A Zero-Overhead IC Identification Technique Using Clock Sweeping and Path Delay Analysis

Nicholas Tuzzio, Kan Xiao, Xuehui Zhang, and Mohammad Tehranipoor
Dept. of Electrical & Computer Engineering, University of Connecticut
Storrs, CT, USA
npt05001,kanxiao,xhzhang,tehrani@engr.uconn.edu

Abstract

The counterfeiting of integrated circuits (ICs) has become a major issue for the electronics industry. Counterfeit ICs that find their way into the supply chains of critical applications can have a major impact on the security and reliability of those systems. This paper presents a new method for uniquely identifying ICs through path delay analysis. There is no overhead in terms of area, timing, or power for this method, since it extracts the intrinsic path delay variation information of the IC. Simulation results from 90nm technology and experimental results from 90nm FPGAs demonstrate the effectiveness of our technique.

Categories and Subject Descriptors

B.7 [**Integrated Circuits**]

Keywords

IC identification, counterfeit ICs, process variations, clock sweeping, path delay analysis.

1. INTRODUCTION

A counterfeit IC is an electronic component whose material, performance, or characteristics are knowingly misrepresented by the vendor, supplier, distributor or manufacturer [1] [2]. They can be parts which have been remarked to resemble different parts, defective parts diverted from disposal and sold, or previously used parts salvaged from scrap electronics [1]. We will focus on a subset of counterfeit ICs that physically and functionally appear to be the correct IC, which are extremely difficult to detect.

According to one estimate, the United States Department of Defense may have purchased between $15-100M USD worth of counterfeit ICs in 2005 alone [3]. If counterfeit ICs were to end up in the supply chain for mission-critical or life-saving applications, the results of the failure of an unreliable or insecure counterfeit part could be catastrophic. To prevent these issues, we propose a novel technique for uniquely identifying ICs without the use of specialized hardware structures. Unique identification helps prevent these issues because it allows us to differentiate between counterfeit ICs and authentic ICs with a high degree of success. Because this method does not require any modifications to the hardware it will be used on, it can be used on both new IC designs and legacy designs already in production or in use.

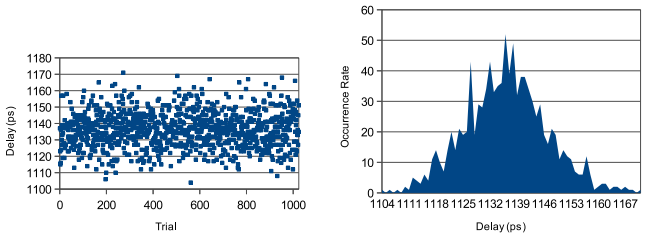
Several techniques have been developed to combat IC counterfeiting. A technique for creating unique identifiers by exploiting an IC's internal process variation was described in [4]. Silicon Physical Unclonable Functions (PUFs) were developed in [5], and many types of PUF have been proposed for ASICs and FPGAs [6] [7]. Another approach to combating IC counterfeiting are metering schemes. Passive metering involves identifying and tracking ICs throughout their lifetime [8]. Active metering prevents an IC from functioning properly until certain activation criteria are met [9]. Work has been done to create unique IC identifiers without any additional hardware. In [10], the authors describe a technique for uniquely identifying ICs by analyzing the unique leakage currents of a series of gates in a circuit.

None of these techniques represent a comprehensive solution to the IC counterfeiting problem. PUF-based identification has yet to be widely accepted due to their reliability issues and their area and logic overhead. They also require high levels of process variation to be present in the circuit and cannot always be inserted in legacy designs. Metering schemes are subject to these problems as well. The method for identifying ICs without additional hardware presented in [10] depends on complex and time-consuming computations and current-measuring techniques which are not often used in industrial tests.

In this paper, we describe a technique for uniquely identifying ICs without any area overhead, and without using uncommon steps in the design or test processes. This technique uses path delay information to create unique binary identifiers. We use a method called "clock sweeping" to obtain this path delay information during testing. We can analyze our ability to accurarely identify ICs under measurement and environmental noise. This technique represents a novel improvement on existing ideas for several reasons. First, our technique can be applied to ICs already in production, including legacy designs. Second, it uses data that can be obtained through use of existing pattern sets and testing hardware capabilities. Third, no additional hardware is necessary- there is no area, power, or timing overhead to the technique. Finally, because the circuit is not added to or modified in any way, no attacks on the circuit are possible.

This paper will be organized as follows: Section 2 will present background theory for our methods. Section 3 will describe the ID creation and IC identification methodologies. Section 4 will present simulation results and experimental results from an FPGA implementation. We will conclude with final remarks in Section 5.

GLSVLSI'12, May 3–4, 2012, Salt Lake City, Utah, USA.



(a) Scatter plot (b) Distribution plot

Figure 1: Path delays for 1,024 simulations of the most critical path in s38417.

2. THEORY AND BACKGROUND

2.1 Process Variations and Path Delay

In theory, the specifications and functionality of different ICs of the same design should be identical. In practice, this is not the case. This happens because of our inability to accurately create IC structures at smaller technology nodes, due to process variations. Thus, values such as the threshold voltage of a transistor (V_{th}), the length of the gate of the transistor (L), or the oxide thickness of the transistor (T_{ox}) can only be guaranteed to be within some range of values. Variation on these and other parameters is important to take into consideration, because these parameters directly affect device performance. For example, the delay of a traditional CMOS inverter can be expressed in two equations, where the high-to-low and low-to-high propagation delays t_{PHL} and t_{PLH} are effected by variation on the three previously mentioned parameters.

Figures 1a and 1b show how process variations can affect the delay of a path in a set of ICs. We measured the delay of the most critical path in the ISCAS'89 benchmark s38417 1,024 times at 25°C using HSPICE. Differing levels of process variations between the Monte Carlo simulations resulted in the path having a different propagation delay in each simulation. Figure 1a shows the path delay for each simulation, and Figure 1b shows the distribution of these path delays. These delays are centered around an average value because process variation results in small changes to the parameters which control the path delay.

2.2 Clock Sweeping

"Clock sweeping" is the process of applying patterns to a path multiple times with different frequencies to find a frequency at which the path cannot propagate its signal, often for purposes of speed binning. By observing the frequencies at which the path can and cannot propagate its signal, we can measure the delay of the path with some degree of precision. Our ability to perform clock sweeping on a path is limited by the degree of control we have on the clock that controls the capturing memory elements (i.e., the flip-flops), the degree to which we can excite paths in the circuit, and the lengths of the paths in the IC.

Figure 2 shows a visual example of clock sweeping being performed on several paths. Assume that paths P1 through P8 are paths in the circuit which end with a capturing flip-flop, and have some delay in nanoseconds. Each of the eight paths can be swept, or tested, at the frequencies f_1 through f_5. All paths are able to propagate their signal at f_1, as this is the rated frequency of the IC design. However, at f_2, the path P3 will usually fail to propagate its signal. At frequency f_3, path P3 will always fail to propagate its signal. Path P8 will succeed in propagating its signal at all five clock frequencies- in this example, it is too short to test. All of the paths have some number of frequencies they will pass at, some they may fail at, and some they are guaranteed to fail at. Process variations change which frequency each path will fail at between different ICs.

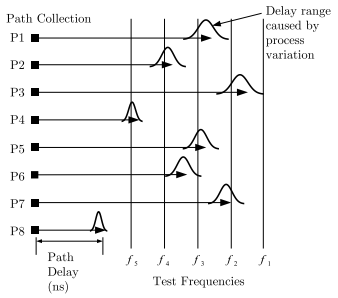


Figure 2: A visual example of clock sweeping being performed on several paths.

3. METHODOLOGY

Our method for uniquely identifying ICs can be broken down into three main parts: (a) Test preparation, (b) ID generation and optimization, and (c) IC identification.

3.1 Test Preparation

The data that we will use to create unique IDs for every IC will be the path delay information collected from the ICs. This data can be collected during the manufacturing test process, but this requires some additional preparation. We can collect this information through application of transition-delay fault (TDF) patterns. TDF patterns seek to identify slow-to-fall or slow-to-rise patterns and as such must be run with a specific clock frequency in mind; by applying the same TDF patterns at different frequencies (f_2, f_3, etc.), we perform the clock sweeping process described above.

Figure 3a shows the pattern generation procedure. We will obtain path delay information using the TDF patterns that would be generated anyways during for manufacturing tests. Some of the TDF patterns will be applied multiple times at different frequencies. However, not all patterns need to be tested at all test frequencies. Consider path P2 in Figure 2. If the circuit is fault-free, path P2 will pass tests at the frequencies f_1 through f_3. It will either fail the test at f_4 or f_5. Thus, if P2 passes tests at f_1, we can call the circuit fault-free, and we only need to perform the additional tests for f_4 and f_5 for identification purposes. We can save time by not testing the circuit at f_2 or f_3. By carefully selecting the set of frequencies that each pattern needs to be applied at, we can significantly reduce the pattern application time and thus the indentification time. Once the test preparation step has been completed, and TDF patterns for the different sets of frequency tests have been created, ICs can use these tests to generate unique IDs.

3.2 ID Generation and Optimization

Once the tests have been applied in the manufacturing test stage, we possess a large amount of path delay information from each manufactured ICs. We will use this data to generate a binary ID for each IC. Three steps are performed to generate the IDs for each IC: (1) Stability checking, a path selection procedure that selects the most stable measured paths from each IC. (2) ID generation, a path delay comparison procedure which generates an unoptimized ID for each IC. (3) ID optimization, an analysis and optimization procedure which increases the randomness and decreases the size of the IDs.

Stability Checking: We are using path delay information to generate IDs. Path delay measurements using clock sweeping do have the potential to be affected by measurement and environmental noise. To reduce the effect of this issue, we will attempt to discard unstable paths early on. Measuring an IC multiple times gives us multiple measurements of each path in that IC. Analysis of multiple measurements of a path can tell us whether that path is stable- it reported the same delay during each measurement- or unstable- reporting differ-

ent delays on different occasions. To select stable paths, we will measure some subset of the IC set multiple times and select the m most stable paths to be used for further analysis on all n ICs.

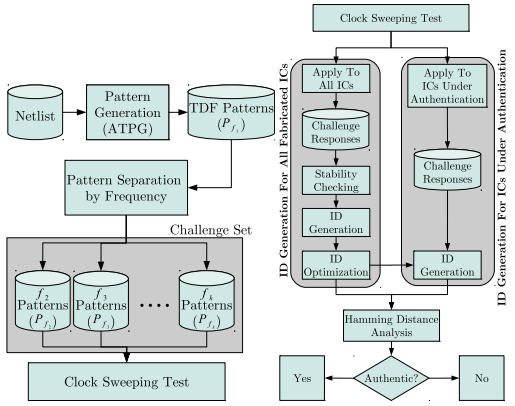
ID Generation: Once the m most stable paths have been selected, we begin to generate IDs for each IC. To generate an ID, we will perform comparisons between the delays of different paths from the ICs. The goal is to perform a comparison that is somewhat uncertain, so that the result the comparison generates is random. We obtain a list of similar paths by sorting a list of m paths from an IC in ascending or descending order. We will call this sorting the "golden ranking", and apply this ordering to the m paths of all n ICs. Once the golden ranking has been obtained, the path delay data for every IC is sorted by this ordering. For each IC, we will traverse this ordered list of delays one element at a time. Whenever a delay in the list is greater than the next delay in the list, we will append a "1" onto the ID for that IC. Whenever a delay in the list is less than the next delay, we will append a "0" onto the ID for that IC. If m paths have been analyzed for each of the n chips, we will end up with n IDs of $m-1$ bits each. The first/golden IC, which was used to determine the ordering, will have an ID of all ones or zeros, depending on whether the sort was ascending or descending.

ID Optimization: There are several optimization techniques we can use on these ICs. Some paths that we are comparing in the ID Generation step may produce the same value for all ICs because the path delays are very different. This will reduce the average Hamming distance between IDs as this bit will be the same in all ICs. If enough paths have been measured, it may be possible for an IC designer to pick and choose which comparisons they are going to perform to create their IDs. The comparisons can be chosen by analyzing each bit of the $m-1$ bit "unoptimized IDs", and deciding which bits have the least bias across all IDs. A bit should be included in the "optimized" ID if the number of zeros in that bit position is close to 50% of the number of IDs- if the number of zeros in that bit position is in the range of $\frac{n}{2} \pm \varepsilon$, where ε is some small constant. This process selects the most unpredictable comparisons across the IC data set. By performing this optimization technique, we obtain with shorter, more random IDs, at the cost of discarding some of the data obtained during the measurement process.

3.3 IC Identification

This process will be run on every IC when they are first manufactured and going through manufacturing tests. These tests will produce an ID for every manufactured IC, will will be stored in a database for later access. Parts of this process will be run again later when we wish to identify an IC. Figure 3b shows these two scenarios. In order to identify an IC from the market, referred to as an IC under authentication (IUA) in Figure 3b, we will need: (1) the ID that the IC is presenting, and (2) the database of IDs from when the ICs were manufactured. Once we have the ID for the IUA, we can check to see if it is in the database using Hamming distance analysis. Hamming distance analysis is done by comparing the ID from the IUA to every ID in the database. If the ID from the IUA is an exact match to one of the IDs in the database, then we have identified the CUT.

Noise in the measurement process might make it so that there are a small number of bits in the ID which are different between the IC's ID in the database and its ID in the field. We addressed this issue through the stability checking technique described in Section 3.2, but differences may still be present. Therefore, we define a *Hamming distance threshold*, based on analysis of the IDs during manufacturing tests. This threshold could be based off the minimum Hamming distance between two different ICs in the manufactured set. The threshold could also be based off of the maximum Hamming distance between IDs from the same IC, which would be created during the stability checking process. Exact thresholds would be derived from analysis of the ID set and measurement noise rates.



(a) Test generation flow. (b) Authentication flow.

Figure 3: Test generation and authentication flow.

3.4 Overhead Analysis

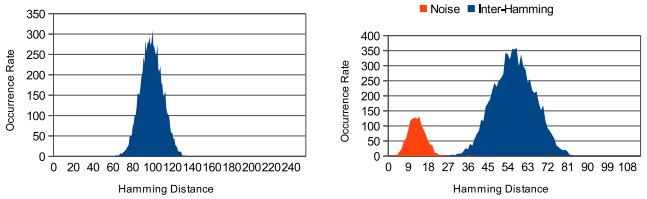
This IC identification procedure incurs no area, power, or timing overhead. There is an overhead in testing time and computational time. Each IC must undergo an additional k rounds of TDF testing at k different frequencies. If we have n ICs with P_{f_1} TDF patterns and N_{sff} flip-flops that we would like to test at k frequencies, the worst case increase in testing time would be $n \cdot (k-1) \cdot P_{f_1} \cdot N_{sff} + N_{sff}$ clock cycles. This is a worst-case scenario as not all P_{f_1} TDF patterns will need to be tested at all k frequencies, as shown in Figure 3a. The stability checking procedure requires us to measure a subset of $n_r \ll n$ ICs j times each, so there is a testing overhead of $n_r \cdot j \cdot (k-1) \cdot P_{f_1} \cdot N_{sff} + N_{sff}$ clock cycles in the worst-case scenario. Judging the relative stabilities of each path in the ICs requires us to individually analyze each of the m paths that were measured j times each in n_r ICs, resulting in a complexity of $O(n_r m j)$ time; however, n_r is much less than n, and j is going to be on the order of dozens to hundreds. The ID generation procedure requires us to individually analyze each of the m paths in each of the n ICs, resulting in an $O(nm)$ runtime. The ID optimization technique analyzes each of the $m-1$ bits across all of the n IDs for a general runtime of $O(nm)$ as well. In general, m will be much smaller than n or n_r.

4. RESULTS AND ANALYSIS

4.1 Simulation Results

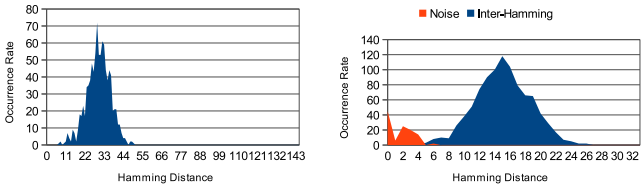
To evaluate this methodology, we performed a series of simulations using HSPICE. The simulations were performed on an implementation of the ISCAS'89 benchmark s38417. This circuit was simulated at the 90nm technology node. To measure the path delay, we identified the 256 top critical paths from the circuit, and performed 128 Monte Carlo simulations. This provides us with 128 simulated ICs worth of data, with 256 data points for each IC. During the Monte Carlo simulations, the threshold voltage V_{th} had at most 5% inter-die variation and 5% intra-die variation, the oxide thickness T_{ox} had at most 2% inter-die variation and 1% intra-die variation, and the transistor gate length L had at most 5% inter-die variation and 5% intra-die variation, with these values representing 3-sigma deviations from specified values. All paths for all ICs were simulated over a range of temperatures from 23°C to 27°C to represent small deviations from room temperature.

We used the simulation data to create 128 255-bit IDs using the methodology described in Figures 3a and 3b. Figure 4a shows the



(a) Unoptimized IDs (b) Optimized IDs with noise

Figure 4: Hamming distance analysis on 128 simulated s38417 circuits.



(a) Unoptimized IDs (b) Optimized IDs with noise

Figure 5: Hamming distance analysis on 44 FPGA s38417 circuits.

results of Hamming distance analysis on these unoptimized IDs. The 255-bit IDs have, on average, a 99-bit or 39% inter-Hamming difference. The fact that this average is less than 50% means that some bit-positions in the IDs have a bias towards zero or one. By optimizing the IDs to remove these bits, using the optimization procedure described in Section 3.2, we improve the average Hamming distance of the ID set and reduce the size of the IDs, as shown in Figure 4b. The optimization process reduced the size of the IDs from 255 bits to 114 bits, and increased the average inter-Hamming distance from 39% to 50%.

Figure 4b also shows the noise rates of the simulated IDs. Each IC was simulated at temperatures from 23°C to 27°C at 1°C increments to represent small variations around room temperature. We performed Hamming distance analysis on each set of IDs coming from the same IC to see how many bits of the ID would change over the ID range. The average number of bits changing as a result of temperature was 13 bits, or about 11%.

The Hamming distance distribution and noise rate distribution are both roughly normal distributions. By treating them both as normal distributions with their own averages and standard deviations, we can come up with probability density functions (PDF) for each set. We can use these two PDFs to find the Hamming distance at which it is equally likely that the two IDs are from different ICs or from the same IC. In the sets shown in Figure 4b, this Hamming distance is 27 bits. If two IDs are compared and have less than 27 different bits between them, they are most likely from the same IC. If two IDs are compared and have more than 27 different bits between them, they are most likely from different ICs.

4.2 FPGA Implementation Results

We implemented this methedology on 44 90nm Xilinx Spartan-3E FPGAs. We implemented the ISCAS'89 benchmark s9234 on our FPGAs, along with RAM for TDF pattern storage and structures for clock control. The IDs were sent from the FPGA boards to a computer for analysis by using an additional microcontroller. The paths in the FPGA were swept at the frequencies f_{1-16}, with $f_1 = \frac{1}{6.4ns}$ and $f_{16} = \frac{1}{3.4ns}$. The frequency step size was 200ps. To obtain noise information, we measured 4 of the implementations 8 times each. After selecting stable paths, the unoptimized IDs were 145-bit strings, with a Hamming distance distribution as shwon in Figure 5a. The average inter-Hamming distance was 30 bits, or about 21%. This is lower than in the simulation example. The optimization process reduces the size of the ID from 145 to 33 bits long, and the distribution of the inter-Hamming distances between these 33-bit IDs are shown in Figure 5b. By optimizing the IDs, we change the average inter-Hamming distance from 30 out of 145 bits to 15 out of 33 bits, or about 45%. In practice, a 33-bit ID would be short, but creating IDs of a greater size only requires measuring more paths.

Figure 5b also shows the noise rates for these IDs. We measured 4 different implementations of s9234 8 times each and compared the IDs from each implementation to each other. On average, there was a 2 out of 33 bit difference between IDs from the same implementation, or about a 6% difference.

Again, we can perform further analysis to compare the noise and inter-Hamming rates. At a difference of 5.75 bits, it is equally likely that any two IDs being compared are from the same IC or from different ICs. IDs with a difference of less than 5.75 bits are probably from the same IC and IDs with a difference of more than 5.75 bits are probably from different ICs.

5. CONCLUSION

In this paper, we proposed a technique that allows us to uniquely identify ICs without any area, power, or timing overhead. In both the simulation and implementation results, we were able to create unique IDs with no collisions for a set of different implementations of the same circuit. In both cases, the average Hamming distance between the IDs was nearly 50%, and the levels of noise were sufficiently low that we could distinguish between IDs from the same IC and IDs from different ICs. These results would improve with a larger data set, a more accurate clock with a smaller frequency step size, and larger levels of process variations.

6. ACKNOWLEDGEMENTS

This work is supported in part by grants from National Science Foundation (NSF) under CNS 0844995 and Army Research Office (ARO).

7. REFERENCES

[1] H. Livingston, "Avoiding counterfeit electronic components," *Components and Packaging Technologies, IEEE Transactions on*, vol. 30, pp. 187 –189, march 2007.

[2] M. Tehranipoor and C. Wang, *Introduction to Hardware Security and Trust*. Springer, 2012.

[3] J. Stradley and D. Karraker, "The electronic part supply chain and risks of counterfeit parts in defense applications," *Components and Packaging Technologies, IEEE Transactions on*, vol. 29, pp. 703 –705, sept. 2006.

[4] K. Lofstrom, W. Daasch, and D. Taylor, "Ic identification circuit using device mismatch," in *Solid-State Circuits Conference, 2000. Digest of Technical Papers. ISSCC. 2000 IEEE International*, pp. 372 –373, 2000.

[5] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, (New York, NY, USA), pp. 148–160, ACM, 2002.

[6] G. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pp. 9 –14, june 2007.

[7] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems*, CHES '07, (Berlin, Heidelberg), pp. 63–80, Springer-Verlag, 2007.

[8] F. Koushanfar and G. Qu, "Hardware metering," in *Design Automation Conference, 2001. Proceedings*, pp. 490 – 493, 2001.

[9] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, (Berkeley, CA, USA), pp. 20:1–20:16, USENIX Association, 2007.

[10] Y. Alkabani, F. Koushanfar, N. Kiyavash, and M. Potkonjak, "Information hiding," ch. Trusted Integrated Circuits: A Nondestructive Hidden Characteristics Extraction Approach, pp. 102–117, Berlin, Heidelberg: Springer-Verlag, 2008.