Chapter 3
Digital Circuit Vulnerabilities to Hardware Trojans

Hassan Salmani and Mark Tehranipoor

3.1 Introduction

Adopted in the interest of economy, the horizontal integrated circuit design process has raised serious concerns for national security and critical infrastructures [1, 2]. An adversary is afforded plenty of opportunities to interfere with its design process, and circuit parametric or functional specifications may be jeopardized, allowing malicious activities [3–6]. Any intentional modification of design specifications and functionality to undermine its characteristics and correctness is called a hardware Trojan.

Hardware Trojans can be realized by including additional circuits at the register transfer or gate-level or by changing circuit parameters like wire thickness or component size at the layout-level, to name a few. Hardware Trojans can reduce circuit reliability, change or disable its functionality at a certain time, reveal its detailed implementation, or grant covert access to unauthorized entities [7, 8]. For instance, a third party intellectual property (IP) provider can enclose extra circuitry within a cryptographic module at the gate-level to leak its secret key.

A number of proposed approaches facilitate hardware Trojan detection by analyzing circuit side-channel signals or by increasing the probability of Trojan full activation. Incurred extra switching activity or induced additional wiring and gate capacitance affects circuit side-channel signals such as power and delay. Path delay fingerprint and delay measurement based on shadow registers are techniques intended to capture Trojan impact on circuit delay characteristics [9, 10]. Transient

H. Salmani (✉)
Howard University, Washington, DC, USA
e-mail: hassan.salmani@howard.edu

M. Tehranipoor
University of Florida, Gainesville, FL, USA
e-mail: tehranipoor@ece.ufl.edu

P. Mishra et al. (eds.), *Hardware IP Security and Trust*,
DOI 10.1007/978-3-319-49025-0_3

current integration, circuit power fingerprint, and static current analysis are power-based Trojan detection techniques [11–13]. Efficient pattern generation is also necessary to discover a Trojan's impact upon circuit characteristics beyond process and environmental variations [14, 15]. In addition to new pattern generation techniques, design-for-hardware-trust methodologies have been proposed to increase switching activity inside a Trojan circuit while reducing main circuit switching activity acting as background noise, to enhance Trojan detection resolution [3, 4, 6].

At the gate-level, hardware Trojan triggers might be driven by signals with low transition probabilities to reduce Trojan contribution into circuit specifications. As a circuit layout is available to an untrusted foundry for fabrication, a hardware Trojan might be inserted in existing white spaces to leave circuit layout size unchanged. Therefore, a systematic approach is required to analyze the susceptibility of gate-level netlists and circuit layouts to hardware Trojan insertion to identify and quantify the vulnerability of a signal or a section to hardware Trojan insertion and attacks.

3.2 The Gate-Level Design Vulnerability Analysis Flow

Functional hardware Trojans are realized by adding or removing gates; therefore, the inclusion of Trojan gates or the elimination of circuit gates affects circuit side-channel signals such as power consumption and delay characteristics, as well as the functionality. To minimize a Trojan's contribution to the circuit side-channel signals, an adversary can exploit hard-to-detect areas (e.g., nets) to implement the Trojan. Hard-to-detect areas are defined as areas in a circuit not testable by well-known fault-testing techniques (stuck-at, transition delay, path delay, and bridging faults) or not having noticeable impact on the circuit side-channel signals. Therefore, a vulnerability analysis flow is required to identify such hard-to-detect areas in a circuit. These areas provide opportunities to insert hard-to-detect Trojans and invite researchers to develop techniques to make it difficult for an adversary to insert Trojans.

Figure 3.1 shows the vulnerability analysis flow performing power, delay, and structural analyses on a circuit to extract the hard-to-detect areas. Any transition inside a Trojan circuit increases the overall transient power consumption; therefore, it is expected that Trojan inputs are supplied by nets with low transition probabilities to reduce activity inside the Trojan circuit.

The *Power Analysis* step in Fig. 3.1 is based on analyzing switching activity; it determines the transition probability of every net in the circuit assuming the probability of 0.5 for "0" or "1" at primary inputs and at memory cells' outputs. Then, nets with transition probabilities below a certain threshold are considered as possible Trojan inputs. The *Delay Analysis* step performs path delay measurement based on gates' capacitance. This allows to measure the additional delay induced by Trojan by knowing the added capacitance to circuit paths. The Delay Analysis step identifies nets on non-critical paths as they are more susceptible to Trojan insertion and harder to detect their changed delay. To further reduce Trojan impact on circuit

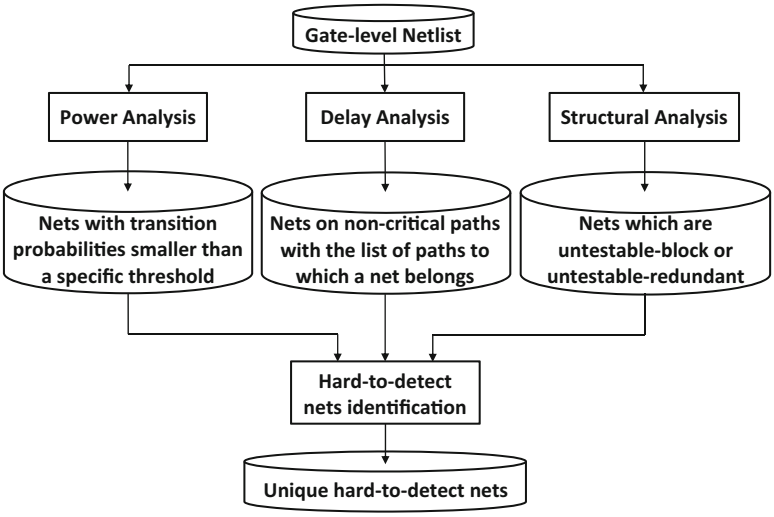

Fig. 3.1 The gate-level vulnerability analysis flow

delay characteristics, it also reports the paths to which a net belongs to avoid selecting nets belonging to different sections of one path. The *Structural Analysis* step executes the structural transition delay fault testing to find untestable blocked and untestable redundant nets. Untestable redundant nets are not testable because they are masked by a redundant logic, and they are not observable through primary output or scan cells. Untestable blocked nets are not controllable or observable by untestable redundant nets. Tapping Trojan inputs to untestable nets hides Trojan impact on delay variations.

At its end, the vulnerability analysis flow reports unique hard-to-detect nets that are the list of untestable nets with low transition probabilities and nets with low transition probabilities on non-critical paths while not sharing any common path. Note that when a Trojan impacts more than one path, it provides greater opportunities for detection. Avoiding shared paths makes a Trojan's contribution to affected paths' delay minimal, which can be masked by process variations, making it difficult to detect and distinguish the added delay from variations. The reported nets are ensured to be untestable by structural test patterns used in production tests. They also have low transition probabilities so Trojans will negligibly affect circuit power consumption. As the nets are chosen from non-critical paths without any shared segments, it would be extremely difficult to detect Trojans by delay-based techniques.

The vulnerability analysis flow can be implemented using most electronic design automation (EDA) tools, and the complexity of the analysis is linear with respect to the number of nets in the circuit. The flow is applied to the Ethernet MAC 10GE circuit [16], which implements 10Gbps Ethernet Media Access Control functions. Synthesized at 90nm Synopsys technology node, the Ethernet MAC 10GE circuit

consists of 102,047 components, including 21,830 flip-flops. The Power Analysis shows that out of 102,669 nets in the circuit, 23,783 of them have a transition probability smaller than 0.1, 7003 of them smaller than 0.01, 367 of them smaller than 0.001, and 99 of them smaller than 0.0001. The Delay Analysis indicates that the largest capacitance along a path, representing path delay, in the circuit is 0.065717825 pF, and there are 14,927 paths in the circuit whose path capacitance is smaller than 70 % of the largest capacitance, assuming that paths longer than 70 % in a circuit can be tested using testers. The Structural Analysis finds that there is no untestable fault in the circuit. By excluding nets sharing different segments of one path, there are 494 nets in the Ethernet MAC 10GE circuit considered to be areas where Trojan inputs could be used while ensuring the high difficulty of detection based on side-channel and functional test techniques.

3.3 The Layout-Level Design Vulnerability Analysis Flow

A physical design tool takes a synthesized netlist and associated technology library information and performs placement and routing considering design constraints such as performance, size, and manufacturability. Cells are typically placed in rows and their interconnections are realized through metal layers above the cells. Large circuits such as system-on-chips take larger areas for placement and require more metal layers for routing to meet circuit constraints, besides circuit functionality. However, a final circuit layout may contain a considerable amount of whitespaces in substrate and empty routing channels in metal layers above the substrate. These empty spaces can be used by an untrusted foundry to place and route Trojan cells with minimum impact on circuit specification. To study the vulnerability of a circuit layout to hardware Trojan insertion, Fig. 3.2 shows a novel circuit vulnerability analysis flow at the layout-level.

3.3.1 Cell and Routing Analyses

A gate-level synthesized netlist, along with design constraints and technology library information, is fed into a physical design tool for placement and routing. The circuit layout, the output of physical design, shows gates' location and their detail wiring through metal layers. In addition to the circuit layout, an updated design netlist with circuit parasitic parameters is obtained. The proposed flow includes two major steps: the *Cell Analysis* and the *Routing Analysis* to study cell distribution and routing congestion. The Cell Analysis step screens the circuit silicon to extract cells' location. It then determines whitespaces distribution and their size. The Routing Analysis step extracts used routing channels in each metal layer and determines unused ones.

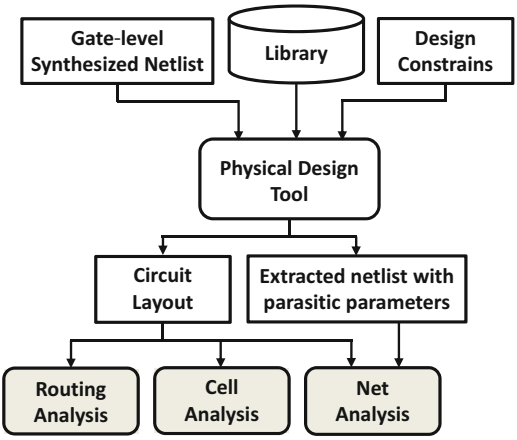

Fig. 3.2 The layout-level vulnerability analysis flow

After obtaining the circuit layout, the Cell Analysis obtains the circuit size and collects placed cells and their coordination by screening the circuit layout. Using these information, whitespaces in the circuit substrate are identified. Any whitespace whose area size is greater than that of the smallest cell in the technology library is considered a potential location for one or more Trojan cells insertion. The Cell Analysis also obtains the distribution of cells and whitespaces across the layout. The Routing Analysis collects used and unused routing channels in metal layers above the substrate. Available routing channels can potentially be used for Trojan cells interconnection and their connections to the main circuit. Similar to the Cell Analysis, the Routing Analysis also collects the distribution of used and empty routing channels in all metal layers. After determining whitespace and unused routing channels distributions of a circuit layout, the vulnerability of a region of circuit layout to hardware Trojan cells placement is defined as:

$$V(r) = WS(r) \times UR(r) \tag{3.1}$$

where $V(r)$ indicates the vulnerability of region r, $WS(r)$ the normalized whitespace of region r, and $UR(r)$ the normalized unused routing channels of region r. It is expected that Trojan cells are inserted in regions with high $V(r)$ where there are equally high $WS(r)$ and $UR(r)$.

While being inserted in a region with high V may not guarantee Trojan detection avoidance, to remain hidden from delay-based detection techniques, Trojans should be inserted in regions with enough whitespace and empty routing channels and tapped to nets on non-critical paths. The value of vulnerability to delay-resistant Trojans ($V_{\mathrm{Td}}(r)$) in the region r can be defined as

$$V_{\mathrm{Td}}(r) = V(r) \times N_{\mathrm{NC}}(r) \tag{3.2}$$

where $N_{NC}(r)$ is the number of non-critical path in the region r and $V(r)$ is the vulnerability of region r as defined by Eq. (3.1).

To stand power-based detection technique, Trojans should be connected to nets with low transition probabilities and placed in regions with enough whitespace and unused routing channel. The value of vulnerability to power-resistant Trojans ($V_{Tp}(r)$) in the region r can be defined as

$$V_{Tp}(r) = V(r) \times N_{LP}(r) \tag{3.3}$$

where $N_{LP}(r)$ is the number of nets with transition probability smaller than a predefined P_{th} in the region r, and $V(r)$ is the vulnerability of region r as defined by Eq. (3.1).

Trojans resistant to multi-parameter detection techniques should be placed in regions with empty space and unused routing channels and connected to nets with low transition probabilities located on non-critical paths. The value of vulnerability to power and delay-resistant Trojans ($V_{Tdp}(r)$) in region r can be defined as

$$V_{Tdp}(r) = V(r) \times N_{NC\&LP}(r) \tag{3.4}$$

where $N_{NC\&LP}(r)$ is the number of nets with transition probability less than a predefined P_{th} on non-critical paths in the region r and $V(r)$ is the vulnerability of region r as defined by Eq. (3.1).

3.3.2 Net Analysis

The Net Analysis performs a comprehensive analysis of each net in a circuit. The analysis determines the transition probability of each net in the circuit. With incorporating the circuit layout information, it is possible to obtain the distribution of transition probability across the circuit layout. Using a timing analysis tool, the slack distribution of worst paths passing through a net can be obtained. Nets with low transition probability located on non-critical paths are suitable candidates for Trojan trigger inputs.

In conclusion, the layout-level vulnerability analysis flow identifies regions of a circuit that are more vulnerable to Trojans resistant to delay-based, power-based, and multi-parameter-based detection techniques. Furthermore, the vulnerability of a region is quantified, and this provides a detailed and fair comparison between different circuit implementations. With such knowledge, it is possible to incorporate effective prevention techniques with the least impact on main design specifications. In addition, the flow may provide insightful guidance for authenticating circuits after manufacturing.

b18 benchmark is synthesized using Synopsys's SAED_EDK90nm library at 90nm technology node [17] and 9 metal layers for routing is considered. The layout is divided into tiles with the area A_T equal to W^2 where W is the width of the largest

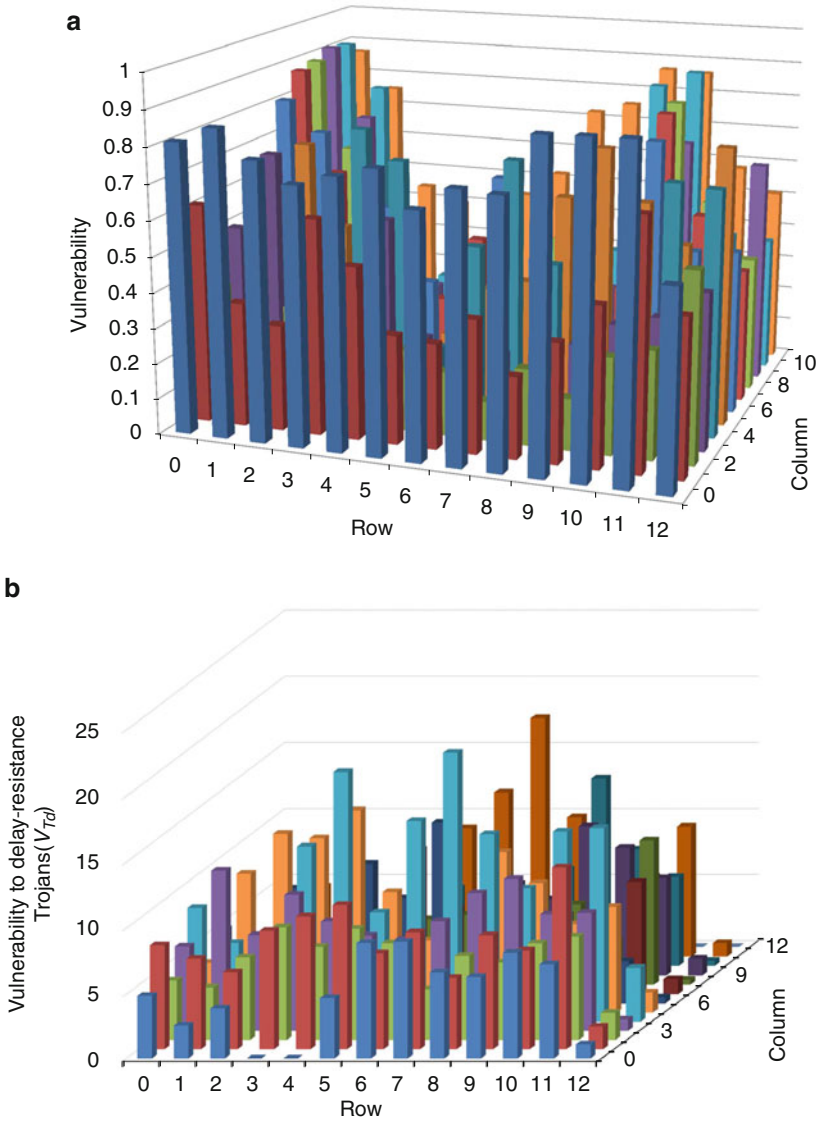

Fig. 3.3 Existence of unused space and routing channels and vulnerability to delay-resistant hardware Trojans in b15 benchmark. (**a**) Unused space and routing channels($V(r)$). (**b**) Delay-resistant Trojans

cell in the design library; that is, $A_T = 560.7424\,\mu m^2$. The average available white space and unused routing channel are about 41 unit of INVX0 and 0.84 per layer, respectively. After performing layout vulnerability analysis flow, Fig. 3.3a presents the vulnerability of b15 benchmark [18] to Trojan insertion at the layout-level as

defined by Eq. (3.1). The average vulnerability is about 0.46 and about 40 % of regions have V above 0.5. These signify considerably high susceptibility of the layout to Trojan insertion.

Figure 3.3b shows the vulnerability of b15 benchmark to Trojans resilient to delay-based detection techniques across the layout. The results indicate the region 95 in Row 7 and Column 4 is the most susceptible region to delay-resistant Trojan with $V_{\mathrm{Td}}(95) = 20.44$, where $N_{\mathrm{NC}}(95) = 28$ and $V(95) = 0.73$. Interestingly, the adjacent region 94 in Row 7 and Column 3 has considerably higher number of non-critical paths ($N_{\mathrm{NC}}(94) = 40$); however, the region 94 has low whitespaces or unused routing channels, $V(94) = 0.21$. Therefore, the susceptibility of region 94 is much lower with $V_{\mathrm{Td}}(94) = 8.4$. The vulnerability of b15 benchmark to Trojans resilient to power-based detection techniques across the layout is shown in Fig. 3.4a with $P_{th} = 1e-04$. The results show that the region 147 in row 11 and column 4 has the maximum vulnerability to power-resistant Trojan with $V_{\mathrm{Tp}}(147) = 7.71$ where $V(147) = 0.70$ and $N_{\mathrm{LP}}(147) = 11$.

Comparing results for delay-resistant Trojans and power-resistant Trojans reveals that b15 benchmark in more susceptible to delay-resistant Trojans as the maximum V_{Td} is greater than the maximum V_{Tp}. Furthermore, regions with the maximum V_{Td} and V_{Tp} are different such that the most susceptible region to delay-resistant Trojan is region 95 and the most susceptible region to power-resistant Trojan is region 147 for b15 benchmark.

Figure 3.4b shows the vulnerability of b15 benchmark to Trojans resilient to multi-parameter power and delay Trojan detection techniques. The results reveal that the region 150 in row 11 and column 7 is the most susceptible region to Trojans resilient to power and delay-based detection techniques with $V_{\mathrm{Tdp}}(150) = 5.32$ where $V(150) = 0.53$ and $N_{\mathrm{NC\&LP}}(150) = 10$. The analysis signifies even with using multi-parameter Trojan detection techniques, it is possible to implement a Trojan whose full activation probability can be about $1 - E40$ while there is still considerable whitespace and unused routing channels in the region 150. This analysis flags regions that are highly susceptible to Trojan insertion; therefore, it can effectively limit Trojan investigation into a limited number of regions. The detailed analysis for b15 benchmark shows that 21 regions out of 169 regions have V_{Tdp} above 3 that indicates the moderate existence of regions with considerable whitespace and unused routing channels and a considerable number of nets with low transition probability on non-critical paths.

The detailed results for b15 benchmark show that the percentage of regions with V_{Td} above 5, 10, and 15 are 75 %, 17 %, and 3 %, respectively. The percentage of regions with V_{Tp} above 2, 4, and 5 are 14 %, 4 %, and 0.6 %, and the percentage of regions with V_{Tdp} above 2, 4, and 5 are 12 %, 2 %, and 0 %. The results emphasize that the b15 benchmark has higher percentage of regions vulnerable to Trojans resilient to delay-based Trojan detection techniques. Further, by using a multi-parameter power and delay Trojan detection techniques these percentages significantly drop.

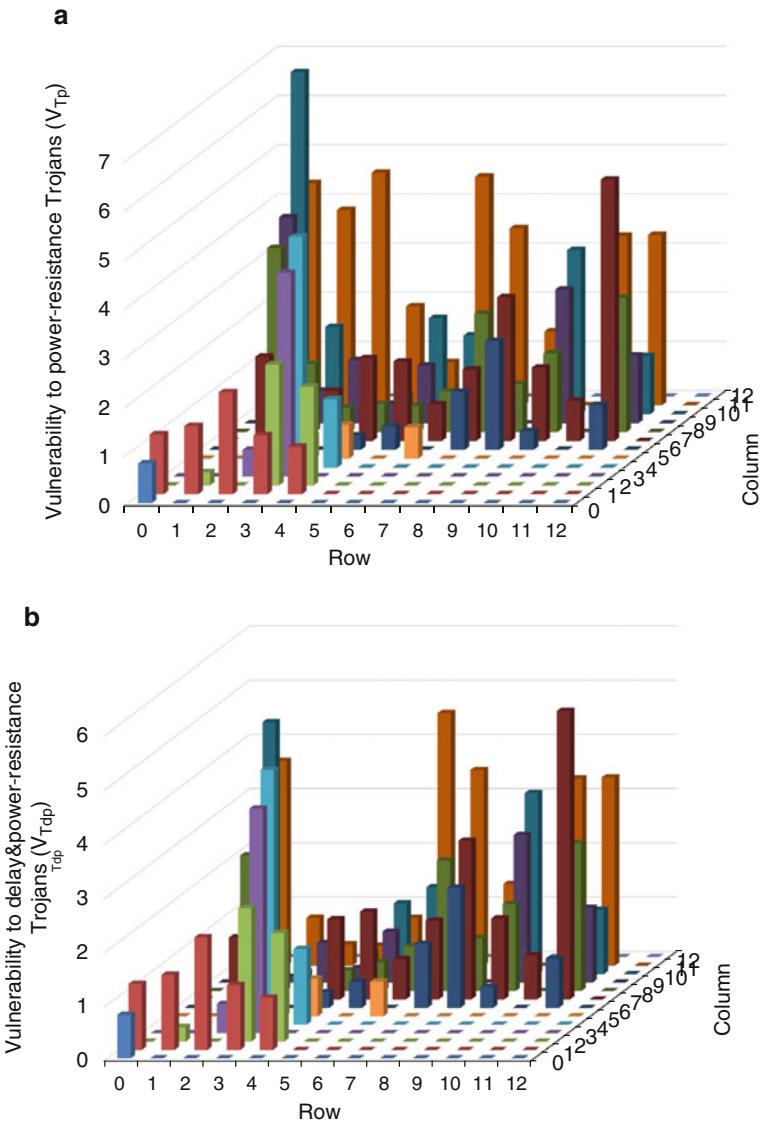

Fig. 3.4 Vulnerability of b15 benchmark to Trojans resilient to power-based and multi-parameter-based detection techniques. (**a**) Power-resilient Trojans with $P_{th} = 1e - 04$. (**b**) Multi-parameter-resilient Trojans

3.4 Trojan Analyses

A Trojan's impact on circuit characteristics depends on its implementation. Trojan inputs tapped from nets with higher transition probabilities will aggrandize switching activity inside the Trojan circuit and increase its contribution to circuit power consumption. Furthermore, the Trojan might affect circuit delay characteristics due to additional capacitance induced by extra routing and Trojan gates. To quantitatively determine the difficulty of detecting a gate-level Trojan, a procedure is developed to determine Trojan detectability based on its impact on delay and power across different circuits. Trojan detectability can establish a fair comparison among different hardware Trojan detection techniques since it is based on induced variations by a Trojan in side-channel signals.

The Trojan detectability metric is determined by (1) the number of transitions in the Trojan circuit and (2) extra capacitance induced by Trojan gates and their routing. This metric is designed to be forward-compatible with new approaches for Trojan detection by introducing a new variable, for example, a quantity related to the electromagnetic field.

Transitions in a Trojan circuit reflect Trojan contribution to circuit power consumption, and Trojan impact on circuit delay characteristic is represented by measuring the added capacitance by the Trojan. Assuming A_{Trojan} represents the number of transitions in the Trojan circuit, S_{Trojan} the Trojan circuit size in terms of the number of cells, A_{TjFree} the number of transitions in the Trojan-free circuit, S_{TjFree} the Trojan-free circuit size in terms of the number of cells, TIC the added capacitance by Trojan as Trojan-induced capacitance, and C_{TjFree} the Trojan-affected path with the largest capacitance in the corresponding Trojan-free circuit, Trojan detectability ($T_{\text{Detectability}}$) at the gate-level is defined as

$$T_{\text{Detectability}} = |t| \tag{3.5}$$

where

$$t = \left(\frac{A_{\text{Trojan}}/S_{\text{Trojan}}}{A_{\text{TjFree}}/S_{\text{TjFree}}}, \frac{\text{TIC}}{C_{\text{TjFree}}} \right) \tag{3.6}$$

$T_{\text{Detectability}}$ at the gate-level is calculated as follows:

1. Apply random inputs to a Trojan-free circuit and obtain the number of transitions in the circuit (A_{TjFree}).
2. Apply the same random vectors to the circuit with a Trojan and obtain the number of transitions in the Trojan circuit (A_{Trojan}).
3. Perform the delay analysis on the Trojan-free and Trojan-inserted circuits.
4. Obtain the list of paths whose capacitance is changed by the Trojan.
5. Determine the Trojan-affected path with the largest capacitance in the corresponding Trojan-free (C_{TjFree}) and the added capacitance (TIC).

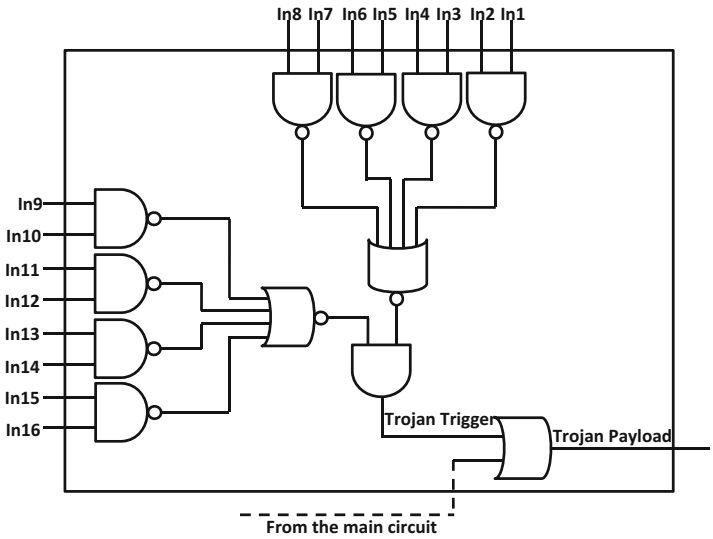

Fig. 3.5 An example comparator Trojan

Table 3.1 The detectability of the comparator Trojan placed at four different locations in Ethernet MAC 10GE circuit

Trojan	A_{TjFree}	S_{TjFree}	A_{Trojan}	S_{Trojan}	TIC (pF)	C_{TjFree} (pF)	$T_{\text{Detectability}}$
TjG-Loc1	106,664,486	102,047	10,682	12	0.000286935	0.041358674	0.851659
TjG-Loc2	106,664,486	102,047	4229	12	0.004969767	0.072111502	0.344132
TjG-Loc3	106,664,486	102,047	3598	12	0.005005983	0.049687761	0.304031
TjG-Loc4	106,664,486	102,047	13,484	12	0.004932996	0.052602269	1.079105

6. Form the vector t (3.6) and compute $T_{\text{Detectability}}$ as defined in Eq. (3.5). Note that Trojan detectability represents the difficulty of detecting a Trojan.

As an example, the comparator Trojan, shown in Fig. 3.5, is inserted at four different locations, namely TjG-Loc1, TjG-Loc2, TjG-Loc3, and TjG-Loc4 (G represents "gate level"), in the Ethernet MAC 10GE circuit, and Table 3.1 shows their detectability. The Ethernet MAC 10GE circuit consists of 102,047 cells, Column 3 S_{TjFree}, while the Trojan size with 12 cells, Column 5 S_{Trojan}, is only about 0.011 % of the entire circuit. TjG-Loc4, in Row 5, experiences the largest switching activity (13,484 in Column 4) and relatively induces high TIC (0.004932996 pF in Column 6). It is expected that TjG-Loc4 will be the easiest Trojan to be detected due to more impact on circuit side-channel signals, and in turn the detectability of TjG-Loc4 ($T_{\text{Detectability}} = 1.079105$ in Column 8) is higher than the others. Although the induced capacitance by TjG-Loc2 (0.004969767 pF), in Row 3, is more than the capacitance induced by TjG-Loc1 (0.000286935 pF), in Row 2, TjG-Loc1 has more significant contribution into circuit switching activity, 10,682 versus 4229 in

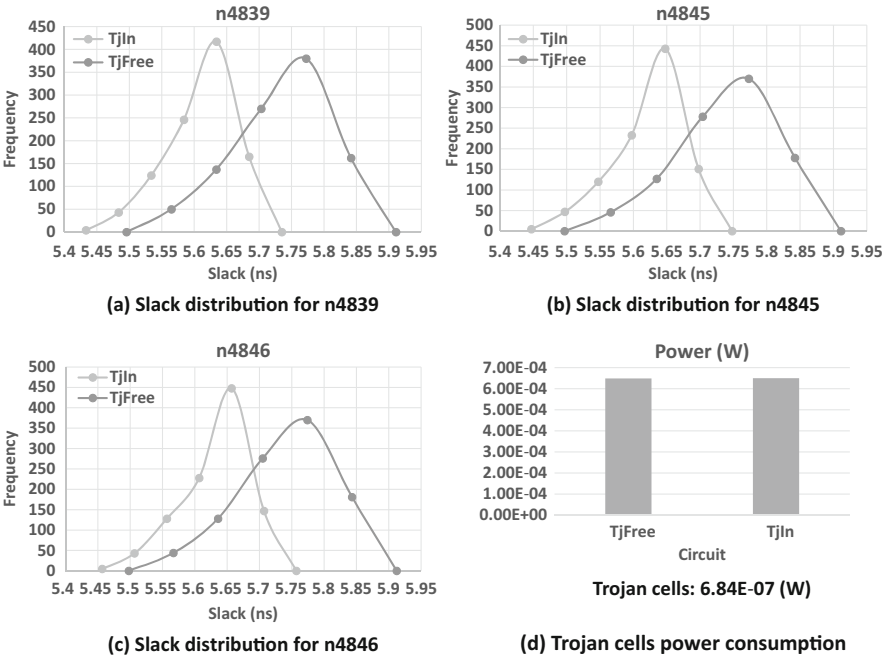

Fig. 3.6 The circuit power consumption and the slack distribution of 1000 worst paths passing through each triggering signal of a 3-bit synchronous counter Trojan inserted in Region 42 of b15 benchmark with $V_{\text{Tdp}}(42) = 4.149$

Column 4. Therefore, TjG-Loc1 has the second largest detectability (0.851659) after TjG-Loc4. Among TjG-Loc2 and TjG-Loc3, although TjG-Loc3, in Row 4, has slightly larger induced capacitance (0.005005983 pF), TjG-Loc2 experiences more switching activity (4229 versus 3598 in Column 4). The two Trojans have close detectability where TjG-Loc2 stands above and TjG-Loc3 remains the hardest Trojan to be detected with the lowest Trojan detectability.

At the layout-level, the V_{Tdp} metric determines the vulnerability of a region to Trojans which are resistant to both delay-based detection techniques and power-based detection techniques. One 3-bit synchronous counter Trojan and one 12-bit comparator separately inserted into b15 benchmark and Figs. 3.6 and 3.7, respectively, show their delay and power impacts. The counter Trojan is inserted in the region 42 at Column 3 and Row 3 with $V_{\text{Tdp}}(42) = 4.149$ ($V(42) = 0.4149$ and $N_{\text{NC\&LP}}(42) = 10$). Figure 3.6a–c shows the slack distribution of the 1000 worst paths passing through the three triggering signals of the counter. The analysis indicates the amount of TID for the three signals is about $1ns$, on average, and the minimum slack for each signal after Trojan insertion still is so large that the worst path is not become a critical path. Figure 3.6d also presents the very small power consumption of the Trojan circuit ($\approx 6.84E-07$ W), and the circuit power consumption before and after Trojan insertion is almost remained the same, about

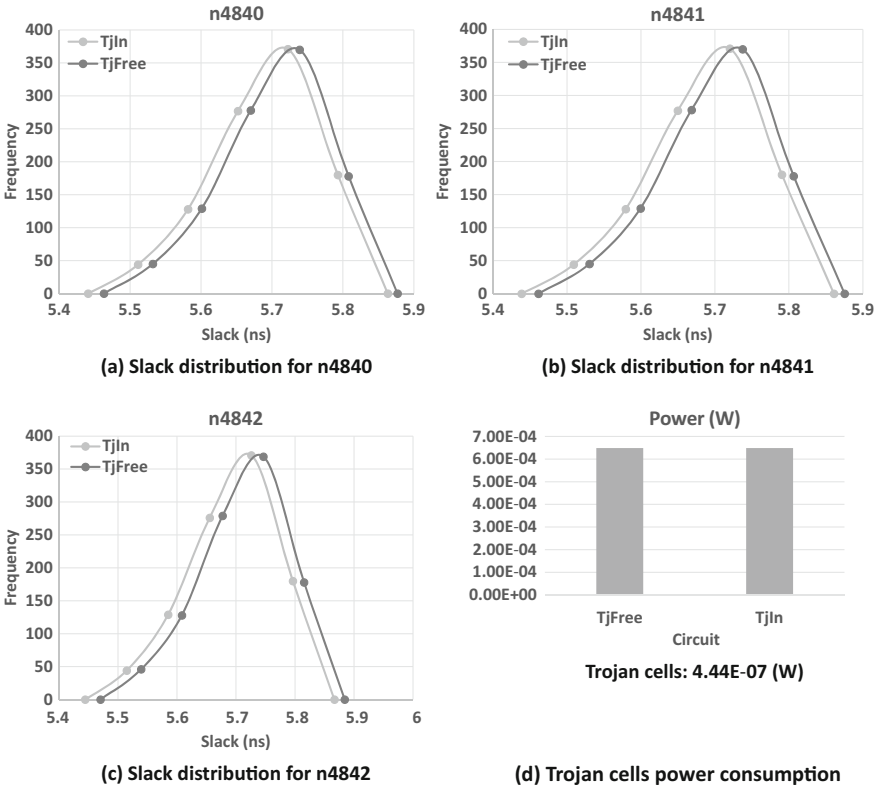

Fig. 3.7 The circuit power consumption and the slack distribution of 1000 worst paths passing through three selected triggering signal of a 12-bit comparator Trojan with minimum slack inserted in Region 43 of b15 benchmark with $V_{\text{Tdp}}(43) = 4.7035$

$6.49E-04$ W. Therefore, the 3-bit synchronous counter Trojan may remain hidden from both delay- and power-based Trojan detection techniques.

A similar analysis is performed for 12-bit comparator inserted in the neighboring region 43 at Column 4 and Row 3 with $V_{\text{Tdp}}(43) = 4.7035$ ($V(43) = 0.78$ and $N_{\text{NC\&LP}}(43) = 6$). The slack distributions of three selected inputs of the comparator with the minimum slacks are presented in Fig. 3.7a–c. The amount of TID is 0.018 ns, on average, and the delay of the worst path is not large enough to be considered a critical path. In Fig. 3.7d, the Trojan power consumption is very small ($\approx 4.44E-07$ W). With small impact on circuit delay characteristics and power consumption, the comparator Trojan may also remain hidden. Comparing the 3-bit synchronous counter to 12-bit combinational comparator indicate that the counter consumes more power than the comparator although the size of comparator circuit larger. This is attributed to the fact that the counter is a sequential circuit and the clock inputs of its flip-flops are connected to circuit clock. Furthermore, TID for the

comparator circuit is smaller than that of the counter this is because of the higher V_{Tdp} value of the comparator. Therefore, the V_{Tdp} metric can effectively identify regions vulnerable to Trojans resilient to delay&power-based techniques.

3.5 Conclusions

This chapter presented a novel gate- and layout-level vulnerability analysis flows presented to determine susceptibility of a gate-level netlist and a circuit layout to hardware Trojan insertion. Based on a circuit topology and placement and routing information, several metrics were defined to quantify the vulnerability of circuit layout to different types of Trojans. The significance of introduced metrics was evaluated by implementing different Trojans. The results indicated the considerable vulnerability of gate-level netlist and circuit layouts to Trojans resistant to delay-based, power-based, and multi-parameter-based Trojan detection techniques. The proposed novel layout vulnerability analysis flow may provide guidance for Trojan prevention during circuit development and Trojan detection after fabrication.

References

1. U.S.D. Of Defense, Defense science board task force on high performance microchip supply (2015) http://www.acq.osd.mil/dsb/reports/2005-02-HPMS_Report_Final.pdf
2. S. Adee, The hunt for the kill switch (2008) http://www.spectrum.ieee.org/print/6171
3. S. Bhunia, M. Abramovici, D. Agarwal, P. Bradley, M.S. Hsiao, J. Plusquellic, M. Tehranipoor, Protection against hardware Trojan attacks: towards a comprehensive solution. IEEE Des. Test **30**(3), 6–17 (2013)
4. M. Tehranipoor, F. Koushanfar, A survey of hardware Trojan taxonomy and detection. IEEE Des. Test Comput. **27**(1), 10–25 (2010)
5. R. Karri, J. Rajendran, K. Rosenfeld, M. Tehranipoor, Trustworthy hardware: identifying and classifying hardware Trojans. IEEE Comput. **43**(10), 39–46 (2010)
6. M. Tehranipoor, H. Salmani, X. Zhang, X. Wang, R. Karri, J. Rajendran, K. Rosenfeld, Trustworthy hardware: Trojan detection and design-for-trust challenges. IEEE Comput. **44**(7), 66–74 (2011)
7. Y. Jin, D. Maliuk, Y. Makris, Post-deployment trust evaluation in wireless cryptographic ICs, in *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE12)* (2012), PP. 965–970
8. X. Zhang, M. Tehranipoor, Case study: detecting hardware Trojans in third-party digital IP cores, in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST11)* (2011), pp. 67–70
9. Y. Jin, Y. Makris, Hardware Trojan detection using path delay fingerprint, in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST08)* (2008), pp. 51–57
10. J. Li, J. Lach, At-speed delay characterization for IC authentication and Trojan horse detection, in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST08)* (2008), pp. 8–14
11. X. Wang, H. Salmani, M. Tehranipoor, J. Plusquellic, Hardware Trojan detection and isolation using current integration and localized current analysis, in *Proceedings of the IEEE International Symposium on Fault and Defect Tolerance in VLSI Systems (DFT08)* (2008), pp. 87–95

12. D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, B. Sunar, Trojan detection using IC fingerprinting, in *Proceedings of the IEEE Symposium on Security and Privacy* (2007), pp. 296–310
13. R. Rad, X. Wang, J. Plusquellic, M. Tehranipoor, Power supply signal calibration techniques for improving detection resolution to hardware Trojans, in *Proceedings of the International Conference on Computer-Aided Design(ICCAD08)* (2008), pp. 632–639
14. M. Banga, M.S. Hsiao, A novel sustained vector technique for the detection of hardware Trojans, in *Proceedings of the International Conference on VLSI Design (VLSID09)* (2009), pp. 327–332
15. F. Wolff, C. Papachristou, S. Bhunia, R.S. Chakraborty, Towards Trojan-free trusted ICs: problem analysis and detection scheme, in *Proceedings of the Design, Automation and Test in Europe (DATE08)* (2008), pp. 1362–1365
16. Ethernet 10GE MAC (2013) http://opencores.org/project,xge_mac
17. Synopsys 90nm generic library for teaching IC design (2016) http://www.synopsys.com/Community/UniversityProgram/Pages
18. ISCAS benchmarks (2016) http://www.pld.ttu.ee/~maksim/benchmarks/