Acceleration of Spatial Correlation Based Hardware Trojan Detection Using Shared Grids Ratio

Fatma Nur Esirci and Alp Arslan Bayrakci

Gebze Technical University, Turkey
{fesirci,abayrakci}@gtu.edu.tr

Abstract. Due to mostly economic reasons almost all countries including the developed ones have to handle integrated circuit designs to a foreign fab for manufacturing, which raises the security issues like intentional malicious circuit (hardware Trojan) insertion by an adversary. A previously proposed method to address these security issues detects hardware Trojan using the spatial correlations in accordance with delay based side channel analysis. However, it is never applied to full circuits and it requires too many path delay computations to select correlated path pairs. In this paper, we first apply the method and present the results for full circuits and then, the method is accelerated by proposing a novel path selection criterion which avoids the computation of path delays. In terms of detection success, the resultant method performs similar to the previous one, but in a much faster fashion.

Keywords: hardware security · hardware Trojan · side channel analysis · spatial correlations · process variations

1 Introduction

The fabrication of chips is a sophisticated process that can only be performed in state-of-art fabrication facilities. Given this increasingly expanding cost and complexity of foundries, the semiconductor business model has largely shifted to a contract foundry business model over the past two decades. For example, Texas Instruments and Advanced Micro Devices, two chip making giants that have traditionally used their in-house facilities for fabricating their chips, have both in 2010 announced outsourcing most of their sub-45nm fabrication to major contract foundries worldwide [15]. One of the most crucial effects of this compulsory shift is on hardware security. Handling the design to manufacturing fab and the difficulty of detecting any malicious alteration on the manufactured chip make the system vulnerable to attacks especially during the manufacturing. Any such malicious alteration on the circuit is called hardware Trojan (HT). Several papers [14] and the IEEE Spectrum magazine articles [1] comprehensively describe the hardware Trojan threat on security and the difficulty of detection.

Especially, the mission critical circuits such as the ones used for cryptography are main targets for such hardware Trojan attacks [5, 11]. There are many

different types of proposed hardware Trojans as well as many detection methods until now [15]. Destructive methods can provide exact results but only for the investigated chip, by also making the chip useless after the destructive analysis. It cannot guarantee the authenticity of the remaining chips either. Therefore, non-destructive methods like side channel analysis are worked on to detect HT in the chips. Side channel analysis based Trojan detection methods investigates the measurable side channel signals like delay [13], power [3] and temperature [12] to reveal any HT existence. Yet, the unavoidable process variations can easily hide the effect of the inserted Trojan. This makes especially the detection of small Trojans very hard.

The spatial correlation based HT detection method in [8] claims to detect even the smallest type of Trojan composed of only one XOR gate under realistic process variations using delay based HT detection. The method takes advantage of spatial correlations that are inherently present in manufactured chips. However, it is not tested on full circuit and also it is computationally complex. In this paper, we first adapt the method in [8] to full circuits and report the results on full circuits. Then, we propose using a new criterion, called shared grids ratio, for the selection of correlated path among numerous candidates. Theoretical cost analysis of both methods as well as the experimental results are presented in the paper. The results show that the proposed improvement can speed up the method about 10 times in correlated path selection on the average, which in turn accelerates the whole method more than 2 times on the average over benchmark circuits. And this enhancement comes with almost no cost on the HT detection capability of the method.

This paper is organized as follows: Section 2 gives some background on circuit representation as graph, delay based HT detection, the effect of variations on detection and summarizes the spatial correlation based HT detection method in [8] by separating it into four stages. Section 3 adapts this method to full circuits and presents the results. A cost analysis for this adaptation is performed in Section 4. Section 5 introduces a new criterion to accelerate the method. The results and comparisons of both the previous and the new method are explained in Section 6.

2 Preliminaries

2.1 Representation of Circuits

We use graph structure to express digital circuits, where each gate in the circuit corresponds to a node and each interconnect between two gates corresponds to an edge of the graph. A path in the circuit starts from a primary input, traverses through gates (nodes) and ends at a primary output. Any edge of the circuit is assumed to have the potential of a Trojan circuit insertion.

2.2 Delay Based Trojan Detection

One of the most effective methods in the literature is delay based HT detection, which is a sub-branch of side channel analysis (SCA). Normally, a smart Tro-

jan is designed to stay at passive state so that it cannot be detected through conventional functional tests. Yet, at least a tiny part (payload) of the Trojan must be inserted on a wire in the circuit in order to be able to alter the signal at that wire when it gets active. Thus, the payload part brings some delay add-on to the original circuit. The power of delay based detection is due to the fact that they do not require to make HT active for detection in contrast to functional test based methods. Also they can be applied by widely used feasible delay tests without destroying the chip in contrast to destructive detection methods. The success of the Trojan detection based on delay is dependent on Trojan size because the bigger the Trojan is, the more delay add-on it has. And its main drawback is the process variations that can easily hide the delay add-on of the Trojan circuitry.

2.3 Variation Effect & Difficulties

Process variation is due to the nature of the chip manufacturing process. It is undesirable but inevitable. The circuits are designed according to specific constraints such as functionality, speed and power consumption. At the post manufacturing stage, the chip set obtained by manufacturing are examined to see if they meet these constraints. Yet, due to manufacturing process variations on circuit components like gate length and threshold voltage, chips do not exactly meet the same specification but instead each manufactured chip comes with different properties.

Any realistic variation model must include both inter-die (between chips) and intra-die (within the chip) variation components. As the integrated circuits scale down in feature size with developing technology, the effect of intra-die variation increases. The intra-die variation component inherently exhibits spatial correlations. As a result of the spatial correlations, the random parameters of the transistors closer to each other are affected more similar from the variations when compared with the ones residing far from each other. If the results of a method are justified by circuit simulations, it is very important to use variation models that can consider all variation components as well as accurate variation amounts corresponding to current technology [9].

The main challenge of SCA based detection is to distinguish the HT effect from the effect of process variations. SCA based detection methods either fail to use accurate variation models or fail to detect very small Trojans. To overcome this challenge, we require a method that enables us to get rid of the variation effect even under the accurate variation model.

The spatial correlation based HT detection method proposed in [8] uses such a variation model and precise transistor level Spice simulations to justify the proposed method. It also claims to detect even the smallest type of Trojans. As opposed to most SCA based methods [15], it can even work in the absence of a golden model when only a fraction of the manufactured chips have an inserted Trojan. Such selective insertion is preferred by the adversary because otherwise destructive analysis of any chip can easily reveal the Trojan existence. However, the scalability of the method is unknown as it is not executed on full circuits.

Also, it is computationally very complex to detect a correlated path pair for each edge (interconnect) in the circuit requiring numerous path delay and correlation coefficient computations. Next section summarizes this method.

2.4 Review of Spatial Correlation Based HT Detection Method [8]

Getting rid of variation effect is a hard task as the variations neither can be avoided nor can be exactly measured due their random nature. One technique is to divide components that are affected from the variations very similar so that the effect of variations is canceled out [11, 16].

Spatial Correlation Based HT Detection [8] extracts correlated paths by taking advantage of spatial correlations. As any Trojan circuit must be connected to at least one edge in the circuit, the method traverses all edges in the circuit to detect whether there is a connected Trojan on that edge. It is composed of the following stages executed for each edge in the circuit: (i) extraction of one suspected path for each edge, (ii) extraction of correlated path candidates for each suspected path, (iii) selection of one correlated path among the candidates, (iv) measurement and division of path delays of suspected and correlated paths. The first three stages are pre-manufacturing but the last stage is post manufacturing and must be applied to manufactured chips.

i It is easier to detect Trojan using short (small delay) paths in the circuit for its increased relative effect on delay. Therefore, for each edge e, the shortest path passing through that edge is selected as the suspected path (P^e_{susp}). The cost of suspected path extraction is not high as the shortest path is detected according to the nominal delay values of nodes (logic gates).

ii The second stage is the extraction of all possible path candidates which may be correlated with the suspected path. For that, spatial correlation information is used. Due to the spatial correlations, a path which has logic gates residing at very close locations with another path must have correlated path delays. This means that if one can find a very closely located path for a suspected path, the ratio of path delays of these two paths can cancel out the variation component. In this case, any alteration like Trojan insertion can be easily revealed by detecting the deviation in path delay ratio. In order to find path candidates correlated with a suspected path, the circuit is divided into grids (Fig. 3) and then, all paths whose gates are located either at the same grid or at the adjacent grids of the suspected path are extracted and collected in correlated path candidates set. At the end, each suspected path has a corresponding correlated path candidates set.

iii At the third stage, first of all, the path delays of all correlated path candidates are computed for all samples (chips). Then, for each path pair consisting of the corresponding suspected path and a correlated path candidate, the correlation coefficient is computed based on these path delays. The correlated path candidate resulting in the best correlation coefficient is nominated as the correlated path (P^e_{corr}) for the corresponding suspected path of edge e.

iv This is the post manufacturing test stage. The path delays of the suspected path and the nominated correlated path are measured and divided to cancel out the variation component, which reveals any HT existence. The computation of delay ratio for a sample edge e is shown by (1), where d denotes the path delay. The algorithm is successful in Trojan detection without requiring golden model if the resultant delay ratios for Trojan-free and Trojan-inserted samples can be separated from each other.

$$R_e = \frac{d(P^e_{susp})}{d(P^e_{corr})} \tag{1}$$

For delay computations above, a delay model called Stochastic Logical Effort (SLE) and constructed by precise transistor level Spice simulations [4] is employed. Path delay computation is performed by summing up the individual delays of gates on the path. Delay of each gate is computed by a fast and accurate gate delay model called Stochastic Logical Effort (SLE) as shown in (2). In this equation $d_r(S)$ is the delay of a logic gate r for sample sample S, $\tau(S)$ is the reference inverter delay, $p_r(S)$ and $g_r(S)$ is the parasitic component and logical effort for the same chip and h_r is the electrical fan-out for gate r. The further details of the model can be found in [4].

$$d_r(S) = \tau(S)(p_r(S) + g_r(S)h_r) \tag{2}$$

As it is quantified in this paper, one of the most time consuming part in the algorithm is stage (iii). Considering current deeply integrated circuits with even millions of gates, a numerous number of path delay and correlation coefficient computations are required as there may be a plenty of correlated path candidates considering all edges in the circuit.

3 CCM: Adaptation of the Method in Section 2.4 to Full Circuit

Spatial correlation based HT detection method in [8] is applied only to randomly selected three edges from each benchmark circuit. The paper also does not devise any method to discriminate Trojan inserted samples from the Trojan-free ones. It only reports the number of misclassified samples when the best separating line between the ratios of Trojan inserted chips and Trojan free ones is assumed. In the actual case the separation line is unknown.

To compensate these shortcomings, we adapt the algorithm to full circuit. Instead of randomly selecting just three edges in [8], we assume all edges in a circuit are suspected for Trojan existence and thus, the algorithm is executed for all edges in the circuit. Throughout the paper, this method is referred as Correlation Coefficient based Method (CCM). CCM is a direct adaptation of the stages explained in Section 2.4. Therefore, it uses correlation coefficients to eliminate correlated path candidates as stage (iii) of Section 2.4 explains. CCM requires correlation coefficients, as it searches for the path with the highest

correlation coefficient to select the path correlated most with the corresponding suspected path among the candidates. Then, this path constitutes the path pair with the suspected path. This pair is used to compute delay ratio shown by (1).

We call an edge to be *covered* if the samples with a Trojan inserted on that edge can be detected by the method after post-manufacturing tests (stage (iv) tests). A Trojan inserted sample is said to be *detected* if, for that edge, the resultant delay ratio distributions of all Trojan-free and Trojan-inserted samples are separate from each other. The two distributions are separate if their 1.5σ, one and half times the standard deviation, have positive difference. The computation of 1.5σ difference between delay ratios of Trojan-free and Trojan-inserted samples for an edge e is shown by (3).

$$\Delta_{1.5\sigma_e} = (\mu_{\hat{R}_e} - 1.5\hat{\sigma}_{\hat{R}_e}) - (\mu_{R_e} - 1.5\sigma_{R_e}) \tag{3}$$

where $\Delta_{1.5\sigma_e}$ is the 1.5σ difference for edge e, μ_{R_e} and σ_{R_e} are mean and standard deviation of the delay ratios for the Trojan-free samples, $\mu_{\hat{R}_e}$ and $\hat{\sigma}_{\hat{R}_e}$ are mean and standard deviation of the delay ratios for the Trojan-inserted samples. Please remind that the delay ratio is computed by dividing the delay of the suspected path to the delay of the corresponding correlated path in the pair (1). Therefore, if the edge can be covered, this means that the CCM has picked the right path pair for that edge to detect any Trojan insertion on it.

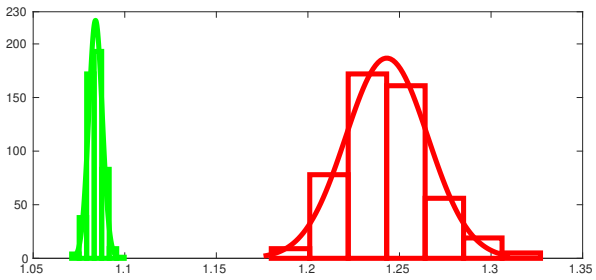

Fig. 1. Histogram of Ratios without HT (green) and with HT (red)

For instance, delay ratio distributions for a covered edge from c1908 benchmark circuit are plotted in Fig. 1. For that edge, the Trojan-inserted samples can be easily separated without a need for a golden model. Therefore, this edge is said to be secured or covered by the method. However, Fig. 2 shows another edge from c1908, which cannot be covered by the method as the delay ratios of Trojan-free and Trojan-inserted samples are intermixed into each other resulting in negative $\Delta_{1.5\sigma_e}$ and not possible to be separated if they were not colored.

Table 1 shows the results for CCM. Number of edges in the circuit, the resultant edge coverage and the number of total correlated path candidates are

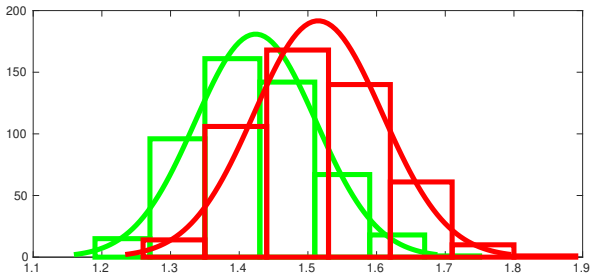

Fig. 2. Histogram of Ratios without HT (green) and with HT (red)

Table 1. Full circuit experimental results for CCM

Benchmark	**# of Edges**	**Edge Coverage**	**# of Candidates for CCM**
432	255	95.7%	36,647
499	296	64.5%	29,320
880	507	94.7%	39,176
1355	856	96.6%	17,200,357
1908	1420	93.4%	5,353,429
2670	1850	95.2%	7,619,334

the respective columns of Table 1. The number of <suspected path, correlated path> pairs is equal to the number of edges, therefore not reported in the table. The full coverage means that any Trojan inserted on any edge can be detected by the method. Results show that the coverage is just about 90% on the average over benchmark circuits in the table. It also shows the number of correlated path candidates required for CCM. This makes the stage (iii) of the algorithm explained in Section 2.4 the most unbearable part of the algorithm. Because the path delay and correlation coefficient for each correlated path candidate are computed in stage (iii).

Table 2. Time consumption for stages for CCM

Benchmark	**Run Time Stage (i)**	**Run Time Stage (ii)**	**Run Time Stage (iii)**
432	4s	54s	110s
499	7s	38s	102s
880	23s	180s	169s
1355	1.2m	13.21h	21.76h
1908	1.38m	4.96h	7.63h
2670	7.2m	6.39h	9.18h

Table 2 shows the amount of time consumption for each stage of CCM, explained in Section 2.4 – except for the post-manufacturing stage (stage (iv)) – when applied to full circuits. This table also verifies that the main bottleneck is stage (iii) for the method.

4 Computational Cost Analysis for CCM

The CCM has the main flaw of computational complexity due to mainly the stage (iii) computations. Because this stage computes the path delay using (2) for each sample chip and for each of the extracted correlated path candidates. Then, using these path delays, the correlation coefficient is computed again for each candidate. However, the number of correlated path candidates as shown in Table 1 can get very large with the increasing circuit size or complexity. Besides, the number of samples must be a big enough number to get accurate results, which also complicates stage (iii) computations.

The resultant computational cost of stage (iii) is represented by (4). In this equation, $\texttt{Nsamples}$ represents the number of chips, $\texttt{Ngates}_{\texttt{full}}$ is the number of logic gates in the full circuit, $\texttt{Cost}_{\texttt{SLE}}$ is the cost of computing SLE in (2) (two multiplications and one addition), $\texttt{Ncand}$ is the number of correlated path candidates, $\texttt{Ngates}_{\texttt{avg}}$ is the average number of gates over all correlated path candidates, $\texttt{Cost}_{\texttt{add}}$ is the cost of one addition used in path delay computation. $\texttt{Cost}_{\texttt{coeff}}$ is the unit cost for correlation coefficient computation. It utilizes arithmetic operations like addition, division and square root. Lastly $\texttt{Cost}_{\texttt{comp}}$ is the cost of comparing floating point numbers to find the max.

$$\begin{aligned}\texttt{Cost}^{\texttt{CCM}}_{\texttt{stage(iii)}} = \texttt{Nsamples} \times \texttt{Ngates}_{\texttt{full}} \times \texttt{Cost}_{\texttt{SLE}} + \\ \texttt{Ncand} \times \texttt{Nsamples} \times \texttt{Ngates}_{\texttt{avg}} \times \texttt{Cost}_{\texttt{add}} + \\ \texttt{Ncand} \times \texttt{Nsamples} \times \log(\texttt{Nsamples}) \times \texttt{Cost}_{\texttt{coeff}} + \\ \texttt{Ncand} \times \texttt{Cost}_{\texttt{comp}}\end{aligned} \tag{4}$$

The first row of the equation shows the cost of computing SLE delays for each gate and for each sample chip, the second row shows the cost of path delay computations using SLE delays computed in the first row and performed for each chip and each correlated path candidate. The third row in the equation shows the correlation coefficient computation using path delays computed in the second row. The second row and especially the third row constitute the main source of complexity. The last row is for finding the candidate with the maximum correlation coefficient.

5 Shared Grids Method (SGM) for Accelerating CCM

When Table 2 is investigated, stage (i), i.e. the suspected path extraction has a negligible cost. However, stage (iii) is about 1.7 times slower on the average than even stage (ii), which makes it the most problematic stage of the method.

The cost analysis for stage (iii) is shown by (4). The rows in that equation that have a factor of (`Ncand` × `Nsamples`) are the main source of the cost. Number of correlated path candidates for each benchmark circuit is shown in Table 1. For instance c2670 having 1850 edges resulted in more than 7.5 million correlated path candidates. Considering a thousand samples as we do in this paper, the factor above becomes about 7.5 billion which is a huge number. In actual case, the number of samples can be much larger resulting in much higher costs for stage (iii).

In this section, we propose a much faster method to select the best correlated path candidate. Due to the spatial correlations, the correlation between two paths depends on the spatial distance between them. But first of all, let us detail the actual problem with CCM.

CCM takes advantage of spatial correlation to find the correlated path pairs. It finds such a pair for each edge in the circuit so that the path delay ratio of the pair cancels the variation component which reveals any HT existence for the corresponding circuit edge. For that purpose, at stage (ii) explained in Section 2.4, the correlated path candidates of each suspected path are extracted so that all of them have their logic gates located very close to the corresponding suspected path. This closeness is guaranteed by first dividing the circuit into grids as shown in Fig. 3 and then selecting the paths residing at the same or adjacent grids of the gates of suspected path. Without loss of generalization let us assume that a suspected path has all its logic gates located at the dark shaded grids in Fig. 3. Then, the stage (ii) of CCM collects all correlated path candidates, whose logic gates are located at either the dark shaded grids or their adjacent grids that are shaded lightly on the figure.

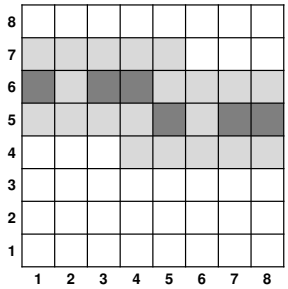

Fig. 3. The division of circuit layout to grids.

Due to the spatial correlation, one expects that all candidates (especially the ones residing at only dark shaded grids for our example case) must have a good correlation and hence a good correlation coefficient. In such a case picking just one correlated path candidate would be fairly enough to have a correlated pair instead of enumerating all of the candidates for each edge (or suspected path), and then computing the path delays and correlation coefficients for all of them.

But when we investigate the candidates, we see that this is not the case. The different correlation coefficients for all correlated path candidates corresponding to just one suspected path are shown as an example in Fig. 4. The candidates are sorted with the ascending coefficient values. For this sample case, some of the candidates may have very bad correlation coefficients down to 0.75. Please note that, empirical results show us that the correlation coefficient must have a value very close to 1 like 0.95 and above in order to be able to cancel the effect of variations and reveal the existence of Trojans. If the Trojan is as small as only one logic gate even a correlation coefficient of 0.95 may not be enough for detection. This necessitates the computation of correlation coefficient for each correlated path candidate to nominate the one with the largest coefficient as the correlated path of the pair.

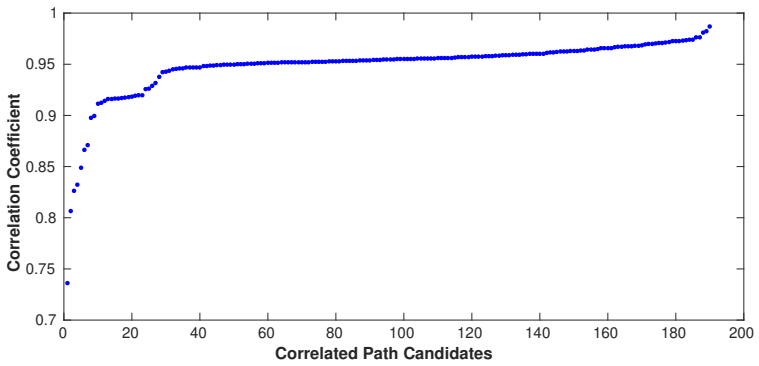

Fig. 4. CCM values for candidates of correlated path

With a further investigation, the actual reason behind that reveals the fact that being at even the same grids with the suspected path does not mean to be highly correlated with it just because the number of shared (common) grids can be fractionally very low. Without loss of generality, let us assume that the suspected path has gates distributed to n_s different grids and one correlated path candidate for that suspected path has all its gates located at n_c different grids, where the number of union and intersection of n_s and n_c grids are denoted by $n_\cup$ and $n_\cap$ respectively. The resultant correlation between these two paths would not be good enough to cancel variation effect if $n_\cap$ is much smaller than $n_\cup$. We name the $n_\cap/n_\cup$ ratio as *shared grids ratio* (SG). Shared grids ratio for a path pair $< \mathtt{P_{susp}}, \mathtt{P^i_{corr}} >$ can be computed as shown in (5). i denotes the index of the correlated path candidate for the suspected path. The correlation between two paths tends to increase by the increasing shared grids ratio.

$$\mathtt{SG^i} = \frac{\mathtt{number\ of\ shared\ grids\ for\ } < \mathtt{P_{susp}}, \mathtt{P^i_{corr}} > \mathtt{\ pair}}{\mathtt{number\ of\ all\ grids\ in\ } \mathtt{P_{susp}} \cup \mathtt{P^i_{corr}}} \tag{5}$$

For the acceleration of stage (iii), we propose selecting the correlated path candidate with largest SG ratio computed by (5) instead of the one with the largest correlation coefficient. For a sample suspected path, Fig. 5 demonstrates how correlation coefficient has a rise trend despite some fluctuations while SG increases. Usage of SG is based on the fact that the more grids the two paths share, the more spatial correlation they would have, which means better detection.

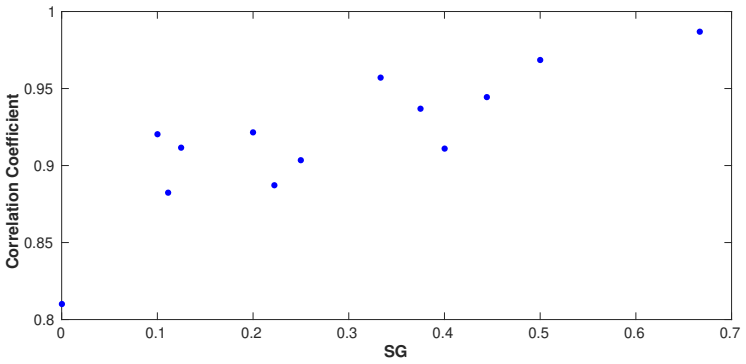

Fig. 5. CCM vs SGM for candidates of correlated path

SG only requires the detection of the number of total grids that both paths reside at ($\mathtt{Cost}_{\mathtt{DNM}}$) as well as the grids that are shared by both paths ($\mathtt{Cost}_{\mathtt{NM}}$). Then, one division is enough to get SG ($\mathtt{Cost}_{\mathtt{div}}$). No path delays and no costly correlation coefficient computation are required. Moreover, it is not performed for each sample as SG does not change from chip to chip. As a result, the new cost of stage (iii) can be written as shown by (6). $\mathtt{Ngrids}_{\mathtt{avg}}$ shows the average number of grids that are occupied by a path pair. It is certain that $\mathtt{Ngrids}_{\mathtt{avg}}$ is much smaller than the number of samples. $\mathtt{Ncand} \times \mathtt{Cost}_{\mathtt{comp}}$ is for finding the correlated path candidate i with the maximum $\mathtt{SG}^{\mathtt{i}}$ similar to (4).

$$\mathtt{Cost}^{\mathtt{SGM}}_{\mathtt{stage(iii)}} = \mathtt{Ncand} \times \mathtt{Ngrids}_{\mathtt{avg}} \times (\mathtt{Cost}_{\mathtt{DNM}} + \mathtt{Cost}_{\mathtt{NM}} + \mathtt{Cost}_{\mathtt{div}}) + \mathtt{Ncand} \times \mathtt{Cost}_{\mathtt{comp}} \tag{6}$$

Especially when the denominator of SG equation is a small number, more than one correlated path candidate can have the largest SG value. In such a case, the best candidate can be detected by computing path delays for only the candidates having that largest shared grid ratio. It should be noted that this cost must be added to (6). But it is difficult to theoretically represent it because the number of such candidates having the same largest shared grids ratio is unknown a priori. Yet, we take into account this additional path delay cost for all experimental results in Section 6. Also, Table 3 reports the total number of such candidate paths as the last column.

6 Results: Comparison of CCM and SGM

For all experiments in this paper realistic variation model considering both inter-die and intra-die variations with spatial correlations [2] is employed. The benchmark circuits are synthesized for 45nm open cell library of Nangate [10]. The most significant random parameters are taken to be transistor channel length (L_{eff}) and threshold voltage (V_t) as devised in [8]. The $3\sigma/\mu$ ratio of 12% and 20% are assumed for L_{eff} and V_t respectively according to the International Technology Roadmap for Semiconductors (ITRS) report [9]. Well known ISCAS'85 benchmark test circuits [6] are used for the experiments. All computations and simulations are performed on HP z620 workstation with Xeon E5-2620, six-core, 2-GHz processors and 24 GB of RAM. A very small Trojan of one XOR gate is employed to test the limits of the proposed method and see their detection performance.

Table 3. Comparison of CCM with SGM Results

Benchmark	Edge Coverage for CCM	Edge Coverage for SGM	# of Candidates for CCM	# of Candidates for SGM
432	95.7%	92.5%	36,647	580
499	64.5%	56.7%	29,320	521
880	94.7%	92.5%	39,176	1782
1355	96.6%	95.8%	17,200,357	102,781
1908	93.4%	91.5%	5,353,429	19,583
2670	95.2%	93.5%	7,619,334	77,390

We compare the correlation coefficient based method (CCM) in Section 3 with shared grid ratio based method (SGM) proposed in Section 5. Please remind that a covered edge means that any Trojan bigger or equal to one XOR gate inserted to that edge can be detected by the method without requiring golden model. Table 3 shows the comparison results. The first deduction from the table is that both CCM and SGM can almost cover or secure the whole circuit resulting in about 90% edge coverage. This means that the methods can detect any Trojan inserted on any place in 90% of the circuit. Excluding c499, which is an obvious outlier, the edge coverage of SGM even becomes about 93% on the average. In January 2008, Dean Collins, deputy director of DARPAs Microsystems Technology Office and manager for the Trust in IC initiative initiates a hardware Trojan detection contest among three companies: Raytheon, Luna Innovations and Xradia. The Trojan circuit is inserted by MIT Lincoln Labs. Collins states to IEEE Spectrum magazine that the goal is a 90 percent detection rate [1], which confirms the sufficiency of 90% coverage.

The last two columns shows the number of correlated path candidates that must be examined for CCM and SGM respectively. The number of candidates is more than 100 times less for SGM because it eliminates all candidates except the ones having the largest shared grids (`SG`) ratio for each suspected path. This

table shows that SGM does not lose accuracy although it performs path delay computations for a much smaller set of correlated path candidates.

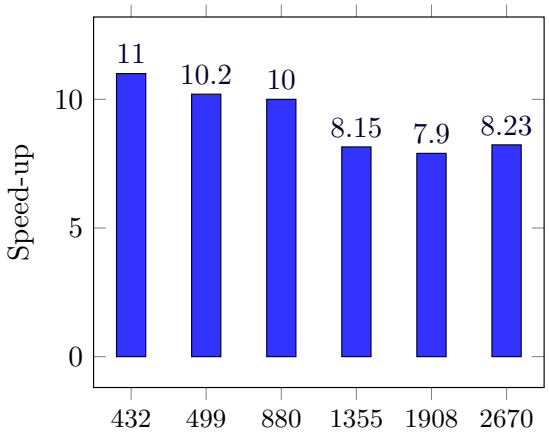

Fig. 6. Stage (iii) speed-up of SGM over CCM

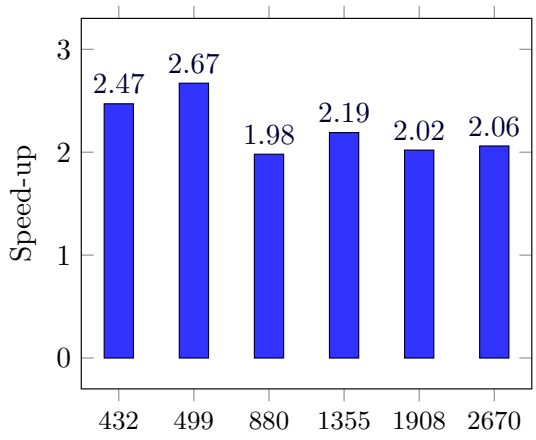

Fig. 7. Complete speed-up by SGM over CCM

As Table 2 suggests the most time consuming part of the spatial correlation based HT detection by CCM is stage (iii). This is why SGM is proposed to speed up that stage. To quantify the speed-up by SGM over CCM at stage (iii) computations, we have recorded the time required for the computation of stage (iii) by both methods. Fig. 6 plots the resultant stage (iii) speed-up for each

benchmark circuit as a bar graph. SGM accelerates stage (iii) of CCM about 9 times on the average over test circuits, which is a serious speed improvement. Please note that all additional path delay computations due to the candidates shown at the last column of Table 3 are taken into account at the speed-up values of Fig. 6 and 7.

The resultant speed-up of SGM over CCM considering the total time for all three stages (from (i) to (iii)) is shown in Fig. 7. When executed on full circuit the proposed SGM can double the speed of the CCM on the average over benchmark circuits. More precisely, SGM achieves about 100% speed improvement with only 3% edge coverage reduction, which shows the efficiency and accuracy of the proposed method.

7 Discussion & Future Work

The spatial correlation based HT detection proposed in [8] is adapted to full circuit and for the first time full circuit results are presented in this paper. The method is accelerated by introducing shared grids ratio instead of correlation coefficient computation. The computational cost analysis of both methods shows the efficiency comparison as well as the empirical results, which show that both methods can secure more than the 90% of the circuit. But usage of shared grids can increase the speed of the whole method more than twice on the average.

Although the method is applied and tested on combinational circuits, it can be generalized to sequential circuits by the help of the techniques like enhanced-scan delay tests [7]. To further accelerate the method, primarily parallelization by GPU utilization can be used. Because, especially stage (ii) and stage (iii) are suitable for distributed computation.

The method in this paper is developed with a focus on improving pre-manufacturing phase and especially to speed up stage (iii), i.e. post-manufacturing tests. However, due to stage (iv), it may take a lot of time to obtain path-delay tests. In other words, the aim of this paper is to decrease the required time to extract path pairs, yet the improvement of stage (iv) requires the extraction of less number of path pairs, which can be a scope of another paper.

References

1. Adee, S.: The hunt for the kill switch. iEEE SpEctrum **45**(5), 34–39 (2008)
2. Agarwal, A., Blaauw, D., Zolotov, V.: Statistical timing analysis for intra-die process variations with spatial correlations. In: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design. p. 900. IEEE Computer Society (2003)
3. Banga, M., Hsiao, M.S.: A region based approach for the identification of hardware trojans. In: 2008 IEEE International Workshop on Hardware-Oriented Security and Trust. pp. 40–47. IEEE (2008)
4. Bayrakci, A.A.: Stochastic logical effort as a variation aware delay model to estimate timing yield. INTEGRATION, the VLSI journal **48**, 101–108 (2015)

5. Bhasin, S., Danger, J.L., Guilley, S., Ngo, X.T., Sauvage, L.: Hardware trojan horses in cryptographic ip cores. In: 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. pp. 15–29. IEEE (2013)
6. Brglez, F.: A neural netlist of 10 combinational benchmark circuits. Proc. IEEE ISCAS: Special Session on ATPG and Fault Simulation pp. 151–158 (1985)
7. Bushnell, M., Agrawal, V.: Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits, vol. 17. Springer Science & Business Media (2004)
8. Esirci, F.N., Bayrakci, A.A.: Hardware trojan detection based on correlated path delays in defiance of variations with spatial correlations. In: Proceedings of the Conference on Design, Automation & Test in Europe. pp. 163–168. European Design and Automation Association (2017)
9. ITRS Commitee: International technology roadmap for semiconductors (itrs) 2011 report. http://www.itrs2.net/2011-itrs.html
10. Nangate: 45nm open cell library. http://www.nangate.com/
11. Narasimhan, S., Du, D., Chakraborty, R.S., Paul, S., Wolff, F.G., Papachristou, C.A., Roy, K., Bhunia, S.: Hardware trojan detection by multiple-parameter side-channel analysis. IEEE Transactions on computers **62**(11), 2183–2195 (2012)
12. Nowroz, A.N., Hu, K., Koushanfar, F., Reda, S.: Novel techniques for high-sensitivity hardware trojan detection using thermal and power maps. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **33**(12), 1792–1805 (2014)
13. Rai, D., Lach, J.: Performance of delay-based trojan detection techniques under parameter variations. In: 2009 IEEE International Workshop on Hardware-Oriented Security and Trust. pp. 58–65. IEEE (2009)
14. Tehranipoor, M., Koushanfar, F.: A survey of hardware trojan taxonomy and detection. IEEE design & test of computers **27**(1), 10–25 (2010)
15. Tehranipoor, M., Wang, C.: Introduction to hardware security and trust. Springer Science & Business Media (2011)
16. Yoshimizu, N.: Hardware trojan detection by symmetry breaking in path delays. In: 2014 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST). pp. 107–111. IEEE (2014)