2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2018)

Cluj-Napoca, Romania
24-26 May 2018

IEEE Catalog Number: CFP18AQT-POD
ISBN: 978-1-5386-2206-3

***** This is a print representation of what appears in the IEEE Digital Library. Some format issues inherent in the e-media version may also appear in this print version.***

IEEE Catalog Number:	CFP18AQT-POD
ISBN (Print-On-Demand):	978-1-5386-2206-3
ISBN (Online):	978-1-5386-2205-6

Additional Copies of This Publication Are Available From:

Curran Associates, Inc
57 Morehouse Lane
Red Hook, NY 12571 USA
Phone: (845) 758-0400
Fax: (845) 758-2633
E-mail: curran@proceedings.com
Web: www.proceedings.com

Trojan Circuits Masking and Debugging of Combinational Circuits with LUT Insertion

A. Matrosova, S. Ostanin
Tomsk State University (TSU)
Tomsk, Russia
mau11@yandex.ru, sergeiostanin@yandex.ru

Abstract—**It is extremely difficult to provide 100% correctness of fabricated high performance circuits. Manufactured circuits may have logical and electrical bugs, Trojan Circuits (TCs) inclusions and so on. Sometimes it is necessary to execute slight correction of the circuit specification. If there is in-field programmability in the circuits, they may be rectified. Here partially programmable circuits are considered. They are derived from gate circuits by covering some sub-circuits by look up tables (LUTs). Some LUTs have one free input and use reserved line. The way of selection of sub-circuits oriented to masking TCs or masking arbitrary logic faults on gate circuit lines is suggested. The selection is based on using observability estimations of internal circuit lines. Note that the internal line observability estimation represents at the same time determined area (its portion in Boolean space) of the incompletely specified Boolean function corresponding to the line. We get the estimations applying operations on ROBDDs. ROBDDs are derived from gate circuit fragments. A combinational circuit (the combinational part of a sequential circuit) from gates is given. We cover its sub-circuits by LUTs either to mask TCs actions or to recover the circuit specification if faults are detected on the last stages of the circuit fabrication. Arbitrary logical faults on lines are possible, several lines may be fault, several TCs may be injected. Algorithm of reprogramming LUTs is developed, some experimental results are given.**

Keywords—partially programmable circuits; malicious circuits (Trojan Circuits); observability; Reduced Ordered Binary Decision Diagrams (ROBDDs)

I. Introduction

It is extremely difficult to provide 100% correctness of fabricated high performance circuits. Manufactured circuits may have logical and electrical bugs, Trojan Circuits (TCs) inclusions and so on. Sometimes it is necessary to execute slight correction of the circuit specification. If there is in-field programmability in the circuits they may be rectified. It is known [1] that small programmability could be enough for that. In [2] the authors suggest covering some gate sub-circuits by Look-Up Tables (LUTs). Certain LUTs may have a free input. This input is connected with the reserved line. These circuits are called as partially programmable ones (PPCs) [3, 4, 5]. PPC may be applied for rectification under bugs [6] and for small changes of specification (Engineering Change Order (ECO)) [7]. PPC is one of possible ways of circuit rectification.

In papers [2, 8] the authors either use some heuristics [2] for selection of LUT inclusions or consider that PPC is already given and consists of gates and LUTs [8]. Our approach is different: we determine points of probable inserting LUTs being oriented to rectification under bugs and TCs insertions. For that we use random estimations of internal circuit lines observabilities that are calculated with applying operations on Reduced Ordered Binary Decision Diagrams (ROBDDS). ROBDDs are derived from circuit fragments. Note that the internal line observability estimation represents at the same time determined area (its portion in Boolean space) of the incompletely specified Boolean function corresponding to the line.

In papers [7, 8] reprogramming LUTs is reduced to solving Quantified Boolean Formula (QBF) SAT problems. Our approach is based on reprogramming LUTs with a permutation of LUT input variables. We consider that both a bug detected on a circuit line and a TC inserted into a line change the incompletely specified Boolean function corresponding to the line of a correct circuit. We suggest masking either bug or TC in the same way.

When outsourcing services are used for VLSI fragments, TCs insertion is possible. TC may destroy VLSI or provide leakage of confidential information. TCs are usually not detectable neither during VLSI verification nor VLSI testing as they act in very rare situations.

A Trojan Circuit (TC) consists of two parts (Fig. 1). One of them is a Trojan trigger that is switched on when the special combination of signals appears on TC inputs. Another part is a Trojan payload that is switched on by the trigger sub-circuit. Internal nodes IN_1, IN_2 (Fig. 1) are applied to switch on the trigger of the TC and the payload output is inserted into the line between two gates of the correct circuit fragment. When the TC acts, the value on the fragment output (Fig. 1) is changed from 0 to 1. It is desirable to mask TCs. We suggest masking TCs applying partially programmable circuits.

In high performance VLSIs it is possible detecting faults on the last stages of their fabrication. It decreases the manufacturing yield and increases the design cycle. Using PPCs we may rectify VLSIs reprogramming some LUTs. In [2] the method of increasing yield based on using PPC was suggested. They apply conventional logic circuit and cover some its sub-circuits by LUTs that may be reprogrammed to

978-1-5386-2203-2/18/$31.00 ©2018 IEEE

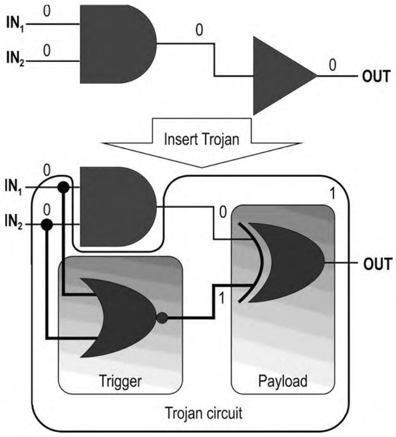


Fig. 1. TC insertion.

recover the circuit specification. It is supposed that single stuck-at faults may be detected and only one fault is possible.

In this paper a combinational circuit (the combinational part of a sequential circuit) from gates is given. It is necessary to cover some sub-circuits by LUTs either to mask TCs actions into the given circuit or to recover the circuit specification if faults are detected on the last stages of circuit fabrication. Arbitrary logical faults on lines are possible, several lines may be fault, several TCs may be masked.

Here a selection of sub-circuits to cover LUTs is based on finding lines that probably may be used for inserting TCs or be fault. These lines have low observability estimations and, consequently, their incompletely specified Boolean functions are poor determined. The selection is executed with applying operations on ROBDDs. ROBDDs are derived from fragments of the given combinational circuit (the combinational part of a sequential circuit).

Denote the given combinational circuit from gates as C. Circuit C turns into partially programmable circuit C_p by covering the chosen sub-circuits by LUTs. Some of the LUTs have free inputs (usually one input for one LUT). Reserved line connects free input with output of the appropriate circuit gate. These LUTs are reprogrammed either to mask TCs actions or mask logical faults of the lines. The procedure of reprogramming is developed. Specifications of circuit C and fault free circuit C_p in which of that TCs do not act are the same. We consider that TC output is inserted into line between circuit elements. Several TCs may be inserted into circuit C_p.

The behavior of an activated TC which output is inserted into line l is similar to manifestation of logical fault of this line on some test patterns (on full states of a sequential circuit). These test patterns are Boolean vectors depending on circuit C_p input variables. Some of circuit output values are changed for opposite ones on these test patterns. Thus, a TC changes the Boolean function corresponding to line l and, consequently, incompletely specified Boolean function of this line. That is why we may apply the same rectifying method both for masking a TC inserted into line l and masking a logical fault of line l. Reprogramming the proper LUTs we keep circuit C_p specification in spite of insertion of TCs or faults appearance on some lines.

In Section II a statement of the problem is discussed. In Section III we consider the way of calculation of random observability estimation. In Section IV an algorithm of reprogramming LUT is suggested. In Section V some experimental results on benchmarks are given.

II. A Statement of the Problem

Combinational circuit C (the combinational part of a sequential circuit) from gates is given. It is necessary to cover some sub-circuits by LUTs either to mask TCs or to debug faults detected on the last stages of circuit fabrication. We consider that TC output may be connected with only one circuit line (Fig. 1). In general case several TCs may be inserted into circuit C and several lines may be fault. We consider the way of inserting LUT with free input as it is suggested in [2] (Fig. 2).

Node v is linked with LUT input u_i by thick line l. The dashed line is reserved one. It links free input with the gate output ω.

More general, LUT with output w has m inputs and covers sub-circuit C_{LUT} from gates. C_{LUT} has (m-1) inputs. It is necessary to mask line l applying free input. Mark this input by m. Further we will call this LUT as rectifying one. Systems of Boolean functions (specifications) of circuits C and C_p without TCs and faults are the same. C_w is sub-circuit with output w, its input variables are input ones of circuit C_p. Let line l be linked with TC output. If TC is inserted into line l, sub-circuit C_w changes its functioning when TC acts. The function of circuit C_p is also changed. It is necessary to recover the specification of circuit C_p. Our preliminary studies show that possibilities of providing LUT rectification by way suggested in [2] (Fig. 2) are very rare. That is why we suggest connecting dashed (reserved) line with node v and free input m (Fig. 3). Here line l is thick one.

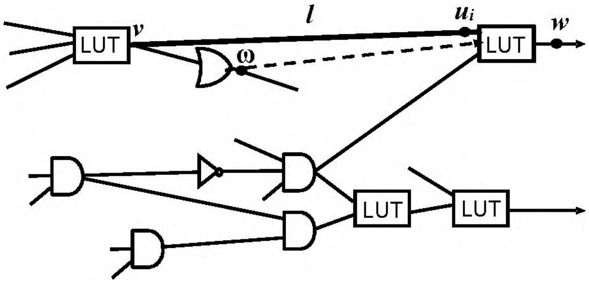


Fig. 2. LUT with free input and reserved line.

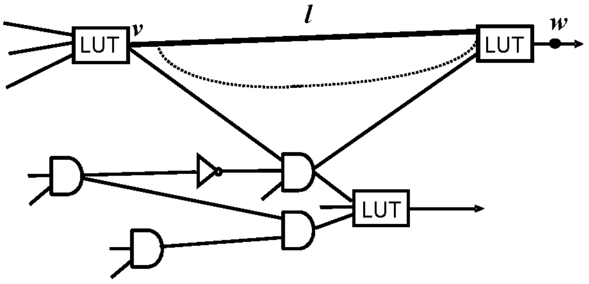


Fig. 3. Reserved line connects node v and free LUT input.

Take into account the following. For each internal line and internal node we may derive incompletely specified Boolean function.

We find lines that of which observability estimations are less than the given threshold. The determined areas are also less than the given threshold for the corresponding incompletely specified Boolean functions.

As a result, we get a set of suspicious lines in circuit C. A set of suspicious lines is divided into two sets. We include into the first set lines running from branch-points of circuit C and into the second one – the rest suspicious lines. Lines of the first set have to be masked with using rectifying LUTs.

The certain lines of the second set may be covered by LUTs so that input and output of the line are inside of the proper LUT (Fig. 4). Such LUT is not reprogrammed. We would like to cover in such way as much as possible suspicious lines from the second set. Lines that cannot be covered we include in the first set and mask them with rectifying LUT (Fig. 3).

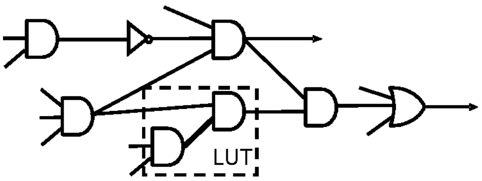


Fig. 4. Covering suspicious line by LUT.

After covering some gate sub-circuits in the above mentioned way we may program rectifying LUTs in different ways so that suspicious lines and reserved lines may be transposed. After that we get circuit C_p. As a result, violator having circuit C_p does not know beforehand which of input lines of rectifying LUT is inessential. Even if he determines essential lines and includes TCs in them, we may reprogram rectifying LUTs of circuit C_p and mask TCs activity. Thus C_p is tolerant to TC insertion. If we use C_p for masking logical faults on lines, suspicious lines and reserved lines cannot be transposed before faults detection.

III. Calculation of Observability Estimations and Finding Incompletely Specified Boolean Functions

Observability is a possibility of monitoring of changing 1(0)-value of the internal line on the corresponding circuit output. We suggest calculating random estimations of internal line observability based on applying ROBDDs operations [9].

To get observability estimation for internal line l of circuit C and the corresponding circuit output we construct ROBDD R (C_l) for sub-circuit C_l. The sub-circuit correlates with the proper corresponding circuit C output. The sub-circuit is derived from circuit C under the condition that internal line l is an input of sub-circuit C_l [9]. During construction of ROBDD $R(C_l)$ variable l is the first variable of the decomposition. It means that ROBDD $R(C_l)$ root is marked by variable l. ROBDD $R(C_l)$ implements function f^l.

Then we get (from $R(C_l)$) ROBDDs $R(f^{l=0})$, $R(f^{l=1})$ which roots are daughter nodes of $R(C_l)$ root. These ROBDDs implement functions $f^{l=0}$ and $f^{l=1}$, accordingly. Multiplications $R(f^{l=0})\,\&\,\overline{R(f^{l=1})}$, $R(f^{l=1})\,\&\,\overline{R(f^{l=0})}$ are implemented and results are merged being presented by ROBDD R^{obs}. Note that ROBDD operations have a polynomial complexity.

Take into account that deriving $\overline{R(f^{l=0})}$, $\overline{R(f^{l=1})}$ from $R(f^{l=0})$, $R(f^{l=1})$ is reduced to permutation of terminal nodes of the corresponding ROBDDs.

We suggest that probabilities of 1 value of all input variables are equal to ½. Using ROBDD R^{obs} we calculate observability estimation for line l and the corresponding circuit C output.

Let 1 value probability $\rho(\varphi)$ of Boolean function φ be correlated with ROBDD internal node μ (node μ is noted by variable x_i). The value is calculated with applying probabilities $\rho(\varphi^{x_i=1})$, $\rho(\varphi^{x_i=0})$ of 1 values of functions $\varphi^{x_i=1}$ and $\varphi^{x_i=0}$ corresponding to daughter nodes of node μ in the following way: $\rho(\varphi)=\rho(x_i)\rho(\varphi^{x_i=1})+\rho(\overline{x}_i)\rho(\varphi^{x_i=0})$.

Illustrate observability calculation by example (Fig. 5).

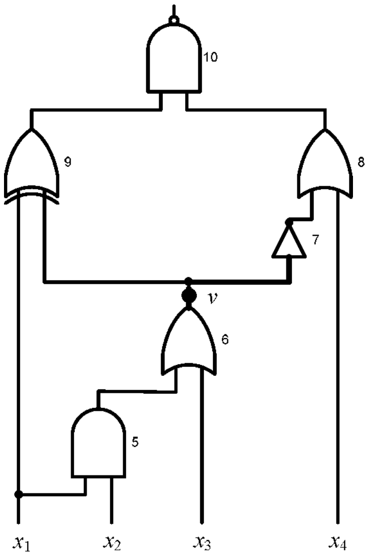


Fig. 5. Combinational circuit C.

Here *v* is one of branch-points and line *l* is input of number 7 gate. Let u_7 be output of gate with number 7. Then

$$f(x_1, x_2, x_3, x_4, l) = u_{10} = \overline{u_9 u_8} = \overline{u}_9 \vee \overline{u}_8 =$$

$$= x_1 u_6 \vee \overline{x}_1 \overline{u}_6 \vee \overline{u}_7 \overline{x}_4 =$$

$$= x_1(x_1 x_2 \vee x_3) \vee \overline{x}_1(\overline{x}_1 \vee \overline{x}_2)\overline{x}_3 \vee \overline{x}_4 l =$$

$$= x_1 x_2 \vee x_1 x_3 \vee \overline{x}_1 \overline{x}_3 \vee \overline{x}_4 l \ .$$

Fig. 6 shows ROBDDs $R\ (f^l)$, $R(f^{l=0})$ and $R(f^{l=1})$.

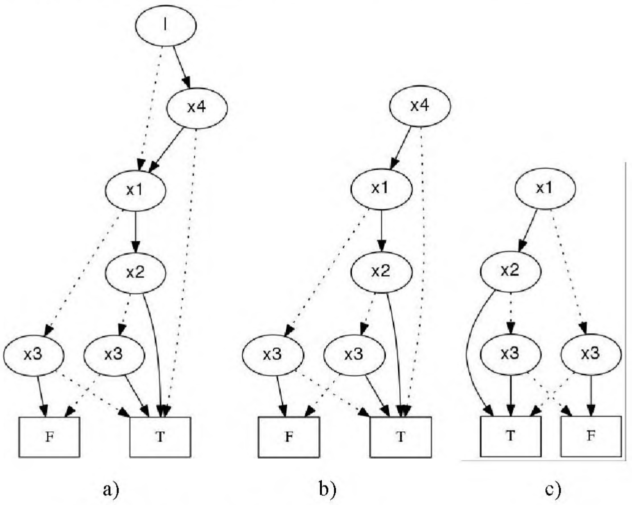


Fig. 6. ROBDDs $R(f^l)$ (a), $R(f^{l=1})$ (b) and $R(f^{l=0})$ (c).

$$R(f^{l=0}) = x_1 x_2 \vee \overline{x}_1 \overline{x}_3 \vee x_1 \overline{x}_2 x_3 \ .$$

$$\overline{R(f^{l=0})} = \overline{x}_1 x_3 \vee x_1 \overline{x}_2 \overline{x}_3 \ .$$

$$R(f^{l=1}) = \overline{x}_4 \vee x_4 \overline{x}_1 \overline{x}_3 \vee x_4 x_1 x_2 \vee x_4 x_1 \overline{x}_2 x_3 \ .$$

$$\overline{R(f^{l=1})} = x_4 \overline{x}_1 x_3 \vee x_4 x_1 \overline{x}_2 \overline{x}_3 \ .$$

Here we multiply Disjoint Sum of Products for calculation of the observability value.

$$R(f^{l=1})\overline{R(f^{l=0})} =$$

$$= (\overline{x}_4 \vee x_4 \overline{x}_1 \overline{x}_3 \vee x_4 x_1 x_2 \vee x_4 x_1 \overline{x}_2 x_3)(\overline{x}_1 x_3 \vee x_1 \overline{x}_2 \overline{x}_3) =$$

$$= \overline{x}_1 x_3 \overline{x}_4 \vee x_4 x_1 x_2 \vee x_1 \overline{x}_2 \overline{x}_3 \overline{x}_4 \ .$$

$$R(f^{l=0})\overline{R(f^{l=1})} =$$

$$= (x_1 x_2 \vee \overline{x}_1 \overline{x}_3 \vee x_1 \overline{x}_2 x_3)(x_4 \overline{x}_1 x_3 \vee x_4 x_1 \overline{x}_2 \overline{x}_3) = 0 \ .$$

Thus, observability estimation of line *l* is equal to 1/8 + 1/8 + 1/16 = 5/16. This line has rather high observability estimation, it is not attractive for TC injection.

If we have additional information about TCs, we may decrease the number of lines applying partial observability estimations of the lines for concrete circuit output and concrete changing signal value on the proper internal circuit node *v* [10].

If circuit *C* has several outputs, and line *l* is connected with some of outputs of this circuit, we calculate the observability estimation in the above mentioned way for each output of sub-circuits containing line *l*. The minimal value is chosen as observability estimation of line *l*. We consider the most rare situations connected with propagation of changing signals from internal line to the circuit output.

Calculate incompletely specified function for line *l* using ROBDD R^{obs} [9]. For that it is necessary to derive ROBDD R_v for sub-circuit C_v.The sub-circuit has output node *v* and its inputs are inputs of circuit *C*. R_v represents the behavior of this sub-circuit. $\overline{R_v}$ represents inversion function of sub-circuit C_v.

Let M_l^1 and M_l^0 be on-set and off-set of incompletely specified Boolean function corresponding to line *l*. R_l^1 and R_l^0 are ROBDDs compactly representing M_l^1 and M_l^0 of this function:

$$R_l^1 = R_v \,\&\, R^{obs}\ ,\ R_l^0 = \overline{R}_v \,\&\, R^{obs}\ .$$

Note that ROBDD R_l^1 presents all test patterns for stuck-at 0 fault on line *l* and ROBDD R_l^0 presents all test patterns for stuck-at 1 fault [9]:

$$R_l^1 \vee R_l^0 = R^{obs}\ .$$

It means that ROBDD R^{obs} applying for random observability estimation of line *l* represents at the same time determined area (its portion in Boolean space) of the incompletely specified Boolean function corresponding to line *l*.

IV. Masking Trojan Circuit and Arbitrary Fault on Line *l* by LUT

First of all we have to pick out sub-circuit C_{LUT} for line *l*. The circuit has one output and the number of its inputs is not more than (*m*-1). The certain input u_i of sub-circuit C_{LUT} is connected with line *l*. We include into the sub-circuit as much gates as possible. In worse case the sub-circuit may contain one gate with two inputs. We consider that the number of LUT inputs is not less than three. Function f_{LUT} is derived from the sub-circuit. It depends on input variables of sub-circuit C_{LUT}. The function on-set is represented by Boolean vectors depending on (*m*-1) variables. Each Boolean vector from the on-set is changed for two vectors depending on m variables with 1 and 0 values of variable u_m (here u_m corresponds to the certain LUT input that is free one). These Boolean vectors comprise on-set of rectifying LUT. The on-set corresponds to the situation when TC is not inserted into circuit C_p. If we reprogram LUT, It is necessary to form the following function depending on *m* variables:

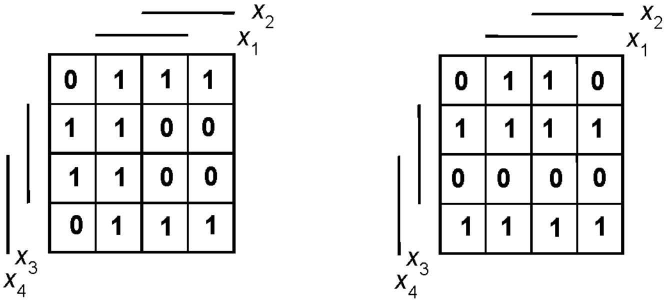
$$f_{LUT}(u_i = 1) \wedge u_m \vee f_{LUT}(u_i = 0) \wedge \overline{u}_m \ .$$

This function essentially depends on (*m*-1) variables: u_1, …, u_{i-1}, u_{i+1}, …, u_m. The function of (*m*-1) variables is derived from f_{LUT} by permutation of variables: variable u_i is changed for variable u_m. As a result variable u_m becomes essential one. Each on-set vector of this function is changed for two Boolean vectors depending on m variables: one with 1 value of variable u_i another with 0 value of variable u_i. Variable u_i becomes inessential. If TC is injected into line l it can't change the correct behavior of circuit C_p.

Take into account that after covering sub-circuits in circuit C we may program LUTs choosing as essential variable either u_i or u_m. It keeps circuit C_p specification, but complicates TC injection.

If nevertheless violator finds the essential line and injects TC into it, this fact may be detected by simulation of circuit C_p on Boolean vectors from on-set and off-set (vectors of determined area) of incompletely specified Boolean function corresponding to line l. It is possible because suspicious lines have poorly determined incompletely specified Boolean functions. When TC acts it changes the function of the line in which it is injected. Under going on Boolean vectors from determined area TC manifests itself obligatory. After detection TC we may mask it by reprogramming LUT in above mentioned way.

Fig. 7 illustrates the reprogramming procedure. Fig. 7a) shows the function in which variable u_2 is essential and variable u_4 is inessential. After variable permutation variable u_4 becomes essential and variable u_2 – inessential.



a) b)

Fig. 7. Permutation of variables.

Thus, after reprogramming the LUT implements function of Fig. 7b) instead of function of Fig. 7a).

If logical fault is detected on line l we reprogram the rectifying LUT in above mentioned way to mask this fault. It is possible to mask logical faults on several lines of circuit C_p.

V. Experimental Results

The experimental results on calculation of observability estimations on combinational MCNC benchmark circuits were performed. Firstly, the initial circuits were transformed into ones only with two-input AND gates which may have inverters in its inputs and outputs if necessary. These circuits were obtained by the tool ABC [11] as some kind of And-Inverter Graphs (AIGs). The experimental results are presented in Table I. **Circuit** is the name of an MCNC benchmark circuit, **PI/PO/Lines** are the number of primary inputs, outputs, and internal lines. **Cnt** is the number of internal lines with minimum observability. **MinObs** is the minimum observability value among all internal lines.

TABLE I. The experimental results

Circuit	PI	PO	Lines	Cnt	MinObs
9symml	9	1	246	1	0.00585938
alu2	10	6	558	6	0.000976563
alu4	14	8	1077	3	0.000183105
apex6	135	99	837	2	0.0000314713
apex7	49	37	241	18	0.000144958
b9	41	21	89	1	0.00146484
c432	36	7	252	3	0.000434422
c8	28	18	135	2	0.00292969
cc	21	20	54	97	0.031250
cm42a	4	10	31	280	0.125000
cm82a	5	3	23	1	0.187500
cordic	23	2	64	4	0.0032959
count	35	16	156	1	0.0000152588
cu	14	11	36	2	0.00244141
decod	5	16	40	592	0.062500
example2	85	66	316	4	0.000244141
f51m	8	8	149	1	0.0117188
i1	25	16	36	27	0.00195313
i5	133	66	225	1	0.00000762939
i6	138	67	592	115	0.187500
i9	88	63	1062	63	0.00390625
lal	26	19	94	1	0.00109863
pm1	16	13	41	1	0.0078125
sct	19	15	73	2	0.00244141
term1	34	10	224	1	0.000305176
ttt2	24	21	174	2	0.000732422
vda	17	39	1187	8	0.00012207
x3	135	99	775	4	0.000179291
x4	94	71	362	353	0.00195313

These calculations show that minimal values of observability estimations are very different. For circuits (cm42a, cm82a, cc, i6 and some others) risks of inserting TCs or finding faults on last stages of circuit fabrication are rather low. For each of the rest circuits we have to choose the threshold taking into consideration the possible number of LUTs and values of observability estimations for all internal lines of the circuit considered. Note that we calculated observability estimations for each line of circuits during the experiment.

VI. Conclusion

The approach to Trojan Circuits masking and debugging based on LUT insertion in combinational circuits (combinational parts of sequential circuits) is suggested. We turns the given circuit into partially programmable one (PPC) consisting of gates and LUTs in order to mask TCs and debug circuits when faults are detected on the last stages of the circuit fabrication. Some LUTs have one free input and reserved line connected with this input. The algorithm of reprogramming these LUTs is developed. The way of choosing lines being suspicious for inserting TC or being fault is suggested. This way is connected with calculations of observability estimations for internal circuit lines. The calculations are based on ROBDDs operations. ROBDDs are derived from combinational circuit (combinational part of a sequential circuit) fragments. Several arbitrary logical faults on circuit lines are possible. Their debugging is executed by reprogramming the proper LUTs. If TCs are inserted into several lines they may be detected by simulation in determined area of the corresponding circuit lines and then masked. Scalability of the approach is connected with complexity of ROBDDs involved from combinational circuits.

Acknowledgment

The reported study was supported by Russian Science Foundation, research project № 14-19-00218.

References

[1] S. Sarangi, S. Narayanasamy, B. Carneal, A. Tiwari, B. Calder, and J. Torrellas, “Patching Processor Design Errors with Programmable Hardware,” IEEE Micro, vol. 27, no. 1, pp. 12–25, Jan. 2007.

[2] S. Yamashita, H. Yoshida, and M. Fujita, “Increasing yield using partially-programmable circuits,” in Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), 2010, pp. 237– 242.

[3] A. Matrosova, S. Ostanin, I. Kirienko, and V. Singh, “Partially programmable circuit design,” in Proc. of 2014 East-West Design Test Symposium (EWDTS), 2014, pp. 1–4.

[4] A. Y. Matrosova, S. A. Ostanin, and I. E. Kirienko, “Providing Reliability of Physical Systems: Partially Programmable Circuit Design,” Russian Physics Journal, vol. 57, no. 6, pp. 847–852, Oct. 2014.

[5] A. Matrosova, S. Ostanin, and I. Kirienko, “Increasing Manufacturing Yield Using Partially Programmable Circuits with CLB Implementation of Incompletely Specified Boolean Function of the Corresponding Sub-Circuit,” in 2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits Systems, 2015, pp. 267–270.

[6] A. Matrosova, S. Ostanin, and V. Andreeva, “Patching circuit design based on reserved CLBs,” in 2016 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), 2016, pp. 1–5.

[7] H. Mangassarian, H. Yoshida, A. Veneris, S. Yamashita, and M. Fujita, “On error tolerance and Engineering Change with Partially Programmable Circuits,” 2012, pp. 695–700.

[8] S. Jo, T. Matsumoto, M. Fujita, “SAT-based automatic rectification and debugging of combinational circuits with LUT insertions”, Proc. Of IEEE Asian Test Symposium, 2012, pp. 19-24.

[9] A.Yu. Matrosova, S.A. Ostanin, I.E. Kirienko, “Generating all test patterns for stuck-at faults at a gate pole and their connection with the incompletely specified Boolean function of the corresponding subcircuit,” Proceedings of the 14th Biennial Baltic Electronics Conference. Tallinn, Estonia, 2014, pp. 85-88.

[10] A. Y. Matrosova, I. E. Kirienko, V. V. Tomkov, and A. A. Miryutov, “Reliability of Physical Systems: Detection of Malicious Subcircuits (Trojan Circuits) in Sequential Circuits,” Russ. Phys. J., vol. 59, no. 8, pp. 1281–1288, Dec. 2016.

[11] R. Brayton and A. Mishchenko, “ABC: An Academic Industrial-Strength Verification Tool,” in Computer Aided Verification, vol. 6174, T. Touili, B. Cook, and P. Jackson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 24–40.