ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2018 Управление, вычислительная техника и информатика № 42

# ПРОЕКТИРОВАНИЕ И ДИАГНОСТИКА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

УДК 681.5.09
DOI: 10.17223/19988605/42/10

**A.Yu. Matrosova, E.V. Mitrofanov, S.A. Ostanin, N.B. Butorina, E.G. Pakhomova, S.A. Shulga**

## DETECTION AND MASKING OF TROJAN CIRCUITS IN SEQUENTIAL LOGIC

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

Inserting malicious sub-circuits that may destroy a logical circuit or provide leakage of confidential information from a system containing the logical circuit demands detection of such sub-circuits followed their masking if possible. We suggest a method of finding a set of sequential circuit nodes in which Trojan Circuits (TC) may be inserted. After simulating the sequential circuit on the proper input sequences we may find TC if it is present and mask it by the special sub-circuit. The method is based on applying the precise (not heuristic) random estimations of internal nodes controllability and observability calculated with using a structural description of the combinational part of the sequential circuit. These estimations are computed with applying a State Transition Graph (STG) description, if we suppose that TC may be inserted out of the working area (out of the specification) of the sequential circuit. In addition the algorithms of transfer sequence detection for a set of internal states are used. Reduced Ordered Binary Decision Diagrams (ROBDDs) for the combinational part and its fragments are applied for getting both the estimations and transfer sequences by means of operations on ROBDDs. It is known that these operations have a polynomial complexity. Note that if TCs are inserted out of the working area, they cannot be detected both under verification and testing in the working area. Techniques of masking TCs are proposed. The experimental results on ISCAS and MCNC benchmarks show applicability of the approach. Masking sub-circuits overhead is appreciated.

**Keywords:** sequential circuits; controllability and observability of combinational circuit nodes; State Transition Graph (STG); Malicious circuit (Trojan Circuit); Reduced Ordered Binary Decision Diagram (ROBDD); working area.

The enhanced utilization of outsourcing services for a part of VLSIs (Intellectual Property cores, reprogramming components based on FPGA and so on) to cut VLSI cost increases risk of inserting Trojan Circuits (TCs) that may destroy VLSI or provide leakage of confidential information [1–3]. TCs as a rule act in rare operation situations, therefore they are not detectable neither during VLSI verification nor VLSI testing. TC consists of two parts. Trojan trigger is switched on when the certain combination of signals appears on TC inputs. Trojan payload is operation unit that is switched on by trigger sub-circuit. It is necessary to detect such malicious sub-circuits and, if possible, to mask their actions. It is important to be more precise in finding circuit nodes suitable for inserting TC.

In [4] a vulnerability analysis of circuits is performed at the behavioral level. In this paper a similar analysis is done at the gate level. It means that our proposed technique achieves detection rates that are not affected by synthesis and optimizations.

Design-for-Trust (DFTr) techniques are proposed in [5]. Here prevention schemes based on inserting extra circuitry for obscuring the circuit at different levels of design abstraction making the reverse-engineering at the foundry difficult are presented.

One of solutions is Split Manufacturing [6] that means segregation of fabrication steps among different foundries. These techniques can incorporate reconfigurable logic along with standard logic [7].

In [8], authors proposed an automated low-overhead online methodology to aid in the detection of TCs. They focus on the detection of small TC instances (with less than five logic gates) that cause logic malfunction on activation through rare internal logic conditions. These conditions are determined by using heuristic estimations of controllability. The main advantage of the proposed technique is that the impact of the activated

TC need not propagate all the way to the primary output for the checker to detect it. This feature guarantees that a TC instance is detected as soon as it is triggered, independent of whether the logic malfunction caused by the TC actually propagates to the primary output.

In [9] Functional Analysis for Nearly-unused Circuit Identification (FANCI) tool is suggested. It flags suspicious wires in design, which have the potential to be malicious. FANCI uses approximate Boolean functional analysis to detect these wires.

In this paper in contrast with [8, 9] detection of suspicious nodes is based on using precisely calculated random estimations of controllability and observability of a combinational part internal node. The suggested approach guarantees finding all internal states (compactly represented by ROBDD) that may provide triggering the node. The approach is oriented to a threat model when the designer in a design house is untrusted. The estimations calculations like those in [8, 9] are based on using structural description of the combinational part. In this paper representation of the sequential circuit behavior by State Transition Graph (STG) is additionally used. The calculations are executed with operations on Reduced Ordered Binary Decision Diagrams (ROBDDs, further just BDDs). The algorithms of transfer sequence detection for a set of internal states are also based on using BDD operations and directed toward BDD simplification. Techniques of masking TCs are proposed. The experimental results on benchmarks illustrate applicability of the suggested approach and show that overhead for masking TC may be rather small.

In Section II techniques of precise calculation of controllability and observability estimations for combinational part nodes of a sequential circuit are briefly described. In Section III the way of calculation of precise controllability estimations for combinational part nodes out of working area is given. In Section IV possibilities of detecting transfer sequence for a set of internal states both without finding the sequence itself and with finding one sequence are discussed. In Section V the techniques of masking TC are proposed and experimental results are considered.

## 1. Precise calculation of controllability and observability estimations with using structural combinational part description

1(0)-controllability of an internal node is a possibility of delivering 1(0) value to it, observability is a possibility of observation of changing 1(0) value of an internal node on the proper circuit output. Precise calculation of random controllability and observability estimations is based on using of the corresponding BDDs [10] and operations on them. These estimations are derived for pair of nodes connected with the input and the output of a TC, correspondingly.

To calculate precisely 1(0)-controllability estimation for internal node $v$ [11] of combinational part $C$ we derive BDD $R^{cont}(1)$ ($R^{cont}(0)$) using the combinational circuit which output is pole $v$, and inputs coincide with circuit $C$ inputs. ($R^{cont}(0)$ is obtained from $R^{cont}(1)$ by permutation of terminal nodes.

To calculate precisely observability estimation for internal node $v$ [11] of combinational part $C$ and the proper circuit output we derive first BDD $R(C_v)$ for sub-circuit $C_v$. The sub-circuit corresponds to the proper circuit $C$ output and is obtained from circuit $C$ under the condition that internal node $v$ is an input of sub-circuit $C_v$ [10]. During construction of BDD $R(C_v)$ variable $v$ is chosen as the first variable of the decomposition. It means that BDD $R(C_v)$ root is marked by variable $v$.

Let BDD $R(C_v)$ implements function $f$. We derive from $R(C_v)$) BDDs $R(f^{v=0})$, $R(f^{v=1})$ which roots are children nodes of $R(C_v)$ root. These BDDs implement functions $f^{v=0}$ and $f^{v=1}$ accordingly. Multiplications $R(f^{v=0})\overline{R(f^{v=1})}$, $R(f^{v=1})\overline{R(f^{v=0})}$ are executed and results are merged being represented by BDD $R^{obs}$:

$$R^{obs} = R(f^{v=0})\overline{R(f^{v=1})} \vee R(f^{v=1})\overline{R(f^{v=0})}. \quad (1)$$

Getting $\overline{R(f^{v=0})}$, ($\overline{R(f^{v=1})}$) from $R(f^{v=0})$, ($R(f^{v=1})$) is reduced to permutation of terminal nodes of the corresponding BDDs. Note that BDD operations have a polynomial complexity.

Calculating precise controllability and observability estimations we suppose that 1 value probabilities of all input variables are equal to ½. Using BDDs $R^{cont}(1)$ and $R^{obs}$ we calculate 1 controllability and observability random estimations for node $v$.

Probability $p(\eta)$ of 1 value of Boolean function $\eta$, corresponding to a BDD internal node $\mu$ is calculated with using probabilities $p(\eta_\mu^{x_i=0})$, $p(\eta_\mu^{x_i=1})$ of 1 values of functions $\eta_\mu^{x_i=0}$ and $\eta_\mu^{x_i=1}$, corresponding to children nodes of node $\mu$ in the following way (node $\mu$ is marked by variable $x_i$):

$$p(\eta) = p(x_i)\,p(\eta_\mu^{x_i=1}) + p(\overline{x_i})\,p(\eta_\mu^{x_i=0}).$$

Moving from 1 terminal node of the corresponding BDD with using the above mentioned formula for internal nodes we reach the BDD root. As a result random estimations of 1(0)-controllability or observability are calculated.

Thus random estimations are obtained by using a structural description of a combinational part. But the behavior of this part as a rule is wider than the working area represented by a State Transition Graph (STG). The point is that a TC may be triggered just out of the working area (out of the specification). If we know the STG description (the specification) from which the combinational part of the sequential circuit is obtained, we may calculate precisely random estimations of controllability out of the working area. As for precise random observability estimations they are always calculated by using only structural description of the combinational part.

## 2. Deriving precise controllability estimations out of working area

Let a behavior of a sequential circuit be represented by STG. To derive a sequential circuit we have to encode internal states of STG. As a result we get system of incompletely specified Boolean functions. Changing this system for completely specified Boolean functions system we facilitate possibilities of TCs inserting. The matter is that getting minimized system of completely specified Boolean functions we, as a rule, increase both set-off and set-on area of these functions in comparison with the system of incompletely specified Boolean functions. As a result the full states (depending on input and state variables) that are out of the working area (it is represented by STG) appear. These full states cannot be reachable during sequential circuit verification and testing in the working area. If these full states are used for triggering TC, then we cannot detect TC in above mentioned way. We suggest calculate 1(0)-controllability precise estimations for internal nodes out of the working area.

STG is known to be a description of Finite State Machine behavior in which symbols of input and output alphabets are encoded. Consider an example of STG (Table 1).

Here $x_1$, $x_2$, $x_3$ – input variables of the circuit and $y_1$, $y_2$, $y_3$ – output variables. Columns of the table are derived into 4 sections. The first section represents input cubes (ternary vectors). The second section represents internal states. The third section represents next internal states. The forth section represents output vectors. After encoding internal states by 1-hot code (in our example), we derive system $F$ of incompletely specified Boolean functions (Table 2). The table is also divided into 4 sections. The second and the third sections represent encoded internal states.

Table 1

**State Transition Graph**

| $x_1 x_2 x_3$ | $q$ | $q'$ | $y_1 y_2 y_3$ |
|---|---|---|---|
| 0 – – | 1 | 1 | 0 0 1 |
| – 0 – | 1 | 1 | 0 0 1 |
| 1 1 – | 1 | 2 | 1 0 1 |
| – – 0 | 2 | 2 | 0 1 1 |
| – – 1 | 2 | 3 | 1 1 1 |
| 1 0 – | 3 | 3 | 0 1 0 |
| 0 – – | 3 | 4 | 1 1 0 |
| – 1 – | 3 | 4 | 0 1 1 |
| – – 0 | 4 | 4 | 0 1 1 |
| – – 1 | 4 | 1 | 1 1 1 |

Table 2

**System of incompletely specified Boolean functions**

| $x_1 x_2 x_3$ | $z_1 z_2 z_3 z_4$ | $z_1' z_2' z_3' z_4'$ | $y_1 y_2 y_3$ |
|---|---|---|---|
| 0 − − | 1 0 0 0 | 1 0 0 0 | 0 0 1 |
| − 0 − | 1 0 0 0 | 1 0 0 0 | 0 0 1 |
| 1 1 − | 1 0 0 0 | 0 1 0 0 | 1 0 1 |
| − − 0 | 0 1 0 0 | 0 1 0 0 | 0 1 1 |
| − − 1 | 0 1 0 0 | 0 0 1 0 | 1 1 1 |
| 1 0 − | 0 0 1 0 | 0 0 1 0 | 0 1 0 |
| 0 − − | 0 0 1 0 | 0 0 0 1 | 1 1 0 |
| − 1 − | 0 0 1 0 | 0 0 0 1 | 0 1 1 |
| − − 0 | 0 0 0 1 | 0 0 0 1 | 0 1 1 |
| − − 1 | 0 0 0 1 | 1 0 0 0 | 1 1 1 |

Change symbol «0» in code words of internal states for symbol «−» (don't care). As a result we obtain system $F^*$ of completely specified Boolean functions (Table 3). This minimization is possible because we use the unordered code (1-hot) for encoding of internal states [12−13].

Table 3

**System of completely specified Boolean functions**

| $x_1 x_2 x_3$ | $z_1 z_2 z_3 z_4$ | $z_1' z_2' z_3' z_4'$ | $y_1 y_2 y_3$ |
|---|---|---|---|
| 0 − − | 1 − − − | 1 0 0 0 | 0 0 1 |
| − 0 − | 1 − − − | 1 0 0 0 | 0 0 1 |
| 1 1 − | 1 − − − | 0 1 0 0 | 1 0 1 |
| − − 0 | − 1 − − | 0 1 0 0 | 0 1 1 |
| − − 1 | − 1 − − | 0 0 1 0 | 1 1 1 |
| 1 0 − | − − 1 − | 0 0 1 0 | 0 1 0 |
| 0 − − | − − 1 − | 0 0 0 1 | 1 1 0 |
| − 1 − | − − 1 − | 0 0 0 1 | 0 1 1 |
| − − 0 | − − −1 | 0 0 0 1 | 0 1 1 |
| − − 1 | − − −1 | 1 0 0 0 | 1 1 1 |

The system products represented by cubes of the first and the second sections depend on input and state variables. Columns of the third and the forth sections correspond to functions representing next states and outputs of the sequential circuit. Each function $f^*$ of system $F^*$ is presented by Sum of Products (SoP) originated by cubes of Table 3 marked with 1 values in the column corresponding to the function $f^*$ of this table. Table III is used to derive the sequential circuit comprising from gates. Let $C$ be the combinational part of the sequential circuit obtained and $v$ – internal pole (Fig. 1).

Note that cubes corresponding to the first and the second columns of Table II represent the working area of the sequential circuit. Form the SoP from these cubes. Derive BDD $R^w$ from the SoP. Let $R^{nw}$ be an inversion of $R^w$. Then 1(0)-controllability for node $v$ within working area may be calculated using BDDs:

$$R^{cont\ w}(1) = R^{cont}(1)R^w, \quad (2)$$

$$R^{cont\ w}(0) = R^{cont}(0)R^w, \quad (3)$$

and 1(0)-controllability out of working area may be calculated using BDDs:

$$R^{cont\ nw}(1) = R^{cont}(1)R^{nw}, \quad (4)$$

$$R^{cont\ nw}(0) = R^{cont}(0)R^{nw}. \quad (5)$$

If among internal nodes there exist ones for which 1(0)-controllability estimations seemed less than the threshold given, then these nodes are included into set $V$ of suspicious nodes. If we consider that Trojan Circuits are inserted in working area, we derive random controllability estimations applying BDDs $R^{cont\ w}(1)$ ($R^{cont\ w}(0)$). If we suppose that Trojan Circuits are inserted out of working area, we derive estimations applying $R^{cont\ nw}(1)$ ($R^{cont\ nw}(0)$). If we have only structural description of sequential circuit and know nothing about circuit working area, we derive estimations applying $R^{cont}(1)$ ($R^{cont}$ (0)).

If among internal nodes there exist ones for which the chosen 1(0)-controllability estimations seem less than the threshold given, then these nodes are included into set $V$ of suspicious nodes. But if the chosen controllability for node $v$ is equal to 0, then node $v$ is excluded from further consideration.

We may cut set $V$ using precise estimation of node $v^*$ observability: node $v^*$ is connected with TC output. If the precise estimation of node $v^*$ observability is more than the proper threshold, we exclude corresponding node $v$ from further consideration. We use the observability estimation if we know possible type of TC.

Instead of full observability estimation described above we suggest using partial estimation applying ROBDDs derived by the following multiplications: $R(f^{v=1})\overline{R(f^{v=0})}$, $R(f^{v=0})\overline{R(f^{v=1})}$. The proposed partial estimations of observability for node $v^*$ correspond 0, 1 values of the proper combinational part output.

Note that inserting TC changes values both on pole $v^*$ and the proper output. If partial observability estimation for $v^*$ is equal to 0, we exclude pole $v$ from consideration. Otherwise BDDs used for calculating random estimations for poles $v$, $v^*$ may be applied for evidence of existence of an activating sequence providing malicious TC action and finding the sequence itself if it is necessary.

Node $v$ may be also excluded from consideration if there is no rather short transfer sequence triggering TC with input $v$ and output $v^*$.

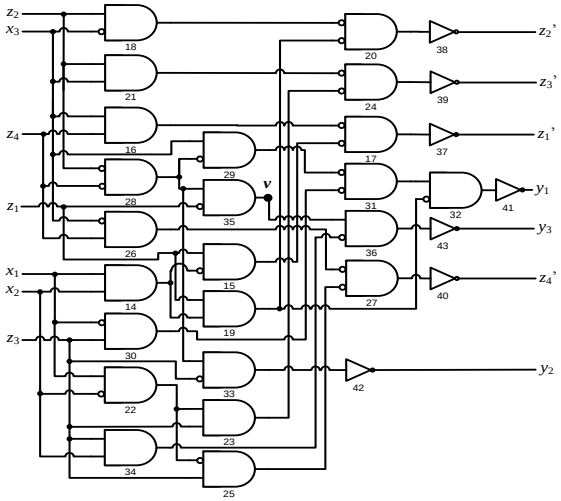
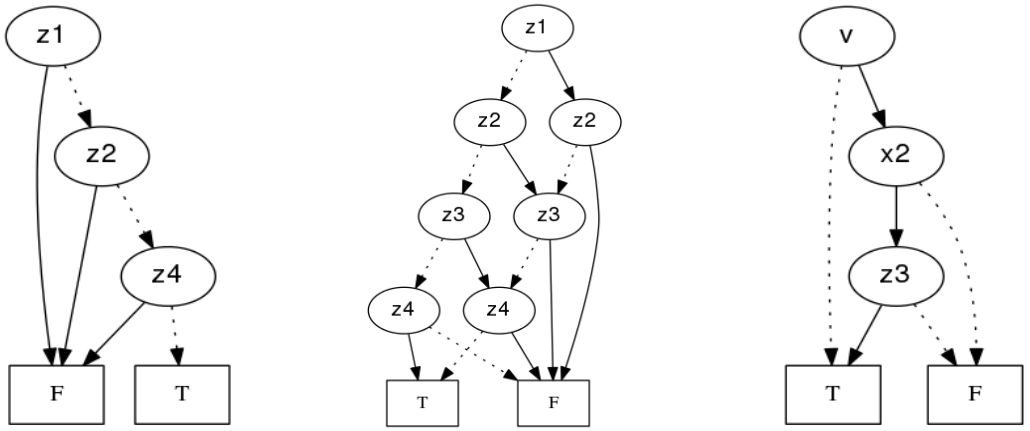


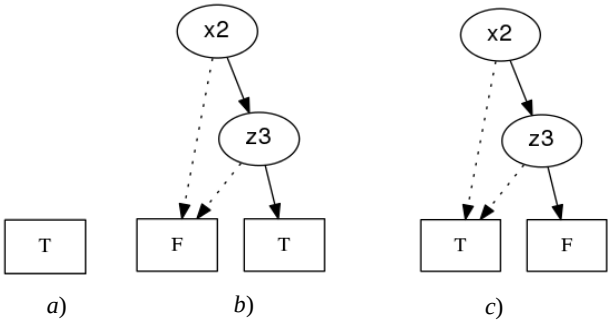

Fig. 1. Combinational circuit $C$ and pole $v$

Illustrate getting the estimations of controllability and observability for node $v$ and output $y_3$ of circuit $C$ (Fig. 1) obtained from Table 3. Construct corresponding BDDs. Let $R_v^{cont}(1)$ be $R^{cont}(1)$ for node $v$ (Fig. 2*a*), $R^w$ be BDD representing working area described by Table 2 (Fig. 2*b*), and BDD $R_{y_3}(C_v)$ be $R(C_v)$ for output $y_3$ (Fig. 2*c*). BDDs $R(f^{v=0})$, $R(f^{v=1})$ are represented by Fig. 3*a*, 3*b*. BDD $R_{v,y_3}^{obs}$ is obtained by formula (1) (Fig. 3*c*).




*a*) *b*) *c*)

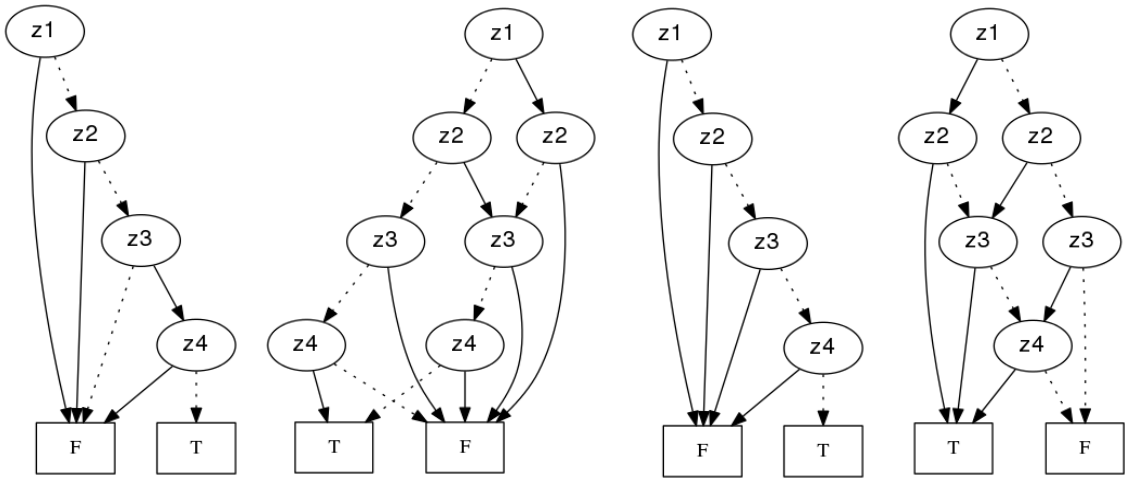
Fig. 2. *a*) BDD $R_v^{cont}(1)$ ; *b*) BDD $R^w$ ; *c*) BDD $R_{y_3}(C_v)$




*a*) *b*) *c*)

Fig. 3. *a*) BDD $R(f^{v=0})$; *b*) BDD $R(f^{v=1})$; *c*) BDD $R^{obs}_{v,y_3}$

BDDs for calculating 1(0)-controllability for node *v* within working area by formulae (2) and (3) are represented on Fig. 4*a*, 4*b*. Similarly BDDs for calculating 1(0)-controllability for node *v* out of working area by formulae (4) and (5) are constructed (Fig. 4*c*, 4*d*).




*a*) *b*) *c*) *d*)

Fig. 4. a) BDD $R_v^{cont\ w}(1)$ ; b) BDD $R_v^{cont\ w}(0)$ ; c) BDD $R_v^{cont\ nw}(1)$ ; d) BDD $R_v^{cont\ nw}(0)$

Using these BDDs we calculate 1(0)-controllability and observability random estimations for node *v* and output $y_3$:

$p(R^{obs}_{v,y_3}) = ½ \cdot 1 + ½ \cdot (½ \cdot 1 + ½ \cdot 0) = 0.75,$

$p(R_v^{cont\ w}(1)) = ½ \cdot (½ \cdot (½ \cdot 0 + ½ \cdot (½ \cdot 1 + ½ \cdot 0)) + ½ \cdot 0) + ½ \cdot 0 = 0.0625,$

$p(R_v^{cont\ w}(0)) = ½ \cdot (½ \cdot (½ \cdot (½ \cdot 0 + ½ \cdot 1) + ½ \cdot 0) + ½ \cdot (½ \cdot (½ \cdot 1 + ½ \cdot 0) + ½ \cdot 0)) + ½ \cdot (½ \cdot (½ \cdot (½ \cdot 1 + + ½ \cdot 0) + ½ \cdot 0) + ½ \cdot 0) = 0.1875,$

$p(R_v^{cont\ nw}(1)) = ½ \cdot (½ \cdot (½ \cdot (½ \cdot 1 + ½ \cdot 0) + ½ \cdot 0) + ½ \cdot 0) + ½ \cdot 0 = 0.0625,$

$p(R_v^{cont\ nw}(0)) = ½ \cdot (½ \cdot (½ \cdot 0 + ½ \cdot (½ \cdot 0 + ½ \cdot 1)) + ½ \cdot (½ \cdot (½ \cdot 0 + ½ \cdot 1) + ½ \cdot 1)) + ½ \cdot (½ \cdot (½ \cdot 0 + + ½ \cdot 1) + ½ \cdot 1) = 0.6875.$

## 3. Finding transfer sequences for a set of internal states

Execute multiplication BDD $R^{cont\ nw}(1)$ or $R^{cont\ nw}(0)$ for node *v* and BDD $R^{obs}$ for node *v**. The multiplication result is represented by BDD $R^f$ . Here we consider that a TC is inserted out of working area (out of specification).

Products originated by paths from $R^f$ root till 1 terminal pole represent sets of full states of sequential circuit. Reaching any state from these sets provides malicious action of TC. Select from $R^f$ a set of internal states and form from them the proper SoP depending on state variables. Derive BDD $R^{S_0}$ from the SoP. The BDD represents a set of internal states so that reaching any state from the set and applying the corresponding input Boolean vector (this vector always exists) provide malicious action TC. The procedure of finding existence evidence of a transfer sequence (the length is not more preset value *l*) for some state from a set presented by BDD $R^{S_0}$ is described in detail in [11]. In this algorithm we did not derive the sequence itself but only set up its existence.

Calculations of controllability and observability estimations for internal nodes of structural combinational part of sequential benchmark circuits (ISCAS'89) are executed. The Table 4 contains the initial information about benchmark circuits: name of benchmark (Circuit), number of inputs (N_Is), number of outputs (N_Os), number of flip-flops (N_FFs) and number of gates (N_Gs).

Table 4

**Benchmark circuits**

| Circuit | N_Is | N_Os | N_FFs | N_Gs |
|---|---|---|---|---|
| s298 | 3 | 6 | 14 | 119 |
| s1196 | 14 | 14 | 18 | 529 |
| s400 | 3 | 6 | 21 | 162 |
| s641 | 35 | 24 | 19 | 380 |
| s1488 | 8 | 19 | 6 | 653 |

Results of calculations of length of the transfer sequences are shown in Table 5. The nodes with the smallest value of controllability (VC) are chosen among all internal nodes. Minimal observability (VO) for output nodes of the corresponding gates that have the smallest value of controllability on input node is calculated.

Here we consider inserting TC when input of the gate is *v* and output of the gate is *v**.

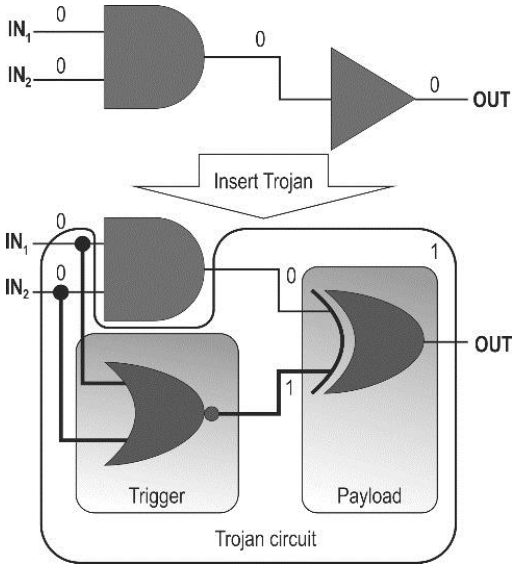



Fig. 5. Inserting malicious sub-circuit

The corresponding gates are candidate places where TCs may be inserted (Fig. 5). For such a gate the set of internal states which can be reached and used for TC activation is constructed and represented by the ROBDD. After that we built transfer sequence with length *l* ($l \le 1000$) for one of the internal states from the proper set. Lengths of transfer sequences are presented in the fifth column (L_S) in Table V. If for nodes *v*, *v** transfer sequences has $l > 1000$ or has no transfer sequences at all, the node *v* has to be removed from suspicious nodes. In Table V we have three such nodes.

Table 5

**Experimental results**

| Circuit | Gate | VC | VO | L_S |
|---|---|---|---|---|
| s298 | n95 | 0.03125 | 0.0664062 | No transfer sequence |
| | n40 | 0.03125 | 0.5 | 7 |
| | n139 | 0.125 | 0.138749 | 9 |
| | n25 | 0.125 | 0.5 | 7 |
| s1196 | n439 | 0.000030516 | 0.249458 | 1 |
| | n438 | 0.000070569 | 0.497288 | No transfer sequence |
| | n335 | 0.000137322 | 0.49707 | 1 |
| s400 | n35 | 0.015625 | 0.5 | 38 |
| | n146 | 0.0156564 | 0.0593154 | 37 |
| | n147 | 0.0309255 | 0.0402537 | 33 |
| | n65 | 0.5044682 | 0.5 | 9 |
| s641 | n491 | 0.000012144 | 0.25 | No transfer sequence |
| | n486 | 0.00245458 | 0.5 | 4 |
| | n489 | 0.00310373 | 0.09375 | 4 |
| s1488 | n589 | 0.00390625 | 0.0494067 | 15 |
| | n98 | 0.00582886 | 0.5 | 4 |
| | n636 | 0.00775146 | 0.5 | 3 |
| | n619 | 0.0078125 | 0.0340039 | 5 |

Then we may find the transfer sequence itself for each node of the obtained set $V$ using algorithm [14]. This algorithm like algorithm in [11] is oriented to cutting calculations but it is more complicate in comparison with algorithm represented in [11]. Applying the derived transfer sequences for set $V$ we may detect node $v$ in which TC is inserted. Based on the result we may mask TC attack.

## 4. Trojan Circuit masking

If we suppose that Trojan Circuit is inserted not out of working are, we may mask it in the following way (Fig. 6).

Here masking sub-circuit together with MUX and XOR are out of sequential circuit area. The sub-circuit implements the same function that the sub-circuit of the combinational part with output $v^*$. When Trojan Circuit is triggered the proper output keeps the correct value.

In the case of injecting Trojan circuit into out of working area we suggest the more simple way of masking (Fig. 7). The masking sub-circuit implements the function represented by BDD $R^f$. Connecting the proper output with MUX we keep the correct behavior of a sequential circuit.

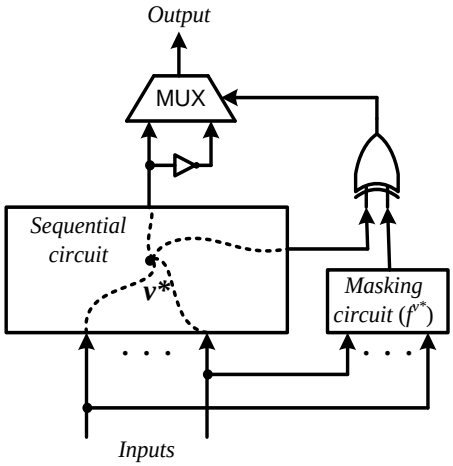



Fig. 6. Masking TC scheme

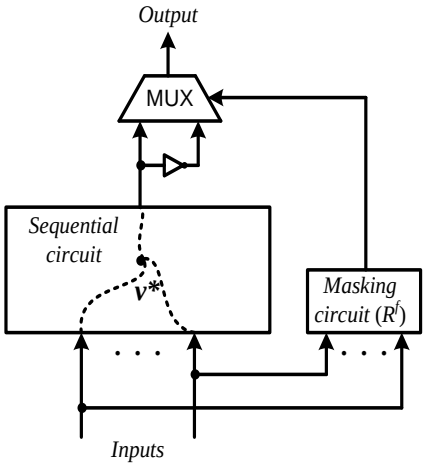



Fig. 7. Masking TC scheme inserted into out of working area

In this case we need STG description of the sequential circuit behavior and so we use MCNC [15] sequential benchmark circuits in KISS2 format for experiments.

The set of circuits has been made from KISS2 format (from STG description) by 1-hot encoding of states and using a logic synthesis and optimization in ABC system [16].

For experiments we have limited to TCs which can be inserted into internal nodes with low controllability estimations without taking into consideration observability estimations. This approach is suited for any type of TC. When we know the type, we may use more simple BDDs $R^f$ and consequently to cut overhead.

Experiments show that for each internal node with low controllability there exists rather short transfer sequence [11] triggering TC. For the benchmark circuits considered the transfer sequence lengths are not more than 8 (in average 1.1).

Calculations of controllability estimations for internal nodes of combinational part of sequential circuits and the estimations out of working area are represented in Tables 6, 7. In these tables overhead estimations of masking sub-circuits corresponding to 10 nodes with lesser controllability estimations for each circuit are also presented. There are the names of benchmarks (Circuits), numbers of gates (N_Gs), minimum nonzero values of controllability estimations (Min_VC), parts of gates (their output nodes) with values of controllability less or equal to 0.05 (%_Gs$^1$), 0.005 (%_Gs$^2$) and 0.0005 (%_Gs$^3$) in percentage, sizes of minimum masking sub-circuits as a percentage from initial circuit (%_Min) and sizes of maximum masking sub-circuits as a percentage from initial circuit (%_Max) for 10 internal nodes with lesser controllability estimations.

Benchmark circuits and masking sub-circuits are received in ABC and they consist of 2-input logic-gates.

Table 6

**Experimental results for TC in working area**

| Circuit | N_Gs | Min_VC | %_Gs$^1$ | %_Gs$^2$ | %_Gs$^3$ | %_Min | %_Max |
|---|---|---|---|---|---|---|---|
| cse | 145 | 0.0000305176 | 17.2 | 1.4 | 0.7 | 3.4 | 11.0 |
| dk14 | 102 | 0.03125 | 2.9 | 0.0 | 0.0 | 1.0 | 11.8 |
| dk16 | 142 | 0.015625 | 7.0 | 0.0 | 0.0 | 2.1 | 16.2 |
| ex1 | 176 | 0.0000305176 | 14.2 | 4.5 | 2.8 | 0.6 | 10.8 |
| keyb | 193 | 0.00138255 | 16.1 | 1.6 | 0.0 | 1.0 | 33.2 |
| kirkman | 126 | 0.0000305176 | 10.3 | 1.6 | 0.8 | 0.8 | 30.2 |
| sand | 388 | 0.000000159256 | 10.1 | 2.8 | 0.8 | 0.3 | 12.9 |
| sse | 88 | 0.000731945 | 10.2 | 1.1 | 0.0 | 2.3 | 23.9 |
| styr | 305 | 0.000000953674 | 16.7 | 2.6 | 2.0 | 2.0 | 15.4 |
| tbk | 669 | 0.000000000232831 | 21.5 | 3.9 | 2.2 | 0.4 | 6.0 |
| train11 | 44 | 0.03125 | 2.3 | 0.0 | 0.0 | 6.8 | 13.6 |

Table 7

**Experimental results for TC out of working area**

| Circuit | N_Gs | Min_VC | %_Gs$^1$ | %_Gs$^2$ | %_Gs$^3$ | %_Min | %_Max |
|---|---|---|---|---|---|---|---|
| cse | 145 | 0.000000238419 | 54.5 | 54.5 | 54.5 | 0.7 | 26.9 |
| dk14 | 102 | 0.000976562 | 58.8 | 34.3 | 0.0 | 2.0 | 14.7 |
| dk16 | 142 | 0.00000000186265 | 55.6 | 55.6 | 55.6 | 0.7 | 6.3 |
| ex1 | 176 | 0.000000178814 | 76.7 | 38.6 | 19.3 | 0.6 | 5.1 |
| keyb | 193 | 0.0000000596046 | 58.5 | 58.5 | 58.5 | 0.5 | 9.8 |
| kirkman | 126 | 0.000000476837 | 17.5 | 4.0 | 3.2 | 0.8 | 19.0 |
| sand | 388 | 0.00000000000909495 | 52.8 | 52.8 | 52.8 | 0.3 | 1.5 |
| sse | 88 | 0.0000038147 | 51.1 | 51.1 | 51.1 | 1.1 | 8.0 |
| styr | 305 | 0.0000000000218279 | 58.4 | 58.4 | 58.4 | 0.3 | 11.5 |
| tbk | 669 | 0.00000000000363798 | 60.7 | 60.7 | 60.7 | 0.1 | 23.8 |
| train11 | 44 | 0.000488281 | 59.1 | 15.9 | 2.3 | 4.5 | 13.6 |

Masking TC with using out of working area requires as a rule smaller overhead (in average from 1.1% to 12.8%) in comparison with masking TC with using only structural description of a combinational part (in average from 1.9% to 16.8%).

## Conclusion

Possibilities of triggering TC are examined. The investigation is based on getting precise estimations of internal node controllability and observability by using structural combinational part description. The methods of getting precise estimations may be used for comparison with results of the different heuristic methods. The approach to TCs detection inserted out of working area may be applied when they are not detectable during sequential circuit verification and testing in working area. The experiments on benchmarks show applicability of the suggested approach. The techniques of masking TCs are proposed. Masking circuits overhead for chosen internal nodes of the benchmarks are acceptable.

REFERENCES


1. Karri, R., Rajendran, J., Rosenfeld, K. & Tehranipoor, M. (2010) Trustworthy Hardware: Identifying and Classifying Hardware Trojans. *Computer*. 43(10). pp. 39–46. DOI: 10.1109/MC.2010.299
2. Salmani, H., Tehranipoor, M. & Plusquellic, J. (2012) A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time. *IEEE Trans. Very Large Scale Integr. VLSI Syst.* 20(1). pp. 112–125. DOI: 10.1109/TVLSI.2010.2093547
3. Dupuis, S., Ba, P.S., Natale, G.D., Flottes, M.L. & Rouzeyre, B. (2014) A novel hardware logic encryption technique for thwarting illegal overproduction and Hardware Trojans. *IEEE 20th International On-Line Testing Symposium (IOLTS)*. pp. 49–54. DOI: 10.1109/IOLTS.2014.6873671
4. Imeson, F., Emtenan, A., Garg, S. & Tripunitara, M.V. (2013) Securing Computer Hardware Using 3D Integrated Circuit (IC) Technology and Split Manufacturing for Obfuscation. *Proc. of the 22Nd USENIX Conference on Security*. Berkeley,CA. pp. 495–510.
5. Liu, B. & Wang, B. (2014) Embedded reconfigurable logic for ASIC design obfuscation against supply chain attacks. *Design, Automation Test in Europe Conference Exhibition (DATE)*. pp. 1–6. DOI: 10.1109/IOLTS.2014.6873671
6. Chakraborty, R.S., Pagliarini, S., Mathew, J., Ranjani, R.S. & Devi, M.N. (2017) A Flexible Online Checking Technique to Enhance Hardware Trojan Horse Detectability by Reliability Analysis. *IEEE Trans. Emerg. Top. Comput.* 5(2). pp. 260–270. DOI: 10.1109/TETC.2017.2654268
7. Waksman, A., Suozzo, M. & Sethumadhavan, S. (2013) FANCI: Identification of Stealthy Malicious Logic Using Boolean Functional Analysis. *Proc. of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. pp. 697–708. DOI: 10.1109/SP.2011.32.
8. Matrosova, A., Ostanin, S. & Kirienko, I. (2014) 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. *14th Biennial Baltic Electronic Conference (BEC)*. pp. 85–88.
9. Matrosova, A.Y., Kirienko, I.E., Tomkov, V.V. & Miryutov, A.A. (2016) Reliability of Physical Systems: Detection of Malicious Subcircuits (Trojan Circuits) in Sequential Circuits. *Russian Physics Journal.* 59(8). pp. 1281–1288. DOI: 10.1007/s11182-016-0903-8
10. Busaba, F.Y. & Lala, P.K. (1994) Self-checking combinational circuit design for single and unidirectional multibit error. *Journal of Electronic Testing.* 5(1). pp. 19–28. DOI: 10.1007/BF00971960
11. Matrosova, A. & Ostanin, S. (2000) Self-checking FSM design with observing only FSM outputs. *International Journal of VLSI Design*. pp. 153–154. DOI: 10.1109/OLT.2000.856629
12. Matrosova, A., Andreeva, V. & Melnikov, A. (2016) ROBDDs application for finding the shortest transfer sequence of sequential circuit or only revealing existence of this sequence without deriving the sequence itself. *IEEE East-West Design Test Symposium (EWDTS)*. pp. 1–4.
13. Yang, S. (1991) *Logic Synthesis and Optimization Benchmarks User Guide Version 3.0*.
14. *ABC: A System for Sequential Synthesis and Verification*. [Online] Available from: http://www.eecs.berkeley.edu/~alanmi/abc/.



***Matrjsova Anjela Yurievna,*** Dr. Sc., Professor. E-mail: mau11@yandex.ru
***Mitrofanov Evgeny Vladimirovich.*** E-mail: gvaz@yandex.ru
***Ostanin Sergey Alexandrovich,*** Cand. Sc., Associate Professor. E-mail: sergeiostanin@yandex.ru
***Butorina Nataly Borisovna.*** E-mail: nnatta07@mail.ru
***Pahomova Elena Grigorievna,*** Cand. Sc., Associate Professor. E-mail: peg@tpu.ru
***Shulga Sergey Anatolievich.*** E-mail: shsa@me.com
National Research Tomsk State University



Поступила в редакцию 10 ноября 2017 г.



***Матросова Анжела Ю., Митрофанов Евгений В., Останин Сергей А., Буторина Наталья Б., Пахомова Елена Г., Шульга Сергей А.*** (Томский государственный университет, Российская федерация).


**Обнаружение и маскирование вредоносных подсхем в последовательностных схемах.**


**Ключевые слова:** последовательностные схемы; управляемость и наблюдаемость полюсов комбинационной схемы; вредоносные подсхемы; сокращенные упорядоченные двоичные диаграммы решений (ROBDD); рабочая область.

DOI: 10.17223/19988605/42/10

Внедрение вредоносных подсхем (Trojan Circuits), которые могут разрушить логическую схему или обеспечить утечку конфиденциальной информации из системы, содержащей логическую схему, требует обнаружения таких подсхем и, если возможно, их маскирования. Мы предлагаем метод поиска множества полюсов последовательностной схемы, в которые могут быть вставлены вредоносные подсхемы. После моделирования последовательностной схемы на корректных входных последовательностях мы можем обнаружить вредоносную подсхему, если она присутствует, и замаскировать ее специальной подсхемой. Метод основан на применении точных (не эвристических) оценок управляемости и наблюдаемости внутренних полюсов, полученных с использованием структурного описания комбинационной составляющей последовательностной схемы. Эти оценки вычисляются с использованием микропрограммного описания автомата (STG) в предположении, что вредоносная подсхема может быть вставлена вне рабочей области функционирования последовательнстной схемы, определенной спецификацией. Также используются алгоритмы поиска установочной последовательности для множества внутренних состояний. Для получения оценок управляемости и наблюдаемости, а также поиска усановочной последовательности используются сокращенные упорядоченные двоичные диаграммы решений (ROBDD-графы). Известно, что операции над ROBDD-графами имеют полиномиальную сложность. Следует учесть, что если вредоносные подсхемы вставлены вне рабочей области функционирования, то они не могут быть обнаружены как при верификации, так и при тестировании в рабочей области функционирования. Предложены методы маскирования вредоносных подсхем.