TOIC: Timing Obfuscated Integrated Circuits

Mahabubul Alam
Electrical Engineering
Pennsylvania State University
State College, PA
mahabubul.alam@psu.edu

Swaroop Ghosh
Electrical Engineering
Pennsylvania State University
State College, PA
szg212@psu.edu

Sujay S. Hosur
Electrical Engineering
Pennsylvania State University
State College, PA
ssh29@psu.edu

ABSTRACT

To counter the threats of reverse engineering (RE) and Trojan insertion, researchers have considered gate-level obfuscation in integrated circuits (IC) as a viable solution. However, several techniques are present in the literature to crack the obfuscation with varying degree of success raising the concern about their secrecy. In this article, we have presented TOIC (Timing Obfuscated Integrated Circuits), a novel technique where sequential elements are obfuscated to hide the true timing paths in the design. TOIC can act as a standalone countermeasure against IC reverse engineering or can be incorporated with existing gate camouflaging techniques to maximize adversarial RE effort. Previous research has shown that limiting access to internal nodes can improve the adversarial RE effort at the cost of poor testability. *TOIC can impose prohibitively large decamouflaging time complexity by limiting the controllability and observability over the internal nodes in an IC while preserving complete testability.*

KEYWORDS

Reverse Engineering, Camouflaged Flip-Flop, Threshold-defined Switch, SAT, Timing Obfuscation.

ACM Reference Format:
Mahabubul Alam, Swaroop Ghosh, and Sujay S. Hosur. 2019. TOIC: Timing Obfuscated Integrated Circuits. In *Great Lakes Symposium on VLSI 2019 (GLSVLSI '19), May 9–11, 2019, Tysons Corner, VA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3299874.3318001

1 INTRODUCTION

Reverse engineering (RE) of integrated circuits (IC) is the process of gaining a full understanding of the functionality to a point where it can be manufactured with little effort. *In optical RE, an attacker delayers the chip, take images of each layer and use image processing techniques to reconstruct the functional netlist of the IC from these images [1].* Vendors e.g., Chipworks and Degate advertise their ability to perform these tasks. Smaller ICs that are manufactured with mature technology nodes are at a higher risk of being reverse engineered as they are easy to delayer. *However, recently Holler et. al have successfully constructed the layout of a silicon ASIC (Intel 22nm) in a non-invasive manner using X-ray ptychography [2].* With such capabilities available to an attackers disposal, it is gradually becoming easier to reverse engineer and manufacture ICs with no design cost.

Current State-of-the-Art: Researchers have considered gate obfuscation as a viable option to thwart RE. In a gate obfuscation, the layout of the gates are designed in such a fashion that the true functionality is hidden from an attacker during optical RE [3]. However, SAT-based and VLSI test-based attacks have been proven successful against obfuscation of combinational design irrespective of the obfuscation methodology chosen or the number of obfuscated gates in the design [4, 5]. *Although the validity of such attacks on large-scale sequential circuits is debate-able, the authors have justified the attacks based on the following argument: for any sequential design with full scan-access, we can divide the design into multiple camougflaged combinational logic blocks (CCLB) from one sequential stage to another, and decamouflage the CCLB′s with the proposed attack methodologies.* The flow to RE a gate obfuscated sequential design is shown in Figure 2. The camouflaged gates do not affect the controllability and observability provided by the scan chain which makes this technique vulnerable to these query based attacks (SAT/VLSI test).

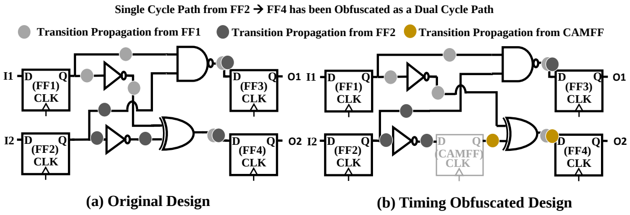

Single Cycle Path from FF2 → FF4 has been Obfuscated as a Dual Cycle Path
Transition Propagation from FF1
Transition Propagation from FF2
Transition Propagation from CAMFF
I1
D
Q
(FF1)
CLK
(FF3)
O1
I2
(FF2)
(FF4)
O2
(CAMFF)
(a) Original Design
(b) Timing Obfuscated Design

Figure 1: (a) Oracle sequential circuit; (b) Timing obfuscated circuit.

In [6], Zhang et al. proposed a technique to obfuscate the timing between successive flip-flops (FF) using unconventional timing. Using the concept of wave pipelining, they have shown that timing paths can be created in the design that looks like a single cycle timing path in a netlist, however, they work as a multi-cycle paths in the chip. Flip-flops are removed from the design selectively to create the multi-cycle paths. In modern chip design, flip-flops provide the desired controllability and observability over the internal nodes through scan access which is an integral part of post-fabrication chip testing and validation. The removal of flip-flops can lead to reduced controllability and observability over the design and significant increase in test time. As very few SAT-based attacks on sequential circuits are known, the timing obfuscation technique was thought to be resistant against these attacks. However, Li et al. showed that with a little modification in the netlist (circuit unrolling), decamouflaging attacks (as shown in Figure 2) can be applied to the timing obfuscated circuit that provides full scan access to an attacker [7]. In [8], Karmakar et al. proposed an encrypted flip-flop which can have inverted or non-inverted output in the fabricated chip based on a secret key. The design is vulnerable to path sensitization attack if the input of the flip-flop is directly accessible through the scan access or any primary input of the chip. Protection from such attacks requires additional scan-chain locking strategies at the cost of significant design overhead. Moreover, the design requires additional secure NVM storage to protect the secret key which can significantly increase the cost of the chip.

Timing Obfuscation: In digital circuits, FFs operate as sequential elements to hold intermediate states between operations. The combinational arc between any two FFs is called a timing path. Data has to travel from start to end of a timing path within a single clock period. In a pipelined design, even a single failing path can corrupt the data transfer between successive stages. As a result, the output of the pipeline can be corrupted.

For a gate obfuscated sequential circuit, all the timing arcs are known to an attacker. *In this paper, we have proposed a novel circuit*

GLSVLSI '19, May 9–11, 2019, Tysons Corner, VA, USA

ACM ISBN 978-1-4503-6252-8/19/05…$15.00
https://doi.org/10.1145/3299874.3318001

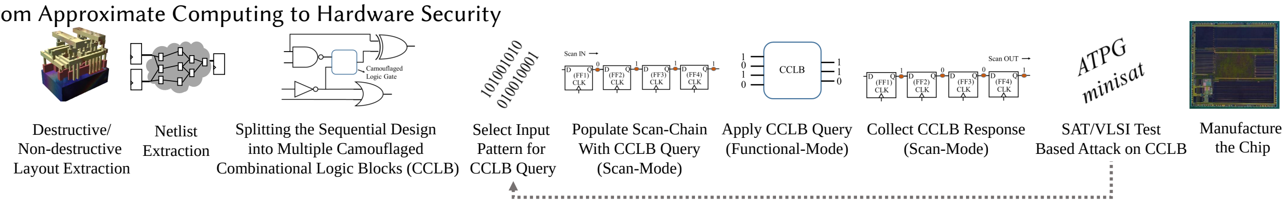

Camouflaged Logic Gate
1010010101
0100100001
Scan IN
(FF1) CLK
(FF2) CLK
(FF3) CLK
(FF4) CLK
CCLB
Scan OUT
ATPG
minisat
Destructive/ Non-destructive Layout Extraction
Netlist Extraction
Splitting the Sequential Design into Multiple Camouflaged Combinational Logic Blocks (CCLB)
Select Input Pattern for CCLB Query
Populate Scan-Chain With CCLB Query (Scan-Mode)
Apply CCLB Query (Functional-Mode)
Collect CCLB Response (Scan-Mode)
SAT/VLSI Test Based Attack on CCLB
Manufacture the Chip

Figure 2: Reverse engineering of gate camouflaged sequential circuits.

obfuscation technique where the actual timing paths are obfuscated through the use of camouflaged FFs. We call it timing obfuscation. We have presented a design for a camouflaged FF which can act as a sequential or a non-sequential delay element in the circuit based on the threshold voltage (V_t) of a few selected NMOS pass transistors. *Replacing some FFs with camouflaged sequential elements, and sparingly deploying non-sequential camouflaged delay elements, we can obfuscate the true timing paths in the design from an attacker.*

In Figure 1(a), the following timing paths exist: FF1→FF3, FF1→FF4, FF2→FF3, and FF2→FF4. FF1→FF3 and FF2→FF3 interfere with each other as they share a common gate ($NAND2$). Because of the presence of the camouflaged FF ($CAMFF$) (Figure 1(b)), the timing arc between FF2→FF4 has been divided into two possible timing arcs (FF2→CAMFF and CAMFF→FF4). A transition at the output of FF2 will take a single clock cycle to be captured by FF4, while a transition at the output of FF2 will take two clock cycles (instead of one for the original circuit) if $CAMFF$ is assumed to be a flop by an attacker. Therefore, he will get corrupted data at the output. For instance, if we reset all the flops in the circuit and then operate the circuit for one clock cycle, FF3 and FF4 will capture 1 and 0 respectively for the original circuit. For the camouflaged circuit, if $CAMFF$ is considered as a flop, FF3 and FF4 will both capture 1 which is different than the original circuit. *The security of the design will lie on the difficulty for an attacker to identify the true and dummy sequential elements in the design.*

Contibutions: (a) Three novel $CAMFF$ designs have been presented and analyzed under temperature and process variations. (b) A novel circuit obfuscation technique where the true timing paths are obfuscated from an attacker through the use of $CAMFF's$. (c) Analysis of the $CAMFF$ in the design and scan-chain. (d) Analysis of the security of TOIC based on several known and expected attacks. (e) A secure scan architecture based on the attack scenarios.

The rest of the article is organized as follows: we have discussed the design of the $CAMFF$ in Section 2. The design obfuscation strategies with camouflaged flip-flops are discussed in Section 3. The security assessment of TOIC is presented in Section 4. Finally, we have presented the simulation results in Section 5.

2 PROPOSED CAMOUFLAGED FLIP-FLOPS

Basic Building Block: In modern planar MOS fabrication technology, the threshold voltage of individual transistors in the design can be controlled by manipulating the doping concentration [10]. The technique can be used to realize threshold voltage defined switches (TDS). The NMOS switch can be fashioned by using either a high threshold voltage (V_{HVT}) or low threshold voltage (V_{LVT}) NMOS transistor biased at a voltage in between those two thresholds (V_{SN}) (shown in Figure 3(a)). If LVT is assigned to the switch during fabrication, it will conduct in the presence of the bias voltage (V_{SN}). For HVT, the switch will always be in the cut-off region. Using threshold defined PMOS and NMOS switches together, we can also build threshold defined CMOS switch ($TDCS$) as shown in Figure 3(f). Both the NMOS and PMOS switches are asserted LVT during fabrication for closed switch operation and HVT for open switch operation in the chip.

Although the TDS switches are easily identifiable in a layout, it is not straightforward for an attacker to identify the threshold voltages. *Threshold voltages of each of the transistors cannot be determined from the images of a delayered chip. While useful, microprobing on thousands of transistors in a large scale integrated circuit is a prohibiting task.* Utilizing the secretive property of the TDS, camouflaged gates are fabricated in 65nm process [11]. In [9], the switches are used to realize threshold voltage defined muxes (TDM) which is shown in Figure 3(b). $D0$ is asserted to the output as the lower V_t of $NMOS1$ turns it on when a voltage higher than its V_t (but less than the V_t of $NMOS2$) is applied to the gate (V_{SN}). TDM and $TDCS$ are the basic building block for the proposed camouflaged FFs.

Design of the Proposed Camouflaged FFs: We have used two and four input $TDM's$ and regular D-flip-flop building blocks (N/P-latches) in $Design-1$ and $Design-2$. $TDCS$ and inverter's with varied driving strengths are used in $Design-3$. In general, all three designs can act as sequential (D-flip-flop) or non-sequential (buffer/inverter) elements in the fabricated chip based on the asserted threshold voltage of selected transistors.

$Design-1$: The $Design-1$ of camouflaged FF is shown in Figure 3(c). We have split the connection between the P-latch and N-latch in a FF and inserted two threshold defined mux ($M1\&M2$) [9] to create 4 logical paths between $Data\_in$ and $Data\_out$. Based on the doping concentration of transistors $M1$ and $M2$, the device can act as a camouflaged FF (CFF), or inverted P-latch (CPL), or inverted N-latch (CNL), or buffer (CB). If LVT is assigned to the NMOS devices at A and C during fabrication, the $CAMFF$ acts as CB. If LVT is assigned to the NMOS devices at B and D, the camouflaged FF acts as CFF.

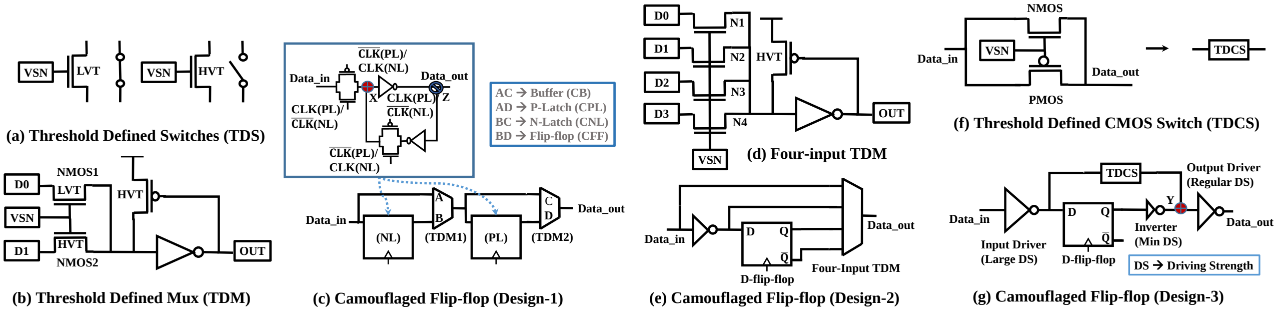


Figure 3: (a) Threshold defined NMOS switches; (b) Pass transistor based 2:1 threshold defined mux [9]; (c) Schematic of the camouflaged FF ($Design-1$); (d) Pass transistor based 4:1 threshold defined mux; (e) Schematic of the camouflaged FF ($Design-2$); (f) Threshold Defined CMOS Switch; (g) Schematic of the camouflaged FF ($Design-3$).

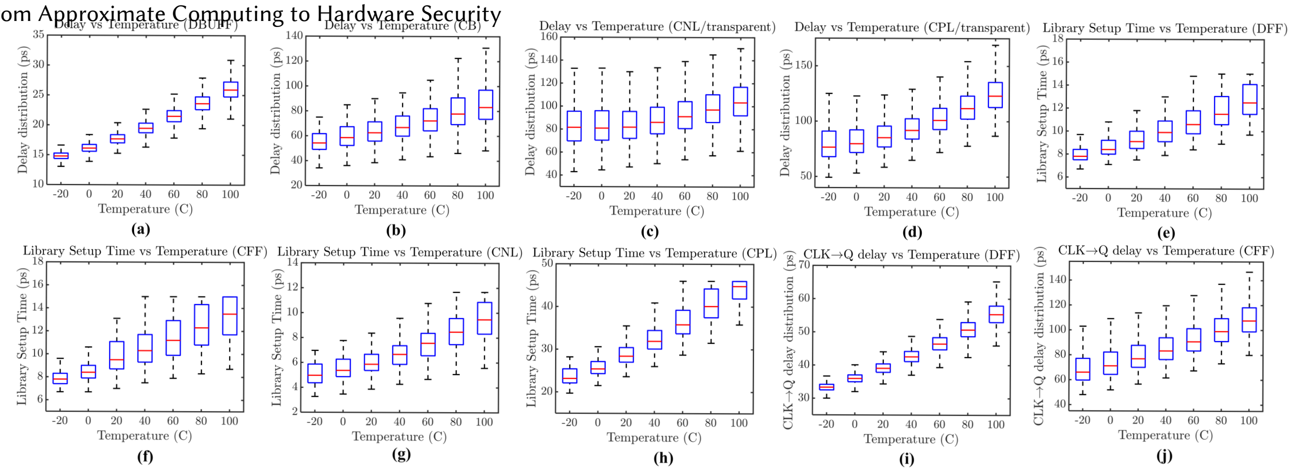


Figure 4: Delay distribution (*Design* − 1) for (a) delay buffer (DBUFF), (b) camouflaged buffer (CB), (c) camouflaged N-latch (CNL), and (d) camouflaged P-latch (CPL); Library setup time distribution for (e) regular flip-flop (DFF), (f) camouflaged flip-flop (CFF), (g) camouflaged N-latch (CNL), (h) camouflaged P-latch (CPL); Clock→Q delay distribution for (i) DFF, and (b) CFF.

Design − 2 : The *Design* − 2 is shown in Figure 3(e). A four-input *TDM* (as shown in Figure 3(d)) is placed in front of a regular D-flip-flop. Based on the assertion of the *LVT* transistor among the four $TDS's$ (N1, N2, N3, and N4) in the four-input *TDM*, the *design* − 2 can act either as a buffer, or an inverter, or a flip-flop, or an inverting flip-flop.

Design − 3 : Regular flip-flops are generally isolated with input and output driver inverters. In our *Design*−3, we have connected an additional inverter between the D-flip-flop output (Q-point) and the output driver with minimum driving strength. The input driving inverter is upsized so that it has significantly larger driving strength than the minimum sized inverter. A *TDCS* is placed between the outputs of the input driver and minimum sized inverter. The gates of *TDCS* are driven by *VSN* (approximately half of the VDD). *TDCS* acts as a closed switch if LVT is asserted during fabrication for both the NMOS and PMOS devices. A contention exists between the input driver and the minimum sized inverter. Larger driving strength of the input driver enables it to overpower smaller inverter, and hence node Y follows the inverted input and the output driver follows the non-inverted input all the time (similar to a buffer). If HVT is asserted, *TDCS* acts as a closed switch and the *Design* − 3 operates as a regular flip-flop with inverted output.

Design Analysis: We have simulated the performance of the camouflaged FF's using 32nm planar MOSFET technology with a nominal supply voltage of 0.9V [12]. The NMOS/PMOS pass transistors in *TDM*/*TDCS* (in Figure 3(b), (d), and (f)) are biased (*VSN*) at 0.5V. The V_t of the LVT NMOS/PMOS and HVT NMOS/PMOS devices are set around 200mV/-200mV and 750mV/-750mV respectively. The CMOS inverter following the NMOS pass transistors in *TDM* is left-skewed to restore the value to CMOS logic levels. The weak keeper PMOS device is used to accelerate 0→1 transition at the inverter input.

Design−1 has an area overhead of nearly 70% than a conventional flip-flop (*DFF*). The average power consumption can be 3X higher (all flip-flop power consumption calculated at 1GHz clock frequency with activity factor of 0.4). The simulated value of the CLK→Q delay distribution for a conventional *DFF* and *CFF* is shown in Figure 4(i)&(j). We have considered 10% variation in V_t and a temperature variation from -20°C to 100°C. On an average, the CLK→Q delay of *CFF* is around 2X higher than *DFF* due to the presence of two *TDM* in the critical path of the flip-flop. The setup time of the *CFF* and *CNL* are similar to *DFF*. However, the *CPL* setup time can be 3X higher. This is because the setup time depends on the master stage delay from the input (D) to node M which remains the same for *CFF*, *CNL*, and *DFF*. For *CPL*, the additional delay for the *TDM* accounts for the large difference in the setup time. The hold time of *CFF*, *CNL*, and *CFF* are similar/smaller than *DFF* values due to the additional diffusion capacitance at node *Z*. In Figure 4, we have summarized the setup time of *CAMFF* under process (10% variation in V_t) and temperature variations for 1000 points monte carlo simulations. The delay of *CB*, *CNL*, and *CPL* (transparent mode) can be 5X higher than conventional delay buffer (Figure 4(a)-(d)).

Design − 2 has a similar area overhead to the *design* − 1. The average power consumption is approximately 2.5X to a regular D-flip-flop. The CLK→Q delay is approximately 50% higher than the regular D-flip-flop. The library setup and hold times are similar to a conventional flop as the master latch stages are identical. The buffer variant has area, power, and delay overhead of approximately 6X, 1.73X, and 1.62X to a conventional delay buffer. The inverter variant has area, power, and delay overhead of approximately 12X, 4.13X, and 1.74X to a conventional inverter.

Design − 3 has an area overhead of approximately 15% to a regular D-flip-flop. The average power consumption is approximately 20% higher than a regular D-flip-flop. The CLK→Q delay is approximately 35% higher than the regular D-flip-flop. The library setup and hold times are similar to a conventional flop. The buffer variant has area, power, and delay overhead of approximately 5X, 12X, and 1.25X to a conventional delay buffer.

3 DESIGN OBFUSCATION WITH CAMOUFLAGED FLIP-FLOPS

In this section, we present the basic strategies to obfuscate the design with camouflaged FFs. We pick *Design* − 3 for the demonstration purpose. We refer its inverted flip-flop variant as *CFF* and buffer variant as *CB* in the subsequent sections. The design obfuscation should be incorporated in a post-synthesized design that already meets all the timing requirements. Any *DFF* in the design can be replaced by a *CFF* as long as the additional CLK→Q delay does not create any setup violations in any timing path involving the *DFF* (fanins and fanouts). As the *CFF* output is inverted, an inverter has to be added to the design as well. A *CB* can be used as a delay element at any node in the design as long as all the timing paths involving that node have safe setup slack. We generally use delay buffers to fix hold violations. Instead, we can use $CB's$ to provide extra security to the design in addition to fixing hold violations. During deobfuscation, any wrong assumption about the identity of the CFF's and CB's in the design may possibly lead to output corruption.

In a sequential design, switching at a node closer to the inputs have a lesser probability to propagate to the primary outputs than the nodes closer to the outputs. Hence, we can insert the camouflaged flops near the end stages of the pipeline (e.g. register-to-out paths) to ensure maximum output corruptibility, and we can put them closer to the inputs (e.g. input-to-register paths) if we want maximum stealth. *Selection of nodes for CAMFF insertion should be driven by the timing constraints of the synthesized design.* The scan-chain of the obfuscated design will have regular flip-flops, camouflaged FFs with inverted outputs, and dummy flops (buffers).

4 ATTACKS AND COUNTERMEASURES

In this section, we asses the security of the obfuscated design with the basic approach and propose countermeasures against various attacks. We assume that an attacker posses the ability to extract the netlist of an obfuscated chip through destructive [1] or non-destructive [2] RE steps. To fully reconstruct the functional netlist, an attacker needs to uncover the true identities of the camouflaged cells. *We also assume that the attacker, (i) has access to the functional chip, (ii) can apply any random set of queries to the chip primary inputs and observe the primary outputs, and (iii) has access to EDA tools for functional simulation of the obfuscated netlist.*

The output of any camouflaged FF in the design will be uncontrollable as long as its true identity is hidden from an attacker. Replacing all the flops in the design with camouflaged flip-flop could make all the internal nodes in the design uncontrollable except those directly accessible through the primary inputs(PI) and/or the scan access. However, it is impractical due to the associated design overhead. Selective camouflaging can leave many nodes in the design controllable for an attacker. He can utilize this advantage to deobfuscate the camouflaged flops in the design which is illustrated in the following example.

Example-1: In Figure 5(a); nodes A, B, C, D, and DFF1-Q are controllable and DFF2-Q is observable for an attacker. CAMFF is a camouflaged FF in the timing path from DFF1 to DFF2. An attacker sets A, B, and D to 0, C to 1, and DFF1-Q to any logic value ('x'), and checks the value captured at DFF2-Q after two clock cycles. If CAMFF is a buffer, DFF2 will capture 1 and if its a flip-flop with inverted output ($Design - 3$), it will capture 0. Using this two-cycle test, an attacker can successfully reveal the identity of the CAMFF as shown in Figure 5(b). For $Design - 1$ and $Design - 2$, dual two cycle tests will be required to deobfuscate the CAMFF which is shown in Figure 5(c).

At this point, we may consider placing camouflaged buffers (CB) at node A, B, C, and D, and replace DFF1 with another CFF to make all these nodes uncontrollable for an attacker. We may also replace DFF2 with a camouflaged FF to limit observability over CAMFF-Q. However, such an approach can be impractical for a commercial design because the design may not work for the desired clock frequency anymore due to the delay added by the camouflaged FF's.

For a large sequential design, the CAMFF can be located at any sequential depth. Node A, B, C, and, D can be the outputs of different flops if the CAMFF is at a higher sequential depth and directly not accessible through the PI's. Controlling these nodes using the PI's can be impossible for a large design in presence of other camouflaged FF's. In such a scenario, an attacker will have to use the scan access to set desired values at these nodes and observe desired node logic values to deobfuscate the CAMFF. With full access to the scan-chain, an attacker will be able to deobfuscate all the CAMFF's in the design one-by-one ($Example - 1$). An obvious countermeasure would be limiting the access to the scan chain for an attacker using key based scan locking mechanisms [13]. However, such techniques require additional control circuitry and secure non-volatile memory for key storage which can be costly. As an alternative solution, we can use the camouflaged FF's in the scan chain to limit an attacker's controllability and observability over the design.

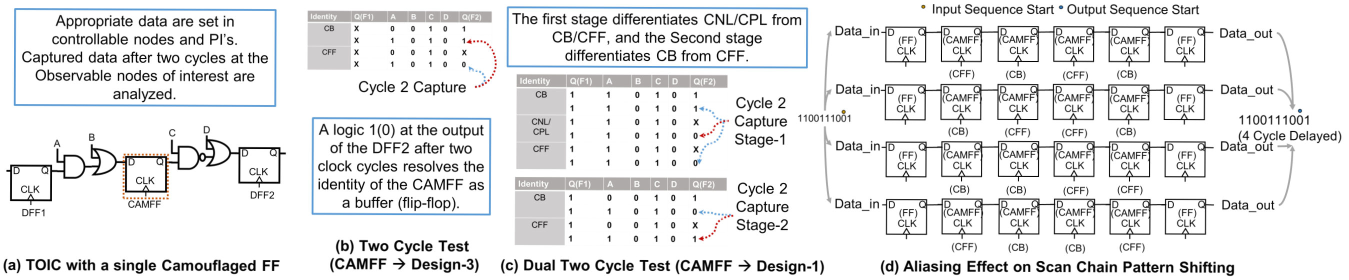
Regardless of the location of the camouflaged FF's in the scan chain, the number of true flip-flops can be figured out by shifting known data patterns through the scan chain as illustrated in the following example.

Example-2: In Figure 5(d), a scan chain is shown with 2 regular DFF's, 2 CFF's, and 2 CB's with four different placements. For all four cases, shifting a known data pattern (e.g. 101100) will produce exactly similar scan digest at the scan output. An attacker will see 4 cycle delay at the output for all possible input data patterns. As the scan chain already has two regular flops, the attacker will know that there are two flops among the four camouflaged FF's. The rest of the camouflaged FF's are dummy buffers.

If all the camouflaged cells are placed at one edge of the scan chain, the number of clock cycles required to shift any desired data pattern from scan input to any regular flop of the scan chain (and from the output of the regular flop to the scan output) can be easily resolved ($Example - 2$). In other words, an attacker would have the necessary controllability and observability over the regular flops of the design to apply the attack demonstrated in $Example - 1$. An intuitive approach of obfuscating the scan chain would be placing them at both edges of the scan chain as shown in Figure 6(a) which we call the edge obfuscated scan chain (*EOSC*). The security of *EOSC* against the reset and shift attack [14] and pattern shifting is illustrated in the following example.

Example-3: In Figure 6(a), 2 camouflaged flops are placed at the left edge, 2 at the right edge, and 2 regular flops at the middle of the scan chain. A global reset will generate a logic 1 at the scan output. However, it does not reveal the identity of CFF2. If we reorder CFF2 and CB2, the scan output logic will still be 1 after a global reset (buffer driven by an inverting flip-flop). A single cycle shift will generate a logic 1 at the output. If we reorder CFF2 and CB2, we will still get a logic 1 after 1 cycle shift at the scan output. Due to similar scan output for different scan chain ordering (aliasing), an attacker can not resolve the true identities of the camouflaged FF's in this manner. Applying the attack demonstrated in $Example - 2$, an attacker will know the total number of dummy buffer among the camouflaged flops in the design. However, the number of flops at the left edge and right edge of the scan chain are still unknown. Without this knowledge, he will not have the required controllability and observability over the regular flops in the scan chain to apply the attack discussed in $Example - 1$.

An attacker can gain controllability and observability over the regular flops of the *EOSC* using the primary inputs (only if the



Appropriate data are set in controllable nodes and PI's. Captured data after two cycles at the Observable nodes of interest are analyzed.

(a) TOIC with a single Camouflaged FF

Identity	Q(F1)	A	B	C	D	Q(F2)
CB	X	0	0	1	0	1
	X	1	0	1	0	1
CFF	X	0	0	1	0	X
	X	1	0	1	0	0

A logic 1(0) at the output of the DFF2 after two clock cycles resolves the identity of the CAMFF as a buffer (flip-flop).

(b) Two Cycle Test (CAMFF → Design-3)

The first stage differentiates CNL/CPL from CB/CFF, and the Second stage differentiates CB from CFF.

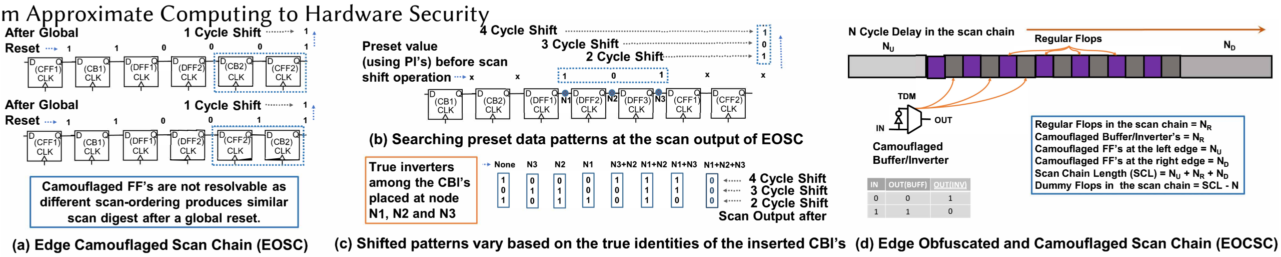
Identity	Q(F1)	A	B	C	D	Q(F2)
CB	1	1	0	1	0	1
	1	1	0	1	0	1
CNL/ CPL	1	1	0	1	0	x
	1	1	0	1	0	0
CFF	1	1	0	1	0	x
	1	1	0	1	0	0

Identity	Q(F1)	A	B	C	D	Q(F2)
CB	1	0	0	1	0	1
	1	1	0	1	0	0
CFF	1	0	0	1	0	x
	1	1	0	1	0	1

(c) Dual Two Cycle Test (CAMFF → Design-1)

(d) Aliasing Effect on Scan Chain Pattern Shifting

Figure 5: (a) A timing obfuscated circuit with a single camouflaged FF; Deobfuscation of the camouflaged FF using (b) two cycle test when $Design - 3$ is used, (c) dual two cycle test when $design - 1$ is used as the CAMFF; (d) Pattern shifting through the scan-chain.


After Global Reset
1 Cycle Shift
1
1
0
0
0
1
(CFF1) CLK
(CB1) CLK
(DFF1) CLK
(DFF2) CLK
(CB2) CLK
(CFF2) CLK
After Global Reset
1 Cycle Shift
1
1
0
0
1
1
(CFF1) CLK
(CB1) CLK
(DFF1) CLK
(DFF2) CLK
(CFF2) CLK
(CB2) CLK
Camouflaged FF’s are not resolvable as different scan-ordering produces similar scan digest after a global reset.
4 Cycle Shift
3 Cycle Shift
2 Cycle Shift
Preset value (using PI’s) before scan shift operation
x
x
1
0
1
x
x
(CB1) CLK
(CB2) CLK
(DFF1) CLK
N1
(DFF2) CLK
N2
(DFF3) CLK
N3
(CFF1) CLK
(CFF2) CLK
True inverters among the CBI’s placed at node N1, N2 and N3
None
N3
N2
N1
N3+N2
N1+N2
N1+N3
N1+N2+N3
4 Cycle Shift
3 Cycle Shift
2 Cycle Shift
Scan Output after
N Cycle Delay in the scan chain
N_U
Regular Flops
N_D
TDM
IN
OUT
Camouflaged Buffer/Inverter
Regular Flops in the scan chain = N_R
Camouflaged Buffer/Inverter’s = N_R
Camouflaged FF’s at the left edge = N_U
Camouflaged FF’s at the right edge = N_D
Scan Chain Length (SCL) = N_U + N_R + N_D
Dummy Flops in the scan chain = SCL - N

None	N3	N2	N1	N3+N2	N1+N2	N1+N3	N1+N2+N3	Scan Output after
1	0	0	0	1	1	1	0	4 Cycle Shift
0	1	1	0	0	1	1	0	3 Cycle Shift
1	0	1	1	0	1	0	0	2 Cycle Shift

IN	OUT(BUFF)	OUT(INV)
0	0	1
1	1	0

(a) Edge Camouflaged Scan Chain (EOSC)

(b) Searching preset data patterns at the scan output of EOSC

(c) Shifted patterns vary based on the true identities of the inserted CBI’s

(d) Edge Obfuscated and Camouflaged Scan Chain (EOCSC)

Figure 6: (a) Camouflaged flops at both edges of the scan chain; (b) Gaining controllability over the edge obfuscated scan chain (*EOSC*) using PI’s; (c) Effect of camouflaged buffer/inverter insertion in the *EOSC*; (d) Edge obfuscated and camouflaged scan chain (*EOCSC*).

regular flops in the original design can be controlled through the primary inputs) which is demonstrated in the following example.

Example-4: The *EOSC* shown in Figure 6(b) has two camouflaged flops at the left edge, two at the right edge, and 3 regular flops at the middle. We assume PI’s can be used to control the regular flops at the middle of the *EOSC*. An attacker runs the chip in functional modes in the cycle with carefully chosen values at the PI’s. At the end of the functional period, the DFF1, DFF2, and DFF3, hold logic 1, 0, and 1 respectively. Then he turns on the scan mode, shifts the data out of the scan chain, and looks for the data patterns captured at the controllable flops at the end of the functional period. He gets a logic 1 at the output after two cycle shift, logic 0 after three cycle shift and logic 1 after four cycle shift. Seeing the preset data pattern at the output, he determines that there are two flops at the right edge of the scan chain. With this knowledge and the knowledge gained from the attack demonstrated in *Example* − 2, he also understands that there is no flop at the left edge of the scan chain. Therefore, he regains controllability and observability over the regular flops in the *EOSC* to apply the attack discussed in *Example* − 1.

We can address the vulnerability demonstrated in *Example* − 4 by inserting camouflaged logic gates in the scan chain which we call *scan chain camouflaging*. We propose to insert camouflaged buffer/inverter at the output of every regular flops in the scan chain. *By putting an inverter between the* $D0$ *and* $D1$ *of* TDM*, we can convert it to a camouflaged cell that can act as a buffer or inverter based on the* V_t *of the NMOS pass transistors (Figure 6(d)).* Note that, the inserted camouflaged buffer/inverter can be power gated during functional operation of the chip and only powered on during scan operation (as they are part of the scan path exclusively) and hence, they will add no power overhead to the design. However, there will be an area overhead. The presence of these camouflaged cells in the scan chain will provide security against the attack stated in *Example* − 4. We illustrate the solution with an example below.

Example-5: In a similar setup of the *Example* − 4 with camouflaged buffer/inverter at the end of each regular flops in the scan-chain, the data pattern that the attacker has to look for at the scan output can be one of possible 8 based on the true identities of the camouflaged buffer/inverters as shown in Figure 6(c).

We term this scan chain with obfuscated flops at the edges and camouflaged buffer/inverters after every regular flop as edge obfuscated and camouflaged scan chain (*EOCSC*). *EOCSC* can effectively limit an attacker’s controllability and observability over the design which ensures the security of *TOIC*. *EOCSC* can also protect standard gate camouflaged designs from the attacks shown in Figure 2. To gain controllability and observability over any flop in an *EOCSC* (Figure 6(d)), an attacker needs to find out (i) the number of true flip-flops at the left and right edges of the scan chain, (ii) the number of inverters between the scan input and the target flop, and the target flop and scan output. If we distribute the camouflaged flops equally on both edges of the scan chain, then $N_U = N_D$ which we call the obfuscated edge length (*OEL*). The number of true flip-flop at both edges of the scan chain can vary between 0 to *OEL*. The number of possible scan chain configurations can be ($2^{N_R} * 2^{N_U + N_D}$) for a brute force attack.

To fully de-obfuscate the *EOCSC*, an attacker has to identify the dummy sequential cells which act as delay elements. The de-obfuscation problem will be similar to diagnosing multiple hold-time failures in the scan-chain which itself has been proven a complicated problem to solve in the past [15]. *The state-of-the-art procedures can only come up with an upper bound and lower bound for the faulty FF in the scan chain in the presence of a single scan-chain failure [16]. The difficulty of the problem lies in the fact that multiple scan-chain failures will produce similar digest (scan-output), and thus they will mask each other.*

Considerations for Design-1 and Design-2: As the camouflaged elements can exhibit more functionalities for *Design* − 1 and *Design* − 2, the brute force attack complexity will be significantly higher. The total number of possible circuit combinations for the obfuscated scan chain will be ($2^{N_R} * 4^{N_U + N_D}$) for both *Design* − 1 and *Design* − 2. The aliasing effect will also be pronounced more in the scan chain. However, the design overhead in terms of additional CLK→Q delay, area, and power overhead will be higher than *Design* − 3.

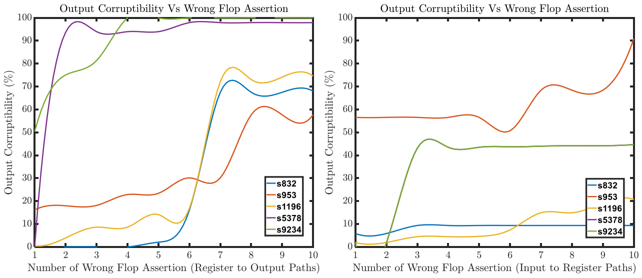
SAT Attack on TOIC: Without controllability over any internal node of the sequential design, an attacker will have to convert the whole circuit into a combinational circuit using time-frame-expansion for a SAT attack [7]. Such an attack can be applied to the circuits without any feedback paths between the sequential elements. The number of time-frame-expansions required will depend on the sequential depth of the circuit. Li et al. showed that the necessary number of time-frame-expansion can have an exponential relation with the decamouflaging time [7]. However, practical circuits will have feedback paths which will create combinational loops during complete circuit unrolling. Existing SAT attacks are ineffective on circuits with combinational loops.

Static Timing Analysis Attack: An attacker can partially de-obfuscate TOIC through a static timing analysis (STA) based attack. In this attack scenario, an STA of the RE generated obfuscated netlist has to be performed with a logic library that has similar timing profile for all the logic gates that are used in the original chip fabrication. If the summation of any two timing path to and from a camouflaged FF is greater than the chip operating clock period, its identity can be resolved as a flip-flop. However, a designer can intentionally create false paths to and from a camouflaged FF to thwart such attack. If the summation of the most delayed timing paths to and from a camouflaged FF is less than a clock period, the camouflaged FF can either be a dummy flop or the timing paths are indeed very short paths.

5 SIMULATION RESULTS

To estimate the design overhead of TOIC, we have synthesized ITC’99 sequential benchmarks [17] using ibm32nm gate library at 500 MHz clock frequency. The area and power of the *CAMFF’s* have been interpolated based on their respective ratio that we have found from simulation with ASU 32nm planar MOSFET models [12] (based on *Design* − 1). We have chosen *Design* − 1 as it has the highest overhead among the three designs and, hence gives us a pessimistic estimation of the overall *TOIC* design overhead. The scan chain length (*SCL*) has been calculated based on the assumption of equal obfuscated edge length at the beginning and the end of the scan chain. We have presented the data based on 5% and 10% $CAMFF's$ among the total possible sequential elements in the design with 50% *CFF* and 50% $CNL/CPL/CB$ among the $CFF's$. We have excluded the camouflaged BUFF/INV power data from the chip power calculation considering them to be power gated during normal operation.

For reasonably large circuits, e.g. b19, TOIC will add a power overhead of only 1.6% for 5% *CAMFF* insertion, and nearly 3.27% for 10% *CAMFF* insertion which is significantly low. The associated area overheads are 15.40% and 15.84% respectively. The camouflaged buffer/inverter between the regular FFs accounts for the majority of the area overhead. The area overhead remains constant with the $CAMFF's$ which is evident from the results. For 10% *CAMFF* insertion, the possible number of circuit configuration is 4^{304} and scan chain configurations can be as large as $(4^{304} * 2^{5899})$ which can not be solved using brute force in any reasonable time-frame.



(a) Near REG2OUT Paths

(a) Near IN2REG Paths

Figure 7: Output corruptibility of ISCAS'89 benchmarks for 12000 random inputs based on wrong flop assertion near (a) reg2out and (b) in2reg timing paths.

We have simulated the ISCAS'89 benchmarks [18] for 12000 random inputs with randomly inserted $CAMFF's$ in the input-to-register and the output-to-register paths respectively and compared them with the original netlist to measure output corruptibility of TOIC. Simulation shows that even a single wrong flop assertion during de-obfuscation of the circuit can produce significantly corrupted output. For instance, for s832, even two wrong flop assertions can attain 100% output corruptibility. Moreover, the insertion of *CAMFF* near output-to-register paths can result in greater output corruptibility as they are closer to the PO's (Figure 7(a)&(b)).

6 DISCUSSION

Testability of TOIC: The controllability and observability is greatly reduced for an attacker as long as the true identities of the camouflaged FF's and the camoufaged $BUFF's/INV's$ in the chain are hidden. From the test perspective *all the* $CFF's$ *and* $DFF's$ *in the obfuscated scan chain are controllable and observable to ensure similar testability as the original design*

Manufacturability: State-of-the-art IC fabrication use multi-thres-hold designs to assist in timing closure and power optimization [10]. Hence, TOIC can be fabricated without altering the existing manufacturing flow.

Clock Tree Routing and Power Distribution Network: The additional camouflaged sequentials will be connected to the clock network which will add additional routing overhead. A DC signal *VSN* has to be routed to the camouflaged cells which can use some tracks of the power distribution network.

7 CONCLUSION

We presented TOIC, a novel circuit obfuscation technique using camouflaged sequential elements. We also demonstrated a scan chain obfuscation methodology using these sequential cells and additional camouflaged buffer/inverters. By restricting the scan access as well as ensuring significant output corruptibility for wrong assertions during de-obfuscation, TOIC can provide substantially greater security than contemporary gate camouflaging techniques.

ACKNOWLEDGMENT

This work is supported by SRC (2847.001), NSF (CNS- 1722557, CCF-1718474, DGE-1723687 and DGE-1821766) and DARPA Young Faculty Award (D15AP00089).

REFERENCES

[1] R. Torrance *et al.*, "The state-of-the-art in ic reverse engineering," in *Cryptographic Hardware and Embedded Systems-CHES 2009.*

[2] M. Holler *et al.*, "High-resolution non-destructive three-dimensional imaging of integrated circuits," *Nature*, vol. 543, no. 7645, p. 402, 2017.

[3] J. Rajendran *et al.*, "Security analysis of integrated circuit camouflaging," in *Proceedings of the 2013 ACM SIGSAC.* ACM, 2013, pp. 709–720.

[4] M. El Massad *et al.*, "Integrated circuit (ic) decamouflaging: Reverse engineering camouflaged ics within minutes." in *NDSS*, 2015.

[5] C. Yu *et al.*, "Incremental sat-based reverse engineering of camouflaged logic circuits," *IEEE TCAD*, vol. 36, no. 10, pp. 1647–1659, 2017.

[6] G. L. Zhang *et al.*, "Timingcamouflage: Improving circuit security against counterfeiting by unconventional timing," in *DATE 2018.*

[7] M. Li *et al.*, "A synergistic framework for hardware ip privacy and integrity protection," Ph.D. dissertation, 2018.

[8] R. Karmakar *et al.*, "Encrypt flip-flop: A novel logic encryption technique for sequential circuits," *arXiv preprint arXiv:1801.04961*, 2018.

[9] J.-W. Jang *et al.*, "A novel threshold voltage defined multiplexer for interconnect camouflaging," Penn State, Tech. Rep., 2017.

[10] S.-L. Wu *et al.*, "Method to fabricate dual threshold cmos circuits," Aug. 1 2000, uS Patent 6,096,611.

[11] A. S. Iyengar *et al.*, "Threshold defined camouflaged gates in 65nm technology for reverse engineering protection," in *ISLPED 2018.*

[12] W. Zhao *et al.*, "Predictive technology model for nano-cmos design exploration," *ACM JETC*, vol. 3, no. 1, p. 1, 2007.

[13] J. Lee *et al.*, "Securing scan design using lock and key technique," in *DFT 2005.* IEEE, 2005, pp. 51–62.

[14] D. Hely *et al.*, "Scan design and secure chip." in *IOLTS*, vol. 4, 2004, pp. 219–224.

[15] Y. Huang *et al.*, "Survey of scan chain diagnosis," *IEEE Design & Test of Computers*, vol. 25, no. 3, 2008.

[16] Y.-L. Kao *et al.*, "Jump simulation: A technique for fast and precise scan chain fault diagnosis," in *ITC-2006.* IEEE, 2006, pp. 1–9.

[17] F. Corno *et al.*, "Rt-level itc'99 benchmarks and first atpg results," *IEEE Design & Test of computers*, vol. 17, no. 3, pp. 44–53, 2000.

[18] F. Brglez *et al.*, "Combinational profiles of sequential benchmark circuits," in *IEEE International Symposium on Circuits and Systems 1989.*

Table 1: Design Overhead for Timing Obfuscation

Bench-mark	Std Cells	Flip-flops	Area (um^2)	Power (uw)	5% *CAMFF* with 50% Dummy Flops					10% *CAMFF* with 50% Dummy Flops				
					Area	Power	SCL	OEL	CBI	Area	Power	SCL	OEL	CBI
b13	147	51	676	22.4	950.07	24.92	53	2	48	974.07	27.48	55	4	46
b15	2846	417	10884	309	13036	316.5	423	6	410	13109	323.9	429	12	404
b17	8707	1317	33431	931	40216	951.8	1334	17	1299	40422	972.6	1351	34	1282
b18	32545	3020	110989	3085	126540	3131	3058	38	2981	127000	3177	3096	76	2943
b19	58911	6042	201993	5595	233120	5686	6118	76	5965	234040	5778	6194	152	5889

* IBM 32nm technology library, SCL - Scan Chain Length, OEL - Obfuscated Edge Length , CBI - Camouflaged Buffer/Inverter.