This article has been accepted and published on J-STAGE in advance of copyediting. Content is final as presented.

IEICE Electronics Express, Vol.VV, No.NN, 1–6

LETTER

# Holistic hardware Trojan design of trigger and payload at gate level with rare switching signals eliminated

**Kai Huang[1], Yun He[1a)], and Xiaowen Jiang[1]**

**Abstract** A unified trigger and payload design scheme was proposed and re-convergent logic was introduced to eliminate rare transition signals which may be taken as suspicious signals in existing Trojan detections. Two Trojan structure templates were proposed and they could be applied to both privilege promotion and deny-of-service attacks. By combining these two structures in different proportion and position, a Trojan benchmark generation algorithm was proposed in which Trojan variations were resistant to feature analysis based detections. The proposed Trojans can obtain a very low activation probability using only primary inputs, which can reduce the restriction on primary trigger signals. So the Trojans will have better operability and adaptability in Trojan insertion. Then we discussed the method to keep concealed in fault diagnostic. At last, we made a comparison between the proposed Trojans and the state-of-the-art Trojan benchmarks on structural and logical features.

**key words:** Activation probability, hardware Trojan design, SCOAP, transition probability

**Classification:** Integrated circuits

## 1. Introduction

With the outsourcing of design and fabrication in integrated circuit (IC) industry, the high risk of malicious inclusions [1, 2] challenge the whole system, which is the so-called hardware Trojans. Theoretically hardware Trojans can be inserted in any phase of the IC design flow, from specification to fabrication and assembly phase [3, 4, 5]. We classify these Trojans into three attack models, third party intellectual property (3PIP) attack, interior attack and foundry attack. Trojans inserted without golden design available for verification are categorized as 3PIP Trojan attack, such as Trojans inserted by untrusted IP providers. Foundry attack refers to Trojans mounted in fabrication process, which will cause there exists no golden chip available for Trojan detection, and the RTL or gate-level netlist is supposed to be Trojan-free. Interior attack includes Trojans inserted between design and manufacture and reference model is available in some format, such as untrusted CAD tools or internal designer insertion.

[1]Institute of VLSI Design, Zhejiang University, Hangzhou, 310027, China

a) yun.ee@zju.edu.cn

**DOI: 10.1587/elex.16.20190431**
**Received July 3, 2019**
**Accepted July 24, 2019**
**Publicized August 5, 2019**

For Trojan research, many Trojan benchmarks have been proposed [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. To keep dormant in functional verification, hardware Trojan is usually triggered by some rare conditions, which is the so-called trigger condition. In the existing Trojan benchmarks, the trigger part and payload part are designed separately and the rare-triggered condition is implemented by a combination of several low-activity or low testability signals, which may be detected by logic analysis techniques [17, 18, 19, 20]. The traditional digital Trojan design methodology can be divided into two categories, the first exploits the rare transition signals in the original design [6, 7, 9, 15], and the second attempts to partition the Trojan circuit into smaller parts and stages [12, 14]. A uniform distribution of states is acquired by linear feedback shift register in [8], but low transition wire is still created when generating rare condition by combinational circuits. All these Trojans implement their rare conditions rely on one or more low activity or low testability signals, which may be detected by logic analysis detections, such as unused logic analysis [21, 22], activation based analysis [23, 24, 25], Sandia Controllability/Observability based analysis [17, 18] and probability analysis [19, 20].

In this paper, we propose a holistic Trojan design of trigger and payload, where the transition probability and testability of all signals are close to that of normal ones to avoid being labeled as suspicious signals. The malicious function still keeps a very low probability of being triggered without prior knowledge. The main contributions of our work includes: 1) We propose a holistic Trojan design of trigger and payload to eliminate the rare switching signals in Trojan circuits, and two schemes of Trojan structures have been proposed by mixing trigger signals with payload signals. 2) Based on these structures, we propose a Trojan benchmark generation method by combining these structures in different proportion and position to obtain resistance against feature analysis. 3) The proposed trojan can obtain low activation probability using only normal signals and produce no rare switching signals, which can reduce restrictions on primary trigger signals and increase operability and adaptability of Trojans, and avoid being detected by probability and testability based analysis.



Copyright © 2019 The Institute of Electronics, Information and Communication Engineers

## 2. Proposed Trojan design schemes

For privilege promotion Trojans, the function can be represented by $f = f_n + C_m$, $f_n$ and $C_m$ represent the normal and malicious function respectively. Fig. 1*a* illustrates the widely used digital trigger circuit in existing benchmarks. Fig. 1*b* depicts the Trojan model proposed in [8]. The stealthiness of trigger circuit is implemented by a combination of rare switching signals, such as $C_m = C_{m1}C_{m2}, \cdots, C_{mk}$. Since the trigger signals are generated first and then coupled with payload signal, there are always some signals, e.g. $Trigger$ and $O$, in the Trojan circuit that stuck at one value most of the time which will be classified as Trojan signals by detection techniques such as [17, 18, 19].

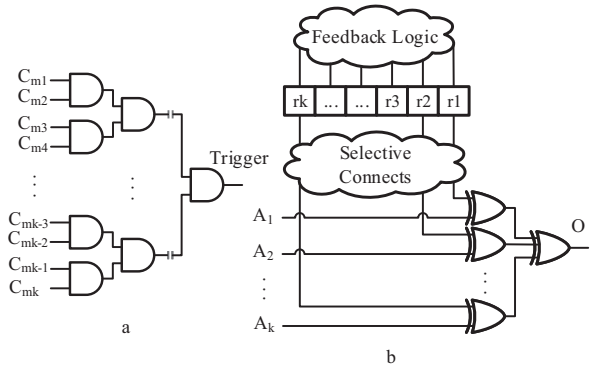


**Fig. 1.** Trigger part of Trojan benchmarks in recent literature: a. Trigger circuit structure widely used in [7, 9]; b. XOR-LFSR structure in [8]

### 2.1 Scheme 1 - product of sum (POS)

To avoid this detectable stuck-at feature, we first introduce reconvergency and rewrite the function as follows:

$$\begin{aligned} f &= f_n + C_m = (f_n + C_{m1})(f_n + C_{m2}) \\ &= (f_n + C_{m1})(f_n + C_{m2}) \cdots (f_n + C_{mk}) \\ &= f_n + C_{m1}C_{m2} \cdots C_{mk} \end{aligned} \tag{1}$$

Two corresponding circuit constructions are depicted in Fig. 2*c* and Fig. 2*d*. The Trojan is activated when $C_m = C_{m1}C_{m2}, \cdots, C_{mk} = 1$ and wire $f$ is changed from $f_n$ to 1. In the Trojan circuit, trigger signals $C_{m1}C_{m2}, \ldots, C_{mk}$ are coupled with victim signal $f_n$ respectively. Fig. 2*a* shows the traditional implementation with rare transition signal $h_1$ generated. Fig. 2*b* illustrates the proposed POS design where all internal signals $h_1, h_2$ have normal transition probability close to $f_n$. Fig. 2*c* and Fig. 2*d* show that malicious conditions $C_{m1}, C_{m2}$ can be extended and bottom-up and top-down structure of POS implementation are exhibited respectively. The bottom-up structure implements all conditions $(f_n + C_{mi})$ first and the top-down implementation integrates conditions iteratively.

### 2.2 Scheme 2 - sum of product (SOP)

Another scheme to integrate trigger signals into payload signal is logic rearrange. We have the following transformation,

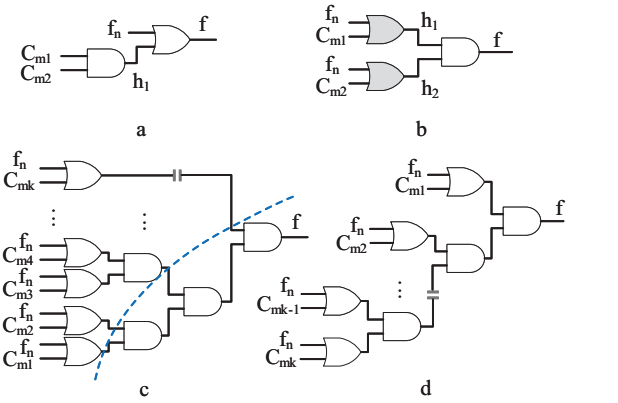


**Fig. 2.** Proposed product of sum (POS) implementation: a. Traditional implementation; b. Proposed POS implementation unit; c. Bottom-up implementation structure; d. Top-down implementation structure

$$\begin{aligned} f &= f_n + C_m = C_{m1}(f_n + C_{m2}) + \overline{C_{m1}} f_n \\ &= C_{m1}[C_{m2}(f_n + C_{m3}) + \overline{C_{m2}} f_n] + \overline{C_{m1}} f_n \\ &= C_{m1}[C_{m2}[\cdots[C_{mk-1}(f_n + C_{mk}) + \overline{C_{mk-1}} f_n]\cdots] + \overline{C_{m2}} f_n] + \overline{C_{m1}} f_n \\ &= f_n + C_{m1}C_{m2} \cdots C_{mk} \end{aligned} \tag{2}$$

The corresponding circuit is depicted in Fig. 3. All internal signals $h_1, h_2, \ldots, h_j$ in the proposed SOP Trojan circuit have normal transition probability near to normal signals. Fig. 3*a* shows an example of one stage implementation. Fig. 3*b* shows that the AND-gate and OR-gate occurs iteratively in Trojan circuit, which exhibits intrinsic resistance to structural feature analysis detections.

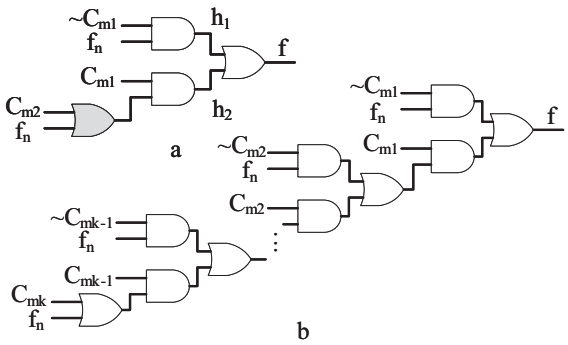


**Fig. 3.** Proposed sum of product (SOP) implementation: a. Proposed SOP implementation unit; b. Full structure example

### 2.3 Deny of service Trojans

For digital triggered Trojans, there is another attack model, the deny-of-service Trojans, which can be represented as

$$\begin{aligned} f &= f_n \cdot \overline{C_m} = \overline{\overline{f_n} + C_m} \\ &= \overline{\overline{f_n} + C_{m1}C_{m2} \cdots C_{mk}} \\ &= \overline{(\overline{f_n} + C_{m1})(\overline{f_n} + C_{m2}) \cdots (\overline{f_n} + C_{mk})} \end{aligned} \tag{3}$$

In this model, the on-set shrunk under malicious condition. To keep dormant in functional verification, $C_m$ should be a rare condition and $\overline{C_m}$ near to 1 to keep $f = f_n$ in most cases. As expressed in (3), we can use the proposed privilege promotion schemes (SOP and POS) to implement the deny-of-service Trojans. The circuit change the function result from $f_n$ to 0 only when the rare condition $C_m = C_{m1}C_{m2}, \cdots, C_{mk} = 1$ is satisfied. So the proposed structures can be applied to both privilege promotion and deny-of-service Trojans without suspicious signals produced.

## 3. Resistance to the state-of-the-art detections

### 3.1 Activation probability

Activation probability is a basic evaluation criterion for Trojan stealthiness analysis in activation based detections, such as functional verification. Trigger activation probability reflects the difficulty of generating the trigger condition without prior knowledge. Trojan activation usually means the Trojan effect has been injected into the original circuit. The difference between trigger activation and Trojan activation lies in the dependency between trigger and payload signals. For simplicity, we suppose that trigger signal and payload signal are independent and Trojan circuit will make some difference to the original circuit when trigger condition is satisfied. For the Trojan structure proposed above, since the rare condition $C_m = C_{m1}C_{m2}, \cdots, C_{mk}$, for simplicity we suppose the conditions are independent and each condition $C_{mi}$ represents a signal, the activation probability can be calculated as,

$$P_m(1) = \prod P_{mi}(1), i = 1, \ldots, k \tag{4}$$

So it is easy to build a stealthy Trojan with extremely low activation probability using our Trojan structure. For simplicity, we take primary inputs as example for first-stage trigger signals, and we suppose all input signals have probability $P(0) = P(1) = 0.5$, so the activation probability can be rewrote as $P_m = (\frac{1}{2})^k$. If we set $k = 32$ in (1) and (2), the Trojan circuit in Fig. 2$d$ and Fig. 3$b$ will have activation probability of $(\frac{1}{2})^{32}$ which is stealthy enough to keep dormant in functional verification without prior knowledge of Trojan activation pattern.

### 3.2 Signal transition probability analysis

The signal probability or transition probability reflect the stuck-at feature of signals. There have been many detections utilizing this signal probability as Trojan feature, where Trojan signals are supposed to have low signal probability or transition probability. And excitation based detections also try to excite these rare switching signals. The 1-probability of internal signals of SOP and POS schemes in Fig. 2$c, d$ and Fig. 3$b$ can be calculated as,

$$P(1) = P_{f_n}(1) + P_{f_n}(0) \prod P_{C_{mj}}(1), j \in [1, k] \tag{5}$$

Where $C_{mj}$ refer to the signals contained in the fan-in cones. So the 1-probability falls into the range of $[P_{f_n}(1), P_{f_n}(1) + P_{f_n}(0)P_{C_{m1}}(1)]$. For example, if we use primary inputs as first-stage trigger signals $C_{m1}C_{m2}, \ldots, C_{mk}$ and we suppose the probability of these signals and $f_n$ are $P(0) = P(1) = 0.5$, the probability of internal signals can be represented as,

$$P(1) = 0.5 + (0.5)^{k+1} \tag{6}$$

Where k is the number of trigger signals contained in its fan-in signals. So the 1-probability of all signals in the Trojan circuit fall into the range $[0.5, 0.75]$. So there is no signal will be regarded as suspicious one in signal transition probability based detections since all signals flip as normal ones without being activated.

### 3.3 Testability analysis

Sandia combinational controllability reflects the difficulty of setting a signal line to a required logic value from primary inputs and combinational observability reflects the difficulty of propagating the logic value of the signal line to primary outputs [26]. Trojan signals are assumed to have high controllability/observability values in detections such as [17, 18]. The controllability and observability in both SOP and POS keep in normal range since the first-stage trigger signals of $C_{mi}$ can be any testable wires such as primary inputs, and the detailed values are present in Table II assuming that $C_{mi}$ have $CC0 = CC1 = 1$. Part of the calculation rules for combinational controllability are listed in Table I and complete rules can be found in [26]. To keep consistent with Synopsys TetraMAX, the "1" added to each rule indicating that a signal pass through one more level of logic gate is ignored, since it is negligible compared to the major part.

**Table I.** Sandia combinational Controllability Calculation Rules

| | $CC0$ | $CC1$ |
|---|---|---|
| AND | $min\{input\ CC0\} + 1$ | $\sum(input\ CC1) + 1$ |
| OR | $\sum(input\ CC0) + 1$ | $min\{input\ CC1\} + 1$ |
| NOT | $input\ CC1 + 1$ | $input\ CC0 + 1$ |

### 3.4 Structural feature analysis

For the simple structures in SOP and POS, the Trojan circuits may be tracked by their structural pattern in feature analysis based detections [27, 28, 29, 30], especially the bottom-up implementation of POS (POS-BU) where exists an AND-logic group. So we propose mixing these two structures showed in Fig. 2$b$ and Fig. 3$a$ in different percentage and position. Fig. 2$b$ and Fig. 3$a$ are the minimum unit of POS and SOP respectively, and the OR-gate in grey in each unit can be replaced with another unit. Fig. 2$cd$ and Fig. 3$b$ show three examples of OR-gate replaced by same kind of unit and Fig. 2$cd$ show two structures replaced in different position. Fig. 4 shows one heterogeneous structure of

$f = f_n + C_{m1}C_{m2}\cdots C_{m6}$ implemented by a combination of POS and SOP units. The full design method is described in Algorithm 1, and more than $2^{n-1}$ variations can be constructed. The activation probability and signal probability keep the same as SOP and POS since the function keeps the same. Sandia controllability and observability values fall between SOP and POS structures. Since SOP and POS can be mixed together in different proportion and position, and registers can be inserted, the variations show different structures and AND-gate and OR-gate distribute in a much normal way.

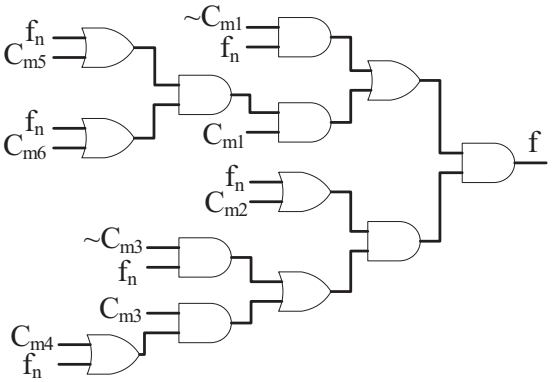


**Fig. 4.** Implementation example of $f = f_n + C_{m1}C_{m2}\cdots C_{m6}$

## 4. Implementation and evaluation

4.1 Trojan benchmarks generation algorithm

Based on the two schemes, SOP and POS, we propose to generate Trojan benchmarks by combining the two structures in Fig. 2*b*, and Fig. 3*a* through replacing the grey OR-gate with a new unit to construct more Trojan variations by changing the proportion and position of the two structures. Fig. 2*cd*, Fig. 3*b* and Fig. 4 show some examples of constructed Trojan circuits. Algorithm 1 shows the full flow of generating benchmarks based on required activation probability and randomly selected trigger signals.

4.2 Trojan benchmarks evaluation

We insert these Trojans into benchmarks on Trust-HUB [16] such as *s*38417 and *vga_lcd*, and replace the original Trojans with our generated ones. Random simulation within $3.2 \times 10^6$ cycles, and simulation with ATPG test patterns generated by Synopsys TetraMAX are conducted on the generated Trojan circuits. All the benchmarks keep dormant in random simulation and no activation test pattern were generated after registers have been inserted. To establish a complete conception of our Trojans, comparison to existing Trojan benchmarks is made in table II. All flip-flops are not considered in the comparison since flip-flops can be inserted in all benchmarks.

The evaluation is based on obtaining same activation probability of $(\frac{1}{2})^n$. For Trojans utilizing original rare switching signals [7, 9], the evaluation is assessed in their original benchmarks. For evaluation of Trojan construction methods, such as POS, SOP and [12, 14], the evaluation is dependent on first-stage trigger signals and payload signal instead of benchmarks. The first-stage trigger signals are supposed to be normal ones with $CC0 = CC1 = 1, P(0) = P(1) = 0.5$ and payload signals with $CO = 0$ since no special signals are required in POS, SOP and [12, 14]. For POS and SOP, as explained in (6), the signal probability is controlled in $0.5 \sim 0.75$, so the minimum probability of SOP and POS is 0.25, while traditional implementation in [12, 14] and [7, 9] will generate a rare transition trigger signal with probability $(\frac{1}{2})^n$. The controllability and observability of SOP and POS increase in a linear way the same as [12, 14], which is normal way in Trojan-free circuits. The relatively small combinational controllability and observability values will cause high false positive and false negative in controllability based detections [17, 18]. So the proposed Trojan circuits

**Algorithm 1:** Trojan benchmark generation

Required activation probability $\theta$;
Trigger signals set $\mathbb{C} = \{C_1, C_2, \ldots, C_n\}$, $\prod_{i=1}^{n} P_{C_i} \leq \theta$;
Signals $L = \{c1, c2\} = pop_2(\mathbb{C})$, *structure* $S = rand\{SOP, POS\}$;
**while** $\mathbb{C} \neq \varnothing$ **do**
  **if** $S == SOP$ **then**
    $L = \{c1, c3\} = pop(\mathbb{C}), S = rand\{SOP, POS\}$;
  **else**
    $S_1 = rand\{SOP, POS, \phi\}$;
    **if** $S_1 \neq \phi$ **then**
      $L_1 = \{c1, c3\} = pop(\mathbb{C})$;
      $S_2 = rand\{SOP, POS, \phi\}$;
      **if** $S_2 \neq \phi$ **then**
        $L_2 = \{c2, c4\} = pop(\mathbb{C})$;
      **else**
        $L_2 = \varnothing$;
      **end**
    **else**
      $L_1 = \varnothing$;
      $S_2 = rand\{SOP, POS\}$;
      $L_2 = \{c2, c4\} = pop(\mathbb{C})$;
    **end**
    $L = \{L_1, L_2\}, S = (S_1, S_2)$;
  **end**
**end**

**Table II.** Comparison of the state-of-the-art digital Trojans

| Features | POS | SOP | [12, 14] | [7, 9] |
|---|---|---|---|---|
| Activation probability | $(\frac{1}{2})^n$ | $(\frac{1}{2})^n$ | $(\frac{1}{2})^n$ | $(\frac{1}{2})^n$ |
| Minimum probability | 0.25 | 0.25 | $(\frac{1}{2})^n$ | $(\frac{1}{2})^n$ |
| Maximum controllability | $n$ | 3 | $n$ | Large |
| Maximum observability | $n$ | $2n$ | $n$ | Large |
| Operability | Easy | Easy | Easy | Hard |
| Overhead/Gates | $2n$ | $3n$ | $n$ | $\leq n$ |

have better logical features near to normal circuits to avoid being detected. Operability represents the difficulty to activate the Trojan by the attacker, which is important for practical consideration. Since traditional Trojans [7, 9] exploit the rare switching signals as the first stage trigger signals to acquire certain low activation probability, it is difficult to make sure that the trigger circuit is controllable. Since our proposed Trojans can obtain same activation probability with normal signals even primary inputs, so they will have better operability. Since traditional Trojan exploits the original signals in Trojan-free circuits, smaller area overhead will be consumed.

## 5. Discussion and Conclusion

Excitation techniques used in fault diagnostic such as stuck-at fault detection, may be utilized to generate test vectors for functional simulation. In the simple implementation of Trojans shown in Fig. 2 and Fig. 3, for stuck-at fault sensitization and path sensitization of signal $C_{mj}$, all other trigger signals $C_{mi}, i \neq j$ should be set to 1, so the activation test pattern will be covered. To confuse the auto test pattern generation program (ATPG), we insert flip-flops into the combinational structure mentioned above to provide extra propagation path. The registers can be placed exactly in the retiming line in Fig. 2*c* and Fig. 5 marked by dash line without changing the functionality, or any other cut set of the circuit according to retiming techniques. The registers should be embedded into the scan chain to provide path sensitization for fault detection so all the signals can switch through the scan chain without activating the trigger condition.

The proposed Trojan structures and benchmarks are especially suitable for 3PIP attack or interior attack model, since they can eliminate the rare switching signals, and no golden chip can be utilized for side channel information based detections. If re-synthesis can be conducted on the Trojan-inserted circuits, the combination-only structure may be degraded to normal structure as Fig. 1*a*. To avoid degeneration, registers can be inserted, and Fig. 5 shows an example of registers inserted which can prevent the structure from degeneration. The dashes represent where registers should be inserted, and the function keeps unchanged according to retiming rules. The registers inserted will prevent synthesis tools from combining logic before and after registers together. Although the logic after registers may be optimized, the transition probability will keep same as (5) since $f_n$ has been coupled with all signals $C_{mi}$ since the re-convergent logic before registers will not be optimized. This can be applied in both SOP and POS in Fig. 2 and Fig. 3.

To the best of our knowledge, we are the first to propose unified trigger and payload design in this paper, and we introduce re-convergent logic to eliminate rare switching signals. We have proposed two structures which can eliminate the rare switching feature of signals for Trojan construction. By combining these structures in different proportion and position, a large number of variations can be generated

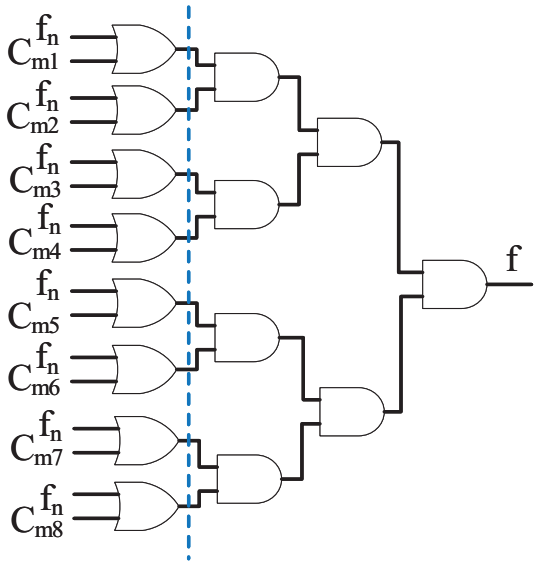


**Fig. 5.** Registers inserted to avoid degradation

which have no suspicious structural features can be tracked in feature analysis detections by using our Trojan benchmark construction method. The retiming technique has been introduced to provide extra propagation path for test pattern generation, which can cause the ATPG program failed in generating activation test patterns. Simulation shows that the proposed Trojan designs can keep dormant in all simulations. Since the Trojan modifies the original designs, there is no reference chip available for side-channel analysis detections, and the slightly higher area overhead is acceptable in modern designs. Even if re-synthesis can be conducted on the Trojans, the registers inserted will prevent the re-convergent logic from being optimized so no rare transition signals will be generated. These Trojan designs will definitely expedite new developments in hardware Trojan detection research.

## Acknowledgments

This work was supported by National Key R&D Program of China (2018YFB0904900, 2018YFB0904902).

### References

[1] S. Adee: "The Hunt For The Kill Switch" IEEE Spectrum **45** (2008) 34 (DOI: 10.1109/MSPEC.2008.4505310).
[2] S. Skorobogatov and C. Woods: "Breakthrough silicon scanning discovers backdoor in military chip," Proceedings of the 14th international conference on Cryptographic Hardware and Embedded Systems (CHES'12) (2012) 23 (DOI: http://dx.doi.org/10.1007/978-3-642-33027-8_2).
[3] M. Tehranipoor and F. Koushanfar: "A Survey of Hardware Trojan Taxonomy and Detection" IEEE Design and Test of Computers **27** (2010) 10 (DOI: 10.1109/MDT.2010.7).
[4] S. Bhunia, *et al.*: "Hardware Trojan Attacks: Threat Analysis and Countermeasures," Proceedings of the IEEE **102** (2014) 1229 (DOI: 10.1109/JPROC.2014.2334493).
[5] R. Karri, *et al.*: "Trojan Taxonomy," In *Introduction to Hardware Security and Trust,* eds. M. Tehranipoor and C. Wang

(Springer, New York, 2012).
[6] H. Salmani, *et al.*: "On design vulnerability analysis and trust benchmarks development," IEEE 31st International Conference on Computer Design (ICCD) (2013) 471 (DOI: 10.1109/ICCD.2013.6657085).
[7] B. Shakya, *et al.*: "Benchmarking of Hardware Trojans and Maliciously Affected Circuits," J. Hardware and Systems Security (2017) 85 (DOI: https://doi.org/10.1007/s41635-017-0001-6).
[8] S. K. Haider, *et al.*: "Advancing the state-of-the-art in hardware Trojans design," IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) (2017) 823 (DOI: 10.1109/MWSCAS.2017.8053050).
[9] J. Cruz, *et al.*: "An automated configurable Trojan insertion framework for dynamic trust benchmarks," Design, Automation Test in Europe Conference Exhibition (DATE) (2018) 1598.
[10] H. Salmani and M. Tehranipoor: "Layout-Aware Switching Activity Localization to Enhance Hardware Trojan Detection," IEEE Transactions on Information Forensics and Security **7** (2012) 76 (DOI: 10.1109/TIFS.2011.2164908).
[11] K. Yang, *et al.*: "A2: Analog Malicious Hardware," IEEE Symposium on Security and Privacy (SP) (2016) 18 (DOI: 10.1109/SP.2016.10).
[12] J. Zhang and Q. Xu: "On hardware Trojan design and implementation at register-transfer level," IEEE International Symposium on Hardware-Oriented Security and Trust (HOST) (2013) 107 (DOI: 10.1109/HST.2013.6581574).
[13] C. Sturton, *et al.*: "Defeating UCI: Building Stealthy and Malicious Hardware," IEEE Symposium on Security and Privacy (2011) 64 (DOI: 10.1109/SP.2011.32).
[14] Jie Zhang, *et al.*: "DeTrust: Defeating Hardware Trust Verification with Stealthy Implicitly-Triggered Hardware Trojans" Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14) (2014) 153 (DOI: https://doi.org/10.1145/2660267.2660289).
[15] W. Hu, *et al.*: "Why you should care about don't cares: Exploiting internal don't care conditions for hardware Trojans," (2017) 707 (DOI: 10.1109/ICCAD.2017.8203846).
[16] Trust-HUB: http://www.trust-hub.org/benchmarks/Trojan.
[17] H. Salmani: "COTD: Reference-Free Hardware Trojan Detection and Recovery Based on Controllability and Observability in Gate-Level Netlist," IEEE Transactions on Information Forensics and Security **12** (2017) 338 (DOI: 10.1109/TIFS.2016.2613842).
[18] X. Xie, *et al.*: "Hardware Trojans Classification Based on Controllability and Observability in Gate-Level Netlist," IEICE Electronics Express **14** (2017) (DOI: https://doi.org/10.1587/elex.14.20170682).
[19] M. Zou, *et al.*: "Potential Trigger Detection for Hardware Trojans," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **37** (2018) 1384 (DOI: 10.1109/TCAD.2017.2753201).
[20] A. Waksman, *et al.*: "FANCI: identification of stealthy malicious logic using boolean functional analysis," Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security (CCS '13) (2013) 697 (DOI: http://dx.doi.org/10.1145/2508859.2516654).
[21] M. Hicks, *et al.*: "Overcoming an Untrusted Computing Base: Detecting and Removing Malicious Hardware Automatically," IEEE Symposium on Security and Privacy (2010) 159 (DOI: 10.1109/SP.2010.18).
[22] J. Zhang, *et al.*: "VeriTrust: Verification for Hardware Trust," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **34** (2015) 7 (DOI: 10.1109/TCAD.2015.2422836).
[23] R. S. Chakraborty, *et al.*: "MERO: A Statistical Approach for Hardware Trojan Detection," Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '09) (2009) 396 (DOI: http://dx.doi.org/10.1007/978-3-642-04138-9_28).
[24] Y. Huang, *et al.*: "Scalable Test Generation for Trojan Detection Using Side Channel Analysis," IEEE Transactions on Information Forensics and Security **13** (2018) 2746 (DOI: 10.1109/TIFS.2018.2833059).
[25] S. Saha, *et al.*: "Improved Test Pattern Generation for Hardware Trojan Detection Using Genetic Algorithm and Boolean Satisfiability," (2015) 577 (DOI: 10.1007/978-3-662-48324-4_29).
[26] L. T. Wang, *et al.*: in *VLSI Test Principles and Architectures: Design for Testability,* ed. L. T. Wang, *et al.* (San Francisco, CA, USA, 2006) 40.
[27] S. Yao, *et al.*: "FASTrust: Feature analysis for third-party IP trust verification," IEEE International Test Conference (ITC) (2015) 1 (DOI: 10.1109/TEST.2015.7342417).
[28] X. Chen, *et al.*: "Hardware Trojan Detection in Third-Party Digital Intellectual Property Cores by Multilevel Feature Analysis," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **37** (2018) 1370 (DOI: 10.1109/TCAD.2017.2748021).
[29] M. OYA, *et al.*: "A Hardware-Trojans Identifying Method Based on Trojan Net Scoring at Gate-Level Netlists," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E98-A** (2015) 2537 (DOI: 10.1587/transfun.E98.A.2537).
[30] M. OYA, *et al.*: "Hardware-Trojans Rank: Quantitative Evaluation of Security Threats at Gate-Level Netlists by Pattern Matching," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **E99-A** (2016) 2335 (DOI: 10.1587/transfun.E99.A.2335).