F = \bigwedge_{\forall V_i = 1}^{i=\overline{1,n}} \left(\bigvee_{\forall M_{ij}=1}^{j=\overline{1,m}} F_j \right) = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^{m} k_j F_j \right), \quad (16)$$

every term of that is written by unit values of the fault detection table row [18] $M = T \times F$ corresponding to the experimental validation vector state $V_i = 1$.

Initial information, formed in compliance with unit values of the experimental validation vector, is full model of faulty behavior of a real object, which forms the experimental validation vector with fixed quantity of units (fault detection table rows) that is equal to k. Every row forms a fault disjunction, written by OR. A number of such disjunctions is equal to k, they are logical multiplied and form full and consistent set of events (faults), which are present in a circuit simultaneously. By multiplication of elementary disjunctions with subsequent simplification of the expressions and using the axioms ($a \vee ab = b$; $a \vee a = a$) DNF that includes all possible combinations, written in the elementary conjunctions form, is turned out. Considering identity of made transformations, the obtained function is equivalent to the initial CNF at logic and it is technological notation of all solutions (fault combinations), which are in a circuit, essentially.

Lemma 2. All faults, verified in the fault detection table rows $M = T \times F$ and marked by zero values of the experimental validation vector $V_i = 0$ are absent in a real object.

In reality, the fault detection table $M = T \times F$ has unit and zero rows concerning the experimental validation vector value:

$$\begin{cases} M_p(0110) \to V_p = 1; \\ M_q(0101) \to V_q = 0. \end{cases} \quad (17)$$

The row p detects presence of two faults $F_2 \vee F_3$ in a circuit. The row q evidences of theoretical verification of the faults $F_2 \vee F_4$ if the vector is equal to 1: $V_q = 1$. But practically the signal $V_q = 0$ shows nonessentiality of the faults $F_2 \vee F_4$ for distortion of circuit outputs. Or these faults are absent in a tested device. Put zero signals for $F_2 \vee F_4$ in the function $F = F_2 \vee F_3$ and obtain the result: $F = F_2 \vee F_3 \big|_{F_2 = F_4 = 0} = F_3$. Analogous, all faults which are determined in the rows, corresponding to zero values of the experimental validation vector, are absent in a circuit. But if it is true they must be removed from DNF, written by unit values of the vector V. So, there are DNF terms and a fault set, which can not exist in a circuit for given experimental validation vector and the procedure of substitution of zero signals in the variables of elementary conjunctions of DNF function can be carried out. But, in consideration of the fact $0 \wedge a \wedge b \wedge c... = 0$ the result of substitution and subsequent transformations to obtain minimal function will have only the terms, which don't have variables (faults) with zero signal value. It means that the faults which concern to zero fault detection table rows (concerning the vector V), will be removed from DNF.

Theorem 1. Minimal set of all possible fault combinations, which are determined by the fault detection table $M = T \times F$, is computed by DNF simulation on an initial conditions set

$$F = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^{m} k_j F_j \right) \Big|_{(\forall F_q = 0) \leftarrow (\exists M_{pq} = 1) \& (V_p = 0)}, \quad (18)$$

specified by zero values of all verified faults, which correspond to zero signals of the experimental validation vector. In compliance with lemma 1 full set of all possible fault combinations, verified by a test, is determined in DNF form

$$F = \bigvee_{i=1}^{2^m} \left(\bigwedge_{j=1}^{m} k_j F_j \right), \quad (19)$$

that forms all solutions, which satisfy unit values of the experimental validation vector $V_q = 1$. It can be decreased by removal of the faults, which are verified by a test theoretically, but really they don't distort the output states on the test patterns, that mean complete absence of them in a real circuit. So, they can be removed from DNF terms, which is a full set of all possible combinations. The removal mechanism, according to lemma 2, is substitution of zero variable values to DNF terms and subsequent simulation (simplification) of the function. If a term has zero-component one of the variables F_i, according to the algebra of logic whole term is turned into 0, that means removal of it from DNF. After minimization subject to lemma 2 the minimal DNF is received that contains the minimal quantity of possible fault combinations (single and multiple ones) that can not be decreased without additional diagnostic information incoming from the multi-probe on the basis of boundary scan register.

So, proposed algebra-logical diagnosis method uses Boolean calculus as the basic apparatus for solving the covering task by getting the disjunctive form that then is minimized by removal of the terms, which have fault variables, relating to the rows with zero values of the vector V. For little quantity of faults in SoC the computational complexity enables to realize fault finding in real time.

Conditional diagnosis of F-IP by DNF

To decrease precautionary faults field the half-division method is used, based on the interactive procedure of internal check point flexing, that provides the obtained fault DNF by additional information to decrease a fault set. In this case as such tester the boundary scan register can be used that is able to determine an internal line state for fault removal or its confirmation. The check point choice strategy is based on approximately half-division of precautionary set (removal of half faults by simulation on every step) and simplification of the initial DNF. The essence of the half-division method on disjunctive normal form that represents all possible fault combinations in a circuit can be demonstrated by the following example:

$$F = (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8). \quad (20)$$

Choice of the first check point $F_9 = 0$ turns the Boolean function into reduced expression:

$$F = \begin{cases} F_9 = 0 \to (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8) = F_4 F_5 F_{10} \vee F_4 F_8; \\ F_9 = 1 \to (F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8) = F_4 F_5 F_9 \vee F_4 F_5 F_{10} \vee F_4 F_8. \end{cases} \quad (21)$$

If $F_9 = 1$, it means confirmation of a line fault and decrease of DNF size do not happen. It is necessary to orient the check point choice algorithm on maximal decrease of the initial DNF after definition of the initial conditions ($F_j = \{0,1\}$) for simulation. Weights of DNF powers, obtained in the process of simulation the both verification states, can be used as the check point choice criterion.

Check point choice rules are regulated by the following assertions.

Assertion 1. If F_j is present in all DNF terms, there exists given fault in a circuit without fail and it is not necessary to test it. Otherwise, if to suppose that verification result is zero, all terms is turned into zero and this fact contradicts to the existence condition of nonzero values of the experimental validation vector V.

Assertion 2. There is a single fault combination in a circuit that is determined by a single DNF term. If it is found one confirmed solution in the DNF term form other terms should be removed from consideration by reversal of them to zero.

The check point minimization problem is reduced to carrying out of two alternative strategies: 1) consideration of variables in the terms of minimal length to corroborate all faults in a term by flexing; 2) verification of such variables, which turn maximal quantity of DNF terms to zero.

Conclusion

Scientific novelty and practical importance of the research: 1) Algebra-logical method and algorithm of fault embedded diagnosis in functional blocks of SoC that uses preliminary analysis of the fault detection table for decreasing its size and the volume of subsequent calculations,

related with DNF forming, which determines all solutions of SoC functionalities diagnosis, are proposed. 2) Reduced SoC Functional Intellectual Property Infrastructure is proposed that is characterized by minimal set of the embedded diagnosis processes in real time and enables to realize the services: testing of the nominal functions on basis of generable input patterns (Automated Test Pattern Generator) and analysis of output reactions; fault diagnosis with given resolution of fault location by means of utilization of the IEEE 1500 multiprobe; fault simulation to provide realization of the first two procedures on the basis of the fault detection table. 3) The mapping model of the deductive structure is represented synthesis process that differs by utilization of the deductive component library, covering all standardized functional elements, it enables to create the SoC functionality deductive model in computer-aided mode. 4) The mapping-model of fault synthesis that differs by utilization of the embedded test generators library for DSP functionality of SoC is proposed: It enables to decrease the time of test construction essentially, at that tests are designed for functional and fault verification.

The algebra-logical representation of the covering problem has appeal that is directed on optimal solution of all synthesis and analysis problems of complex systems, where the mapping problem exists: 1) a specification – a set of library components; 2) faults – test patterns; 3) functionalities – Testbench; 4) faulty elements – reserved ones; 5) object states – surveillance lines.

To decrease the dimension of the mapping problem it is necessary to structure the initial model by means of the hierarchy making, that is typical and used everywhere in computer-aided design of systems (ESL-, TLM-technologies).

A priori definition of the fault detection table in the Boolean function form is attractive by its compactness, that on a concrete experimental validation vector is transformed to compact form, which determines DNF terms as all possible solutions of faulty repairable components.

Further research is oriented on the development of testability structure of the system and hardware BIRA module for embedded repair of whatever components in appearance of faults on production and operating stages.

References

1. **Bergeron J.** Writing testbenches: functional verification of HDL models. – Springer. – 2003. – 512 p.
2. **Rashinkar P., Paterson P., Singh L.** System-on-chip Verification: Methodology and Techniques. – Kluwer Academic Publishers. – 2002. – 393 p.
3. **Zhong Y., Dropsho S. G., Shen X., Studer A., Ding C.** Miss Rate Prediction Across Program Inputs and Cache Configurations // IEEE Trans. on Computers. – 2007. – P. 328–343.
4. **Hahanov V., Kteaman H., Ghribi W., Fomina E.** HEDEFS – Hardware embedded deductive fault simulation // Proceedings of the 3rd IFAC Workshop. – Rydzyna, Poland. – 2006. – P. 25–29.
5. **Yervant Z., Dmytris G.** Guest editors' introduction: Design for Yield and reliability // IEEE Design & Test of Computers. – May-June 2004. – P. 177–182.
6. **Densmore D., Passerone R., Sangiovanni-Vincentelli A.** A Platform-Based taxonomy for ESL design // Design &Test of Computers. – September-October 2006. – P. 359–373.
7. **DaSilva F., Zorian Y., Whetsel L., Arabi K., Kapur R.** Overview of the IEEE P1500 Standard // ITC International Test Conference. – 2003. – P. 988–997.

Received 2008 04 20

V. Hahanov, E. Litvinova, V. Obrizan, W. Gharibi. Embedded Method of SoC Diagnosis // Electronics and Electrical Engineering. – Kaunas: Technologija, 2008. – No. 8(88). – P. 3–8.

Algebra-logical method of optimal memory repair, based on solving of the covering problem of faults by reserved components by means of use of the Boolean algebra apparatus is proposed. The method enables to carry out the memory repair automatically in the process of functioning and it can have hardware or software embedded realization that is a service module of fault repairing. Ill. 3, bibl. 7 (in English; summaries in English, Russian and Lithuanian).

В. Хаханов, Е. Литвинова, В. Обризан, В. Гариби. Метод встроенного диагностирования SoC // Электроника и электротехника. – Каунас: Технология, 2008. – № 8(88). – С. 3–8.

Предложена инфраструктура сервисного обслуживания функциональностей SoC, которая отличается минимальным набором процессов встроенного диагностирования в реальном масштабе времени и дает возможность осуществлять сервисы: тестирование функциональностей на основе генерируемых входных последовательностей и анализ выходных реакций; диагностирование с заданной глубиной поиска дефектов в SoC; моделирование неисправностей в целях выполнения первых двух процедур на основе таблицы неисправностей. Предложен структурно-алгебраический метод встроенного диагностирования дефектов в функциональных блоках SoC, использующий предварительный анализ таблицы неисправностей в целях уменьшения ее объема и последующих вычислений, связанных с построением ДНФ, которая формирует все решения по установлению диагноза функциональностей SoC в реальном масштабе времени. Ил. 3, библ. 7 (на английском языке; рефераты на английском, русском и литовском яз.).

V. Hahanov, E. Litvinova, V. Obrizan, W. Gharibi. Įterptinis SoC sistemų diagnostikos metodas // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2008. – Nr. 8(88). – P. 3–8.

Pasiūlyta SoC funkcionalumo aptarnavimo infrastruktūra. Ji išsiskiria iš kitų minimaliu įterptinės diagnostikos realiu laiku procesų rinkiniu ir leidžia įgyvendinti tokias funkcijas: funkcionalumo testavimą remiantis generuojamomis įėjimo sekomis ir išėjimo reakcijų analize; SoC defektų paieškos gylio nustatymą diagnostikos metu; gedimų modeliavimą, siekiant realizuoti pirmąsias dvi funkcijas, remiantis gedimų lentele. Pasiūlytas struktūrinis-algebrinis įterptinės funkcinių SoC blokų defektų diagnostikos metodas, paremtas išankstine gedimų lentelės analize. Il. 3, bibl. 7 (anglų kalba, santraukos anglų, rusų ir lietuvių k.).