Masking Internal Node Logical Faults and Trojan Circuits Injections with Using SAT Solvers

A. Matrosova
Tomsk State University
Tomsk, Russia
mau11@yandex.ru

V. Provkin
Tomsk State University
Tomsk, Russia
prowkan@mail.ru

***Abstract*— We consider a combination circuit (the combinational part of a sequential circuit) and some nodes which faults are detected on the last stages of the circuit fabrication. Besides, injections of Trojan Circuits (TCs) in certain circuit lines may be detected. It is supposed that TC payload output is inserted into a line of combinational circuit C. In both cases circuit malfunction is connected with changing incompletely specified Boolean functions correlated with the corresponding fault nodes or the circuit lines. That is why masking (patch) circuit has to be an implementation of the correct incompletely specified Boolean functions system. This system is calculated with using SAT solver. Masking (patch) circuit inputs are connected with circuit *C* inputs and outputs are connected with nodes that are fed either by fault nodes or by circuit lines in that of which Trojan circuits are inserted. Experimental results are executed on circuits that behavior cannot be represented by ROBDDs because of their huge sizes. The results show that masking (patch) circuits may be essentially simpler than duplication.**

Keywords—boolean difference, ECO technologies, trojan curcuits, logical faults

I. INTRODUCTION

In spite of careful testing of high performance logical circuits during fabrication it is possible to detect logical faults of some internal circuit nodes and/or Trojan Circuits (TCs) injections into circuit lines during the last stages of circuit manufacturing. We consider Trojan circuits that of which payload output is inserted into internal line of circuit *C* (here we don't discuss methods of detection of the logical faults and the Trojan Circuits injections). In these cases it is possible to apply masking (patch) circuits in the frame of Engineering Change Ordering (ECO) technologies to correct the behavior of the given combinational circuit *C*. Here masking (patch) circuit outputs are connected with circuit *C* internal nodes that are fed either by fault nodes or circuit lines in that of which TCs are injected. Masking (patch) circuit inputs are linked with circuit C inputs.

Note that in this work we consider one of the problems of ECO technologies [1-6], namely, problem of forming patch circuit. Correct information on circuit *C* behavior is represented by its structural description. Commonly forming a patch circuit is based on results of simulation on the sub-set of input Boolean vectors [1-6] of circuit *C*. The last means that it is possible to guarantee the correct behavior of circuit *C* only on this sub-set. We suggest forming a patch circuit based on using incompletely specified Boolean function of the corresponding node (line). In this case circuit *C* malfunction of node *v* (line) may be detected on some input Boolean vectors (at least on one of them) during simulation. In the frame of our approach it is enough only to reveal that a malfunction on the considered node (line) takes place. Deriving masking circuit with using incompletely specified Boolean function we guarantee correct behavior of circuit *C* on all input Boolean vectors. It is the main difference of our approach from those suggested in [1-6]. In this paper we show that ECO technologies may be applied for masking of Trojan Circuit Injections. We don't thwart TCs injections [7], we only mask them after detection. We first represent on-set and off-set of incompletely specified Boolean function of node *v* by two circuits. From each circuit the irredundant sum of prime products (ISoP) is extracted by using SAT solver. These ISoPs are on-set and off-set compact presentation of incompletely specified Boolean function of the node (line). Patch circuit is constructed with applying ESPRESSO [8] and then ABC [9] systems to ISoPs obtained.

In section II the problem statement is considered. In section III deriving the incompletely specified Boolean function for node *v* with using Boolean differences and applying of the function is described. In section IV a simplification of a circuit representing Boolean Difference on variable *v* is suggested. The simplification is oriented to cut Conjunctive Normal Forms (CNFs) that are used for getting incompletely specified Boolean function of an internal node of circuit C. In section V a procedure of deriving patch circuit is given. Experimental results are shown in section VI.

II. PROBLEM STATEMENT

A combinational circuit *C* (the combinational part of a sequential circuit) is considered. Set *V* of internal nodes is given in that of which the circuit behavior differs from the correct one. The correct behavior is represented by structural description of circuit *C*. Different behavior on nodes from *V* may be generated either by logical faults of circuit *C* gates that outputs are nodes from set *V* or by Trojan Circuits (TCs) injected into lines that run to nodes from set *V*. In the last case the nodes are inputs of circuit *C* gates. It is necessary to provide the correct circuit behavior by masking faults on these nodes in the frame of ECO technologies. We suppose that patch (masking) circuit inputs are connected with inputs of circuit *C*. Outputs of patch circuit are linked with nodes that are fed either by nodes from set *V* or by lines. It is desirable to get masking circuit as simple as possible and simplify calculations of incompletely specified Boolean functions for nodes from set *V*. These functions depend on input variables of circuit *C*.

III. DERIVING AND APPLYING OF INCOMPLETELY SPECIFIED BOOLEAN FUNCTIONS FOR NODE *v*

Note that each internal node *v* of circuit *C* is characterized by incompletely specified Boolean function f_v. All test patterns for stuck at 1 fault of node *v* is off-set $M_0(f_v)$ of the function and all test patterns for stuck at 0 fault of node *v* is on-set $M_1(f_v)$ of the function [10].

For node *v* of circuit *C* we may derive completely specified Boolean function $\varphi(x_1, x_2, \ldots, x_n)$ that is realized by sub-

978-1-7281-7166-1/20/$31.00 ©2020 IEEE

circuit C_v of circuit C. Sub-circuit C_v output is node v and sub-circuit C_v inputs are inputs of circuit C. Function $\varphi(x_1, x_2, \ldots, x_n)$ at the same time is implementation of incompletely specified Boolean function f_v. Note that any implementation of f_v may be used as masking sub-circuit for fault of node v from V, but it is desirable to find as much as possible simple implementation.

We have already derived incompletely specified Boolean functions applying operations on ROBDDs [10]. On-set and off-set of such function are compactly represented by two ROBDDs. Unfortunately, for some combinational circuits it is impossible to describe the circuit behavior by the corresponding ROBDDs as the number of the ROBDD nodes is an exponential function of the number of circuit inputs under any order of variables of Shannon decomposition. For such circuits we suggest to get compact presentation of incompletely specified Boolean function by using SAT solver. In this case, we derive two irredundant sums of prime products (ISoP) for on-set and off set of the function. As SAT solvers become more and more efficient we hope that this approach will be applied to more and more complicated circuits.

Consider function $f_i(v, x_1, x_2, \ldots, x_n)$ implemented on the i-th output of circuit C under condition that this function depends on variable v together with variables $x_1, \ldots, x_n$. This function is derived from circuit C by a substitution of the corresponding gate functions instead of internal variables of circuit C. We stop a substitution when reaching variable v. Call the circuit implementing this function as $C_i(v, x_1, x_2, \ldots, x_n)$.

Find Boolean difference $D_v f_i$ of function f_i on variable v. Boolean difference $D_v f_i$ at the same time represents observability of node v on i-th output of circuit C. Actually, this function takes 1 value on Boolean vector depending on variables $(x_1, \ldots, x_n)$, if changing the value of variable v alters the value of i-th output of circuit C:

$$D_v f_i = f_i^{v=0} \oplus f_i^{v=1} \tag{1}$$

Here $f_i^{v=0} = f_i(0, x_1, \ldots, x_n)$, $f_i^{v=1} = f_i(1, x_1, \ldots, x_n)$.

For deriving observability function (f_{obs}) for node v and circuit C as a whole we use the formula:

$$f^{obs} = \bigvee_{i=1}^{m_v} (D_v f_i) \tag{2}$$

where m_v is the number of outputs of circuit C connected with node v.

We may represent on-set $M_1(f_v)$ (all test patterns for stuck-at 0 fault of node v) of incompletely specified Boolean function f_v for node v by the formula :

$$\varphi(x_1, x_2, \ldots, x_n) \,\&\, f^{obs}(x_1, \ldots, x_n) \tag{3}$$

and off-set of function f_v (all test patterns for stuck-at 1 fault of node v) by the formula :

$$\overline{\varphi(x_1, x_2, \ldots, x_n)} \,\&\, f^{obs}(x_1, \ldots, x_n) \tag{4}$$

Further it is necessary to implement the following steps.

1. Represent formulae (3), (4) by two circuits.
2. Simplify these circuits to cut the number of internal variables for the corresponding CNFs.
3. Derive CNFs from simplified circuits.
4. Get irredundant sums of prime products $ISoP_1$, $ISoP_0$ from the CNFs using SAT solver. These sums compactly present on-set and off-set of incompletely specified Boolean function f_v.
5. Derive patch circuit for node v by applying ESPRESSO and ABC systems to the obtained incompletely specified Boolean function.

When we have a set of fault nodes $V = \{v_1, \ldots, v_q\}$, it is necessary to obtain incompletely specified Boolean function for each fault node v_j from V representing it by the corresponding $ISoP_1(v_j)$, $ISoP_0(v_j)$.

IV. A SIMPLIFICATION OF A CIRCUIT IMPLEMENTING BOOLEAN DIFFERENCE $D_v f_i$

Boolean difference $D_v f_i$ may be represented by the following circuit (Fig. 1)

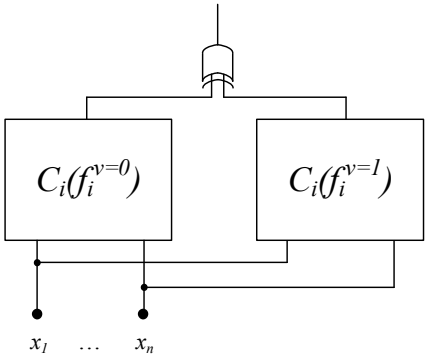

Fig 1. Circuit representing Boolean Difference $D_v f_i$

Note that circuit $C_i(v, x_1, x_2, \ldots, x_n)$ with input $v = 1$ implements function $f_i^{v=1}$ and circuit $C_i(v, x_1, x_2, \ldots, x_n)$ with input $v = 0$ implements function $f_i^{v=0}$. In these circuits may exist the same sub-circuits that do not depend on variable v. It means that it is possible to get the simpler circuit implementing $D_v f_i$ and, consequently, the simpler corresponding CNF. A simplification is based on excluding one of two identical sub-circuits.

An algorithm of deriving simplified circuit that implements $D_v f_i$

Pull out circuit $C_v(out)$ from circuit C in the following way.

1. Include into $C_v(out)$ all gates between node v and the i-th output of circuit C and all connections of these gates inter se. This circuit has the only output corresponding to the i-th output of circuit C. $C_v(out)$ inputs are free inputs of gates included into this circuit and node v is also the input of $C_v(out)$.

2. Divide free inputs of circuit $C_v(out)$ on s sets. Each node of the j-th set is connected with the same internal node w_j of circuit C.
3. Form a set $w_1, \ldots, w_s$ of such internal nodes. These nodes feed gates having an input connected with node v. Let nodes $w_1, \ldots, w_s$ be considered further as inputs of circuit $C_v(out)$ together with node v.
4. Obtain two circuits $C_v^1(out)$ and $C_v^0(out)$ from circuit $C_v(out)$ by fixing the value 1, 0 of input variable v, correspondingly.

Then we pick out circuit $C_v(in)$ from C with outputs $w_1, \ldots, w_s$ and inputs $x_1, \ldots, x_n$.

Simplified circuit is represented as follows (Fig. 2)

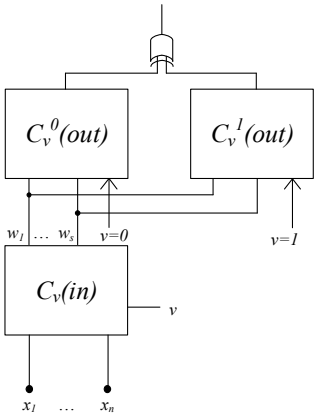

Fig 2. Simplified circuit implementing Boolean Difference $D_v f_i$

For getting CNF that represents observability of node v in multi output circuit C it is necessary to use the following circuit (Fig. 3).

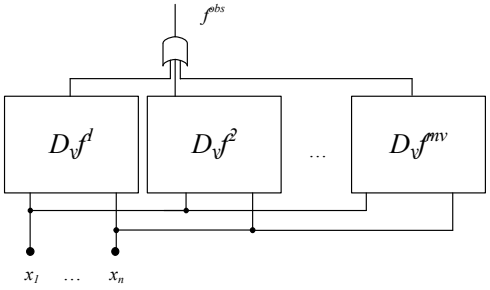

Fig 3. Circuit C_{obs}

After deriving circuit C_{obs} we may get circuits corresponding to formulae (3), (4) using circuits C_{obs}, C_v, gate AND (Fig. 4), additionally NOT for formula (4) (Fig. 5).

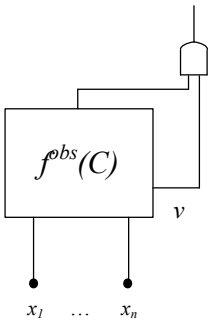

Fig 4. Circuit presenting on-set of incompletely specified Boolean function of node v

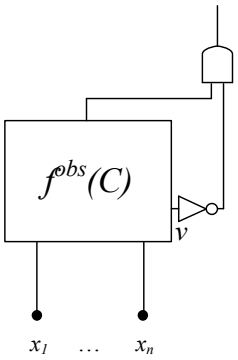

Fig 5. Circuit presenting off-set of incompletely specified Boolean function of node v

These circuits present on-set and off-set of incompletely specified Boolean function of node v. Using these circuits we construct the corresponding CNFs: CNF_{on} and CNF_{off}. Then we obtain $ISoP_1$ and $ISoP_0$ from these CNFs applying SAT solver by the procedure suggested in [11].

Note that experiments on ISCAS circuits show that the suggested simplification cuts half the number of internal variables (in average) of circuits representing incompletely specified Boolean functions.

V. Deriving Masking Circuit

Having got incompletely specified Boolean function represented by two ISoPs we may apply ESPRESSO [8] system to derive the corresponding ISoP of completely specified Boolean function. Having applied ABC [9] system to the obtained ISoP we get one output patch circuit C_p. The patch circuit inputs are connected with inputs $x_1, \ldots, x_n$ of circuit C and the patch circuit output is connected with nodes circuit C that are fed by node v (Fig. 6).

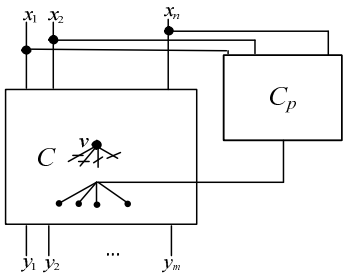

Fig 6. Including of patch circuit

Under masking Trojan Circuit included into a line of circuit C we connect C_p output with an input of gate that is fed by this line (Fig. 7).

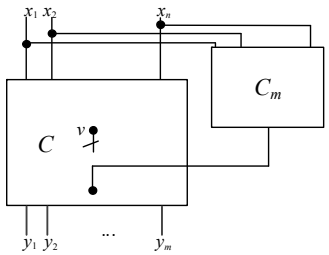

Fig 7. Including of patch circuit for line

For a set $V = \{ v_1, ..., v_q \}$ of fault nodes we derive a patch circuit with q outputs. In this case, it is necessary to get system of incompletely specified Boolean functions.

VI. Experimental results

We consider circuits for that of which we cannot get ROBDDs using computer.

We transformed ISCAS system circuits into other circuits that contain only two inputs or one input gates using ABC system. Patch circuits, as a rule, are derived for nodes with poor determined incompletely specified Boolean functions that is with low value observability nodes [10]. We choose some of such nodes for circuit. Experimental results are shown in Table 1. It is supposed that only one node may be fault or only one Trojan Circuit may be injected. Overhead is ratio of the gate number of circuit C_p to gate number of circuit C_v (in percentage). Note that it is important to appreciate overhead in comparson with sub-circuit C_v because this sub-circuit always may be applied for masking nodes (line) in the frame of the problem considered in this paper.

TABLE I. Complexity of masking circuits

Circuit name	Inputs	Outputs	Gates	Node	Gate number in C_v	Gate number in C_p	Overhead
s38584	1464	1730	13702	n722_n	135	87	64.44%
				new_n5106_	65	0 (constant 1)	0%
				new_n5093_	55	0 (constant 1)	0%
s15850	611	684	3719	new_n2410_1_	506	178	35.18%
				new_n2409_	385	174	45.19%
s13207	700	790	2972	new_n2424_	49	18	36.74%
				new_n2341_	52	31	59.62%
s6669	322	294	2433	new_n1300_	68	50	73.53%
				new_n2156_	70	28	40%

VII. Conclusion

The new method of masking of logical faults of nodes and Trojan circuit injections is developed. It is based on applying incompletely specified Boolean functions for internal circuit nodes. We use SAT solvers for deriving incompletely specified Boolean function for internal node of a combinational circuit that behavior cannot be represented by ROBDDs as the last ones are very huge. An algorithm of the simplification of the circuits representing on-set $M_1(f_v)$ and off-set $M_0(f_v)$ of the incompletely specified Boolean function of internal circuit node v to cut corresponding CNFs is suggested. Irredundant sums of prime products $ISoP_1$, $ISoP_0$ are derived from the CNFs by SAT solver. The patch circuit is obtained from this function with using ESPRESSO and ABC systems. Experimental results show progressiveness of this approach.

References

[1] C.-C.Lin, K.-C.Chen, S.-C.Chang, Marek Sadovska and K-I.Cheng. Logic synthesis for engineering change, in Proc. Design Automation Conference, 1995, pp.647-652.

[2] A. Veneris and I.Hajj. Design error diagnosis and correction via test vector simulation. IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems, 1999, vol.18, pp.1803-1816.

[3] K.-H. Chang, I.L. Markov and V. Bertacco. Fixing design errors with counter examples and resynthesis. In Proc. Asia and South Pacific Design Automation Conference, 2007, pp.944-949.

[4] S. Krishnavami, H. Ren,N. Modi and Puri. DeltaSyn: an efficient logic difference optimizer for ECO synthesis, in Proc. Asia and South Pacific Design Automation Conference, 2009, pp. 789-796.

[5] A.-C. Cheng, H.-R. Jiang and J.-Y. Jou, “Resource-aware functional ECO patch generation,” in Proc. DATE, 2016.

[6] A.Q. Dao, N.-Z. Lee, L.-C. Chen, M.P.-H. Lin, J.-H.R. Jiang, A. Mishchenko, and R. Brayton, “Efficient computation of ECO patch functions,” in Proc. DAC, 2018.

[7] H. Martin, L. Entrena, S. Dupuis and G. Di Natale A Novel Use of Approximate Circuits to Thwart Hardware Trojan Insertion and Provide Obfuscation, IOLTS, 2018.

[8] Logic Minimization Software (http://ramos.elo.utfsm.cl/~lsb/elo211/ aplicaciones/aplicaciones/espresso/ESPRESSO Logic Minimization Software.htm).

[9] ABC: A System for Sequential Synthesis and Verification (https://people.eecs.berkeley.edu/~alanmi/abc/).

[10] Matrosova A., Ostanin S. Trojan Circuits Masking and Debugging of Combinational Circuits with LUT Insertion // 2018 IEEE International Conference on Automation, Quality and Testing, Robotics. AQTR 2018 (THETA 21), 24-26 may 2018, Cluj-Napoca, Romania. [Cluj-Napoca], 2018. P. 462-467. 1 CD-R.

[11] Andre Inacio Reis, Rolf Drechsler, aditors, Advance Logic Synthesis. Springer International Publishing, 2018, pp.168-188.