IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 15, 2019, accepted August 16, 2019, date of publication August 20, 2019, date of current version September 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936401

Robust Logic locking for Securing Reusable DSP Cores

MAHENDRA RATHOR, (Member IEEE), AND ANIRBAN SENGUPTA, (Senior Member, IEEE)
IIT Indore, Indore 453552, India

Corresponding author: Mahendra Rathor (phd1801201004@iiti.ac.in)

This work was supported by the Council of Scientific and Industrial Research (CSIR) under Grant 22/730/17/EMR-II. This article was an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

ABSTRACT A System on Chip (SoC) used in Consumer Electronics (CE) systems integrates a number of reusable Intellectual Property (IP) cores from digital signal processing (DSP), multimedia etc. However, these DSP based IP cores are susceptible to various hardware threats such as piracy, Trojan insertion, overbuilding and reverse engineering. Thus, security of DSP cores is very crucial. An IP core can be secured against aforementioned hardware threats by employing logic locking based security mechanisms. This paper presents a novel robust logic locking using hybrid locking cells for securing DSP cores. The proposed logic locking is based on a novel advanced encryption standard (AES) based reconfigurable hybrid locking cell architecture that ensures strong security against key sensitization, removal and SAT attacks. The strength of the proposed approach has been assessed in terms of probability of obtaining correct key of a locked design in exhaustive trials. Results of proposed work on DSP cores yielded higher logic locking strength and lower design overhead compared to recent prior works.

INDEX TERMS DSP, functional obuscation, intellectual property core, flip-flop.

I. INTRODUCTION

DSP kernels are the prime architect of modern consumer electronics devices such as personal computer, wireless router, modem, digital camera etc. In these devices, DSP kernels facilitate applications such as digital media and communication, image processing and compression etc. However, the rising usage of DSP kernels in the modern CE devices raises grave security concen against various hardware theats suchas IP piracy, overbuilding, Trojan insertion and reverse engineering (RE) attacks etc [1], [2]. The reason for the security concern is the globalization of integrated circuit (IC) design cycle (different stages of an IC deisgn flow are accomplished at different design houses in the supply chain) to attain cost savings in IC manufacturing. However, in the process of cost savings, an IP core design becomes susceptible towards aforementionaed threats in the design houses [3]–[7]. This is because an adversary in the untrusted design house can (i)reverse engineer the design to use it illegally (ii) realize his malicious intents such as inserting Trojan at safe places in the design to obtain the desired functionality (iii) make unauthorized use of an IP core through piracy (iv) overbuid the IPs/ICs to generate illegal profit.

The associate editor coordinating the review of this article and approving it for publication was Bora Onat.

However, IP cores can be secured against these attacks by locking the functionality of the design by inserting some key gates which are actuated through valid keys. This kind of mechanism of obscuring the functionality of a design is known as logic locking [4]–[9]. Proposed approach employs logic locking for security of resusable IP cores for DSP applications by inserting DSP locking cells (DLCs) at appropriate locations in the design. The novelties of proposed approach are described as follows:

(a) A novel structure of DLC is proposed in this paper which exploits filp-flops (FF) alongwith logic gates.
(b) This is the first work in the literature which renders the probability of obtaining the correct key even in exhaustive trials much lesser than 1.
(c) This paper presents the strength of the proposed logic locking in terms of the proposed security metric and ensures that proposed logic locking is significantly more robust than recent similar work.
(d) Proposed work achieves high security at lower design overhead than recent similar work.

II. RELATED WORK

This section discusses some selected logic locking approaches as follows. Logic locking is employed for complex DSP cores by inserting IP core locking blocks (ILBs) in

the design [8], [12]. The ILBs comprises of a hybrid combination of various AND, NAND, NOT, XOR, XNOR gates and actuates on providing a valid 8-bit key. Approaches [8], [12] are the only works available in the literature which introduce logic locking of DSP cores. However, these approaches are also incapable of securing the key of the design from being obtained by an attacker in exhaustive trials. Additionally, these approaches also incur substantial design overhead. On the contrary, proposed approach is capable of (i) hindering the possibility of obtaining correct key in exhaustive trials (ii) reducing the design overhead. These approaches ([8], [12] and proposed) are more aptly suitable for DSP circuits (less suitable for combinational circuits) because of the following reasons:

(i) the proposed approach uses the concept of encoded variable μ during logic locking that is used **for deciding the locations of the inserted DLCs based on** the designer's choice. This encoding process is performed on the high level synthesis (HLS) generated datapath. DSP circuits being complex in terms of gate structure, uses HLS framework to generate datapath architecture. Hence, the proposed approach rightly fits the target hardware. Combinational circuits being less complex in terms of gate complexity do not use HLS for datapath generation. Thus, the proposed approach cannot be applied on combinational circuits directly; (ii) the proposed approach uses DLC which is an amalgamation of several gate strcutures and flip-flops. DSP circuits are usually very complex in terms of gate count (in the order ranging from 10k-50k gates and FFs). Thus insertion of the proposed DLCs in the complex gate structure of DSP circuits does not add any significant design overhead. In fact the overhead due to DLCs is marginal (usually $\sim$5%). on the contrary, combinational/sequential circuits are significantly less complex in terms of gate count (in the order of few thousand gates maximum). Thus insertion of DLCs in the combinational/sequential circuits would result into design overhead. Therefore the proposed approach is more suitable for DSP hardware.

Besides, works such as [4], [10], [11], [20] employ logic locking on combinational circuits using key gates and multiplexers. However, these approaches do not leverage HLS framework for employing security feature during logic locking. Further, these are also incapable of securing the key of the design from being obtained by an attacker in exhaustive trials. However, the proposed approach renders the obtaining of correct key in exhaustive trials almost infeasible. In other words, the proposed approach renders the probability of obtaining correct key in exhaustive trials very lesser than 1 (threatening the fact that the correct key can be obtained in exhaustive trials (using brute force) with probability 1).

Additionally, performing logic locking for DSP designs is different from common circuits (such as combinational). This is because the common circuits are readily available in the form of gate-level netlist/Verilog/VHDL description etc. Therefore, logic locking on common circuits [4], [10], [11] can directly be applied on available descriptions. On the contrary, DSP cores are not readily available in the form of aforementioned design descriptions (instead available in the form of high level description such as C/C++ code or intermediate description such as data flow graph). Therefore, HLS framework needs to be integrated to employ logic locking technique for DSP designs. Proposed work inserts the DSP locking cells in the RTL description which is obtained using HLS process. This is because inserting locking cells into the gate-level netlist is not feasible. The reason is the unavailability of DSP designs in standard gate-level netlist owing to their higher complexity (requiring several thousands of gates). This makes the logic locking of DSP designs different from the common (combinational) circuits wherein locking is performed at gate-level.

Further, some other works [21]–[25] have also employed obfuscation of DSP cores. However, these works exploited structural obfuscation based technique, but did not employ logic locking. Thus the proposed approach is completely different from the structural obfuscation.

III. PROPOSED WORK

A. PROBLEM FORMULATION

Given a control data flow graph (CDFG) representing DSP application, resource configurations, module library, generate a low-overhead, highly secured (using logic locking) DSP IP cores.

B. THREAT MODEL

Trojan insertion, IP piracy, overbuilding and RE attackes are serious hardware threats against the security of DSP cores. Proposed work targets aforementioned threats and secures DSP cores by employing robust logic locking based security. Owing to locking of the IP cores, an attacker fails to launch aforementioned attacks as he/she is incapable to crack the proposed robust logic locking.

C. PROPOSED LOGIC LOCKING METHODOLOGY

The proposed logic locking methodology for DSP IP core is shown in Fig. 1. Inputs for the proposed methodology are: DFG/CDFG representing DSP application, resource configuration, module library, designer selected tuning variable (μ) and keys for the DLCs. These inputs are fed into the proposed logic locking process which finally generates the locked gate-level netlist at the output. The proposed logic locking process is accomplished in three steps as shown in Fig. 1:

(1) Generate a RTL datapath of DSP application through HLS; The HLS framework uses CDFG/C-code/transfer function representing DSP core, designer selected resource configuration and module library as inputs and generates an RTL datapath through three different phases of HLS; (a) scheduling (b) allocation (c) binding [13].

(2) Generate locked RTL datapath by inserting reconfigured DLCs in the RTL datapath based on encoded 'μ' and AES-128 output; designer selected tuning variable 'μ' decides the location for DLCs insertion based on its encoding rules and AES output is used as keys for

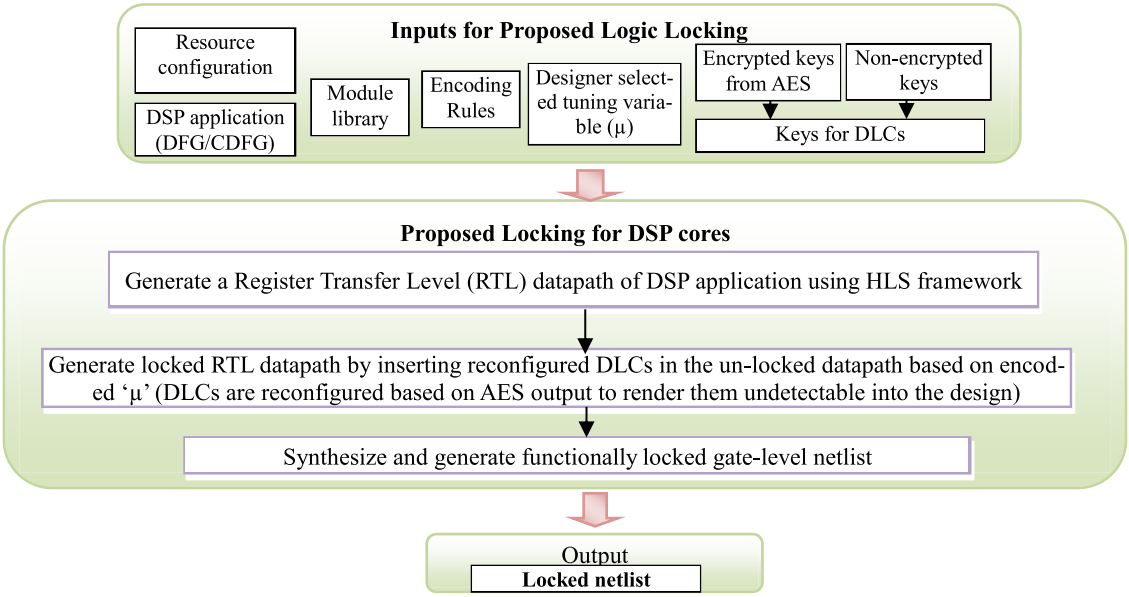

FIGURE 1. Overview of proposed logic locking methodology for DSP IP cores.

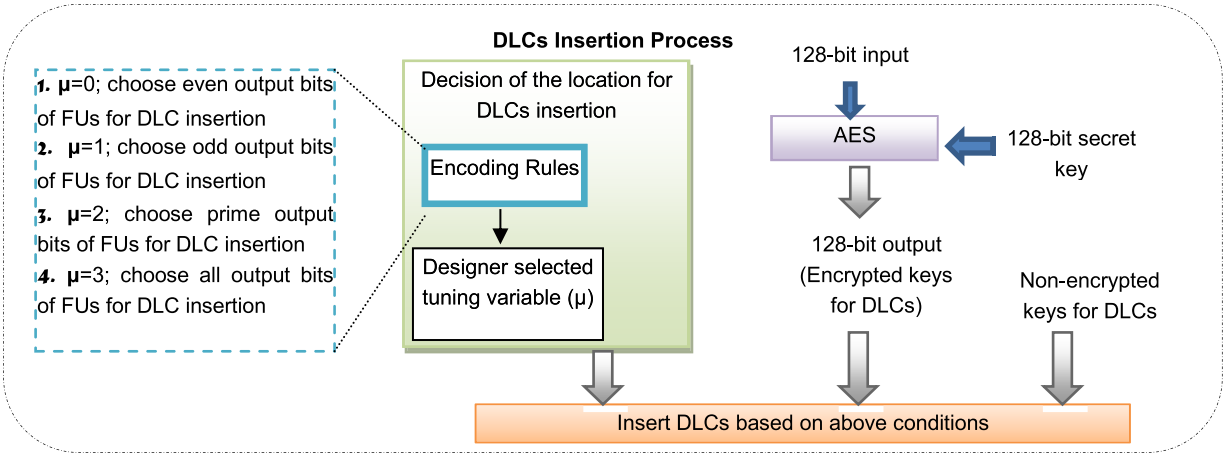

FIGURE 2. The process of DLCs insertion into RTL datapath of DSP cores.

DLCs to generate reconfigured DLC structures. Insertion process of proposed DLCs is shown in Fig. 2 and it is elaborated under the sub-section 3.3.1.

(3) Synthesize and generate locked gate-level netlist.

In order to lock the fuctinallity of the DSP design, a novel reconfigurable structure of DLC is proposed in this paper. This reconfiguration is performed for proposed DLC structures using AES. The proposed reconfigurable DLC is capable of achieving higher security at lower overhead than prior works. In the context of security, probability of finding correct key in exhaustive trials (trying all key combinations) through proposed approach is highly lesser than the probability of finding the same in exhaustive trials through prior works. More explicitly, attacker can obtain correct key through proposed approach only if he/she applies it at first trial. While applying different key combinations, if attacker misses correct key combination at first trial then he/she becomes unable to find correct key in remaining exhaustive trials. On the contrary, in the similar recent approaches, probabaility of finding correct key in exhaustive trials is 1. Additionally, despite of encoding lesser key bits (possibly) than prior arts, the proposed approach enhances security by drastically reducing the probability of finding correct key in exhaustive trials. Further, due to proposed structure of DLCs and achievement of higher security at lesser key bits, proposed approach incurs lower overhead than similar prior work on DSP cores.

1) INSERTION TECHNIQUE OF PROPOSED DLCS

Insertion technique of proposed DLCs is shown in Fig. 2. As shown in the figure, locations for DLCs insertion and structure of DLCs need to be determined prior the DLCs

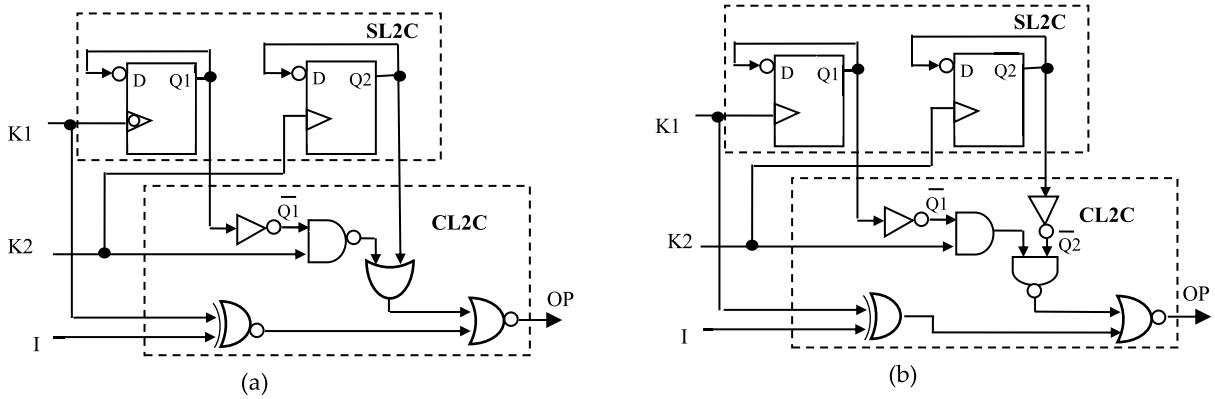

FIGURE 3. Proposed sample reconfigured DLC structures (where, "K1K2" is a two-bit key).

insertion process. The locations of the DLC insertion is decided by the designer selected tuning variable 'μ' which ranges from 0 to 3 and abides by the following encoding rules:

1) $\mu = 0$; choose even output bits of Functional Units (FUs) for DLC insertion
2) $\mu = 1$; choose odd output bits of FUs for DLC insertion
3) $\mu = 2$; choose prime output bits of FUs for DLC insertion
4) $\mu = 3$; choose all output bits of FUs for DLC insertion

Further, total number of DLCs (T_{DLC}) to be inserted in a design is calculated as follows:

$$T_{DLC} = (\text{total number of FUs}) * (\#\text{ of output bits of FUs chosen according to } \mu) \quad (1)$$

For example, if there are three FUs in a design and size of each FU is 32-bit then total output bits of FUs in the design is 3*32 = 96. Thus, for $\mu = 3$, total 96 DLCs are inserted in the design. Once the location of DLCs insertion and number of DLCS are decided based on μ, all the proposed DLCs are inserted at decided sites in the un-locked RTL datapath of a DSP design. Further, the proposed DLC structure is not fixed rather it is reconfigurable according to its key value. As shown in Fig. 2, the keys for a subset of DLCs are obtained from AES output. Hence, the DLCs structures are reconfigured based on keys obtained from AES. Since a proposed DLC requires a two-bit key to be activated, therefore up to 64 DLCs can be reconfigured using 128-bit AES output. Owing to this structural rconfiguration of DLCs, an attacker fails to detect them in the design as he/she does not know the reconfigured DLC structure.

2) OVERVIEW OF PROPOSED IP CORE LOCKING BLOCKS

The proposed DSP locking cell (DLC) comprises of two logic locking sub-cells viz. sequential logic locking cell (SL2C) and combinational logic locking cell (CL2C) as shown in Fig. 3. Each SL2C consists of two D flip-flop (FF). These FFs are refered as key-FFs throughout this paper because their functioning is associated with the key bits. Each key-FF is initialized with '0' (which can be achieved upon power on reset) and each gets triggered on change in key values. Further, each CL2C consists of combination of various logic gates such as EX-OR, EX-NOR, OR, NOR, AND, NAND and NOT gate. Various combinations of logic gates under CL2C and variations of edge triggereing (+ve and −ve edge) in the key-FFs under SL2C can generate different reconfigured functional DLC structures. Further, there are several features of proposed DLCs which render them a better choice over prior works for logic locking of DSP IP cores.

Features:

1) *Pair-wise security*: This security feature of proposed DLCs impedes the key sensitization attack. In this attack, attacker tries to sensitize the keys embedded within activated IC by applying required input pattern [4]. The attacker attempts to find this required input pattern from the locked netlist. However, this input pattern required to sensitize a key can be found only if any other key does not interfere in the path of the sensitization. Since, in the proposed DLC, one key interferes in the path of sensitization of the other, therefore a key cannot be sensitized without knowing/controlling the other interfering key. Further, controlling of interfering keys is not feasible because keys are not accessible to the attacker [4]. Thus, the keys of each propsed DLC are pair-wise secured and hence, key sensitization is not passible.
2) *Prohibiting key-gate isolation*: Key-gates (key-inputs) are considered to be isolated if they are not connected one-another through any path. Keys of such gates can easily be sensitized at primary output [4]. Proposed DLC structure does not contain isolated key-gates and ensure that two key-inputs are dependent on each other.
3) *Ensuring protection against run of key-gates*: Key-gates connected in a sequence form a run resulting into increase in valid key space. Thus, attacker's effort to find the key is reduced [4]. In the proposed DLC, there is no run (sequence) of key-gates that can be replaced by

a single key-gate. Thus, proposed DLCs ensure protection against run of key-gates.

4) *Non-mutable key gates*: If muting a key-gate leads to sensitize another key, then the key gate is said be mutable key-gate [4]. In the proposed DLC, key-gates are non-mutable because a key bit can not be determined by muting any key-gate.
5) *Robust structure*: Mixing of the outputs of Key-FFs and key-gates renders the structure of proposed DLC more robust. Apart from the features discussed above, this robust structure of the proposed DLC also leads to several other advantages such as:
   (a) Obtaining correct key in exhaustive trials by an attacker becomes infeasible unless correct key is only applied at first trial.
   (b) Alongwith combinational gates, D-FFs also contribute in the proposed logic locking. Thus, identification of mere key-gates or only key-FFs is not sufficient for an attacker to lauch attacks. The attacker needs to identify all key-gates and key-FFs. Thus, attcker's effort is highly increased as key-FFs and key-gates are camouflaged in the countless similar resources present in DSP core datapath and controller gate level netlist.
   (c) Due to lesser key-bits, more mumber of DLC keys can be encrypted using AES. Additionally, lesser key bits incur less design overhead.

3) SECURITY OF PROPOSED DLCS

Security Metric: In this paper, security of logic locking (strength) is assessed in terms of the probability of obtaining correct key in exhaustive trials (P_{ck}) (trying all key combinations). This P_{ck} is the security metric for evaluating security of proposed logic locking and comparing with prior works. The equation of P_{ck} is constructed as follows:

$$P_{ck} = \left(P_{appl}^{ck_{(1)}}\right)\left(P_{obt}^{co_{(1)}}\right) + \left(P_{appl}^{ck_{(2)}}\right)\left(P_{obt}^{co_{(2)}}\right) + \cdots + \left(P_{appl}^{ck_{(2^{k_t})}}\right)\left(P_{obt}^{co_{(2^{k_t})}}\right) \quad (2)$$

where, k_t is total number of key-bits and 2^{k_t} is the exhaustive key-combinations/trials.

$\left(P_{appl}^{ck_{(1)}}\right)$ = probability of applying correct key at 1^{st} = probability of applying correct key at 1^{st} trial.

$\left(P_{appl}^{ck_{(2)}}\right)$ = probability of applying correct key at 2^{nd} trial.

$\left(P_{appl}^{ck_{(2^{k_t})}}\right)$ = probability of applying correct key at $(2^{k_t})^{th}$ trial.

$\left(P_{obt}^{co_{(1)}}\right)$ = probability of obtaining correct output on applying correct key at 1^{st} trial.

$(P_{obt}^{co_{(2)}})$ = probability of obtaining correct output on applying correct key at 2^{nd} trial.

$\left(P_{obt}^{co_{(2^{k_t})}}\right)$ = probability of obtaining correct output on applying correct key at trial.

The probability of obtaining correct output on applying correct key at any trial seems to be likely because it should be always 1 (as in case of prior works). However, it shows great significance in case of proposed work. This is because in case of proposed work, attacker is not able to obtain correct output even on applying correct key unless it is applied only in the first trial.

a: STRENGTH OF LOGIC LOCKING OF PRIOR WORKS [4], [8] USING (2)

In case of prior works, probability of obtaining correct output on applying correct key is 1. This is because if correct key is applied at any trial by an attacker then correct output will definitely be obtained irrespective of the trial number (i.e. 1^{st} or 2^{nd} or last trial). Further, the probability of applying correct key at any trial is $1/\ 1/2^{K_t}$. This is because the total possible trials are 2^{K_t} and the favourable trial is only one correct key. Hence, the probability of obtaining correct key in exhaustive trials (P_{ck}) using (2) becomes:

$$\begin{aligned} P_{ck} &= \frac{1}{2^{K_t}} \cdot 1 + \frac{1}{2^{K_t}} \cdot 1 + \frac{1}{2^{K_t}} \cdot 1 + \ldots 2^{2^{K_t}}\ times \\ P_{ck} &= 2^{K_t}\frac{1}{2^{K_t}} \\ P_{ck} &= 1 \end{aligned} \quad (3)$$

It is evident from (3) that the probability of obtaining correct key in exhaustive trials is 1.

b: STRENGTH OF LOGIC LOCKING OF PROPOSED WORK USING (2)

The probability of obtaining correct output on applying correct key at any trial is non-trivial in case of proposed work because it is not always 1. This is the crux of proposed approach. The probability of obtaining correct output on applying correct key only in the first trial is 1. However, for any other trials, this probability becomes 0 because of the structural property of the proposed DLC. Hence, the probability of obtaining correct key in exhaustive trials using (2) becomes:

$$\begin{aligned} P_{ck} &= \frac{1}{2^{K_t}} \cdot 1 + \frac{1}{2^{K_t}} \cdot 0 + \cdots + \frac{1}{2^{K_t}} \cdot 0 \\ P_{ck} &= 1/2^{K_t} \end{aligned} \quad (4)$$

More explicitly, if correct key is only applied at the first trial, only then an attacker can obtain the correct key. If attacker misses the correct key at the first trial, then he/she becomes unable to obtain correct output even on applying correct key in exhaustive trials. Hence security (strength of logic locking) of proposed approach in terms of the probability of obtaining correct key in exhaustive trials becomes $1/1/2^{K_t}$ instead of 1. Lower is the value of P_{ck} higher is the security of proposed approach.

Explanation for the security of the proposed DLCs: The reason of not obtaining correct output in exhaustive trials unless the correct key is applied only in the first trial is

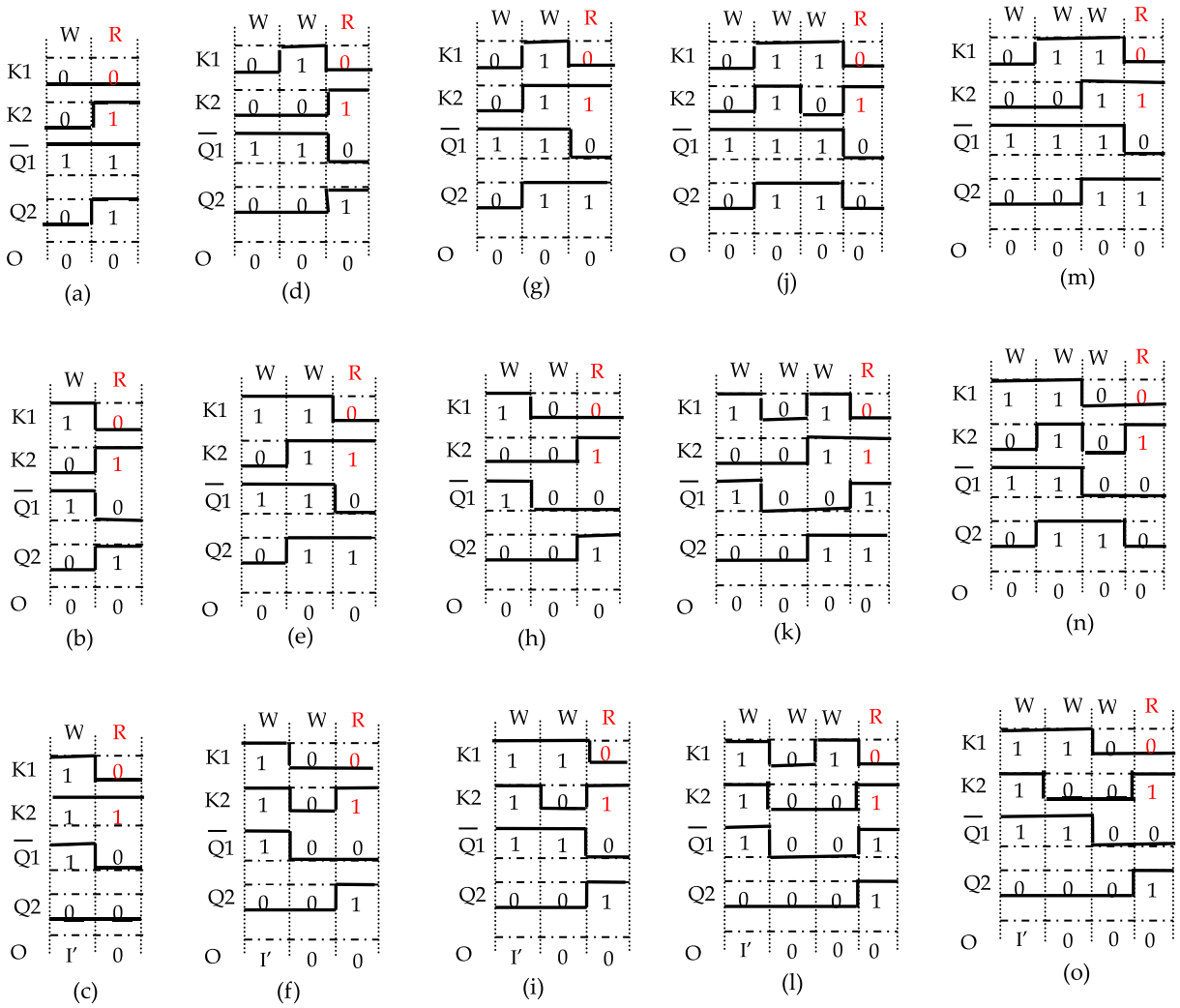

FIGURE 4. Waveforms showing that correct output (O<=I) is not obtained unless the correct key is applied only in the first trial. *Note- R: right key combination, W: wrong key combination. At W, output (O) is either complement of input (I') or 0 and at R, O is always 0.*

provided below with the aid of proposed DLC and timing diagram (shown in Fig. 4):

- The proposed DLC structure exploits FF elements. The unique feature of a FF is that it can function in toggling mode on applying a specific input alongwith a clock. In case of proposed DLCs, arrangement of logic gates and key-FFs are in such a manner that this unique feature of FFs can be leveraged for security of logic locking. To obtain the correct output through proposed DLCs, output of each key-FF must remain at 0 during application of the correct key. Fig. 3 shows that key bits are applied at the clock i/p of the key-FFs. Thus during exhaustive trials, change in key-bits would result into flipping of the FF output. This ensures that the output of a FF does not remain 0 on application of the correct key, if it is applied after a wrong key combination. Thus, attacker becomes unable to obtain correct output in exhaustive trials of key combinations unless the correct key is applied only in the first trial. The aforementioned reason is explained below with the aid of the proposed DLC (Fig. 3(a)) structure in case of different scenarios:

  Note: Each DLC is activated on applying two-bit key therefore exhaustive trials in finding correct key of a DLC are 4.

  (1) **Correct key at first trial**: Considering intial output of each key-FF is 0 (which can be achieved practically upon power-on reset), if correct key "01" is applied on the DLC (Fig. 3(a)) at the first trial, path from input to output bit of an DLC is unlocked and correct output is obtained. Thus, the probability of obtaining correct output on applying correct key at first trial is 1. This was possible because Q1Q2 initially remains at "00".

  (2) **Correct key at second trial**: After applying a wrong key at first trial, assuming correct key "01" is applied on the DLC (Fig. 3(a)) in the second trial. This results into transition happening on either of the key bits (K1 or K2)

as shown in wavefoms in Fig. 4(a), (b) and (c). *Note: 'W' stands for wrong and 'R' stands for right key-combination.*This transition on a key-bit leads to flipping on output of respective key-FF and thus Q1Q2 does not remain at ''00''. For example, if wrong key ''00'' is applied at first trial, then appling correct key ''01'' at second trial leads to transition on key-bit 'K2' (shown in Fig.4(a)) which flips the output Q2. This flipping in the output of key-FF renders output of the DLC incorrect on applying correct key at the second trial. Therefore, the probability of obtaining correct output on applying correct key at second trial is 0.

(3) **Correct key at third trial**: As shown in the waveforms in Fig. 4(d), (e), (f), (g), (h) and (i), if correct key is applied at third trial after two wrong trials, correct ouput is not obtained from the DLC (Fig. 3(a)). This is because applying correct key after two wrong trial leads to transition on key-bits which further flips either Q1 or Q2 or both thus Q1Q2 does not remain at ''00''. Therefore, the probability of obtaining correct output on applying correct key at third trial is 0.

(4) **Correct key at fourth trial**: For the same reason as discussed above, applying correct key after three wrong trials renders the output of the DLC incorrect as shown in waveforms in Fig. 4 (j), (k), (l), (m), (n) and (o). Thus, the probability of obtaining correct output on applying correct key at fourth trial is also 0.

Although, it does not seem intuitive that the probability of obtaining correct output by applying correct key is not always 1, however it has been proved for the proposed DLC structure. Hence correct key cannot be obtained through proposed DLCs in exhaustive trials except first and this is the strength of the proposed logic locking.

D. EXAMPLE AND DEMONSTRATION

Proposed logic locking approach has been demonstrated for a sample application- Finite Impulse Response (FIR) filter design with resource configuration of 1(+), 1(∗). To employ logic locking in this application, proposed DLCs are inserted at the output bits of FUs based on μ. In order to make this logic-locked FIR filter, more robust and secured, custom-AES block is integrated to this locked design. Further, keys for the DLCs are obtained from encrypted output bits of AES, resulting into reconfigured DLC structure. This locked design integrated with custom-AES design has been implemented in QuartusII v13.0 and synthesized. Functionality of the design is verified using modelsim simulator for Cyclone-II: EP2C70F896C6 FPGA. Post-synthesis gate-level structure of locked design is shown in Fig. 5 (a) and (b). Cells highlighted in the blue color in the figures are the D-FFs. As shown in the Fig. 5, logic gates used in proposed DLCs are camouflaged with those of FIR filter and AES design. Similarly, D-FFs used in proposed DLCs are camouflaged with the D-FFs of FIR filter and AES design. Thus post-synhesis, resources of proposed DLCs become completely indistinguishable from those of FIR filter and AES design.

For this sample application, in case of $\mu = 3$, total # DLCs inserted in un-locked design using (1) are 8 (two FUs of size 4-bit) and total # of key bits are 16. Therefore 65536 (2^{16}) are the total possible exhaustive trials. Thus probability of obtaining correct key in first trial during exhaustive attempts using (4) is 0.000015 ($1/2^{16}$) which is much lesser than 1.

Reconfiguration of proposed DLCs: As discussed earlier, encrypted output of AES is leveraged to reconfigure DLC gate structure. For example, reconfiguration of a DLC according to a two-bit key (''10'') obtained from AES is shown in Fig. 6. Similarly, other DLC structures can be reconfigured according to the key bits obtained from 128-bit AES output.

E. HANDLING DIFFERENT ATTACKS SCENARIO

This section discusses the different attack scenarios from an attacker's perspective and presents the proposed logic locking as a more secured countermeasure over prior works against those attacks. An attacker is assumed to have the following prerequisite to perform attacks:

(a) Access to a locked gate-level netlist.
(b) Access to a layout/GDS-II file and advanced tools that are capable of performing reverse engineering to obtain locked gate-level netlist.
(c) Acivated functional IC of the locked design which can be obtained from open market.

Availing the above prerequisite, an attacker can attempt to unlock the design by launching various attacks such as: key-sensitization based attacks [4], SAT attack [14], removal attack [15]. Further, as design becomes un-locked, it becomes vulnerable to IP piracy, overbuilding, Trojan insertion and RE attacks. Handling of various attacks using proposed logic locking is discussed as follows:

1. **Handling key-sensitization based attacks**: In this attack, a key-bit (embedded within functional IC) can be sensitized at primary output by applying a suitable input pattern to primary input. This suitable input pattern is required to be identified by the attacker from the locked netlist without controlling the key inputs. However, in the path of sensitization of a key, if another key bit interferes then key sensitization requires controlling of interfering key bit. In the proposed DLC, one key interferes in the path of sensitization of the other. Thus, this interfering key bit requires to be controlled by the attacker which is not possible because key bits are not accessible to the attacker. Hence key-bits are pair-wise secured. Further complex arrangement of key-gates and key-FFs in proposed DLCs renders the logic locking resilient against following other key-sensitization based attacks (discussed under the subsection 3.3.2) such as: (i) key-sensitization attack based on isolated key-bits (ii) key-sensitization attack based on run of key-gates (iii) key-sensitization attack based on mutable key-gates. Thus, attacker is forced to perform the brute-force attack which is further highly complex in case of proposed DLCs because probability of obtaining correct key in exhaustive trials is not 1 but $1/2^{k_t}$.

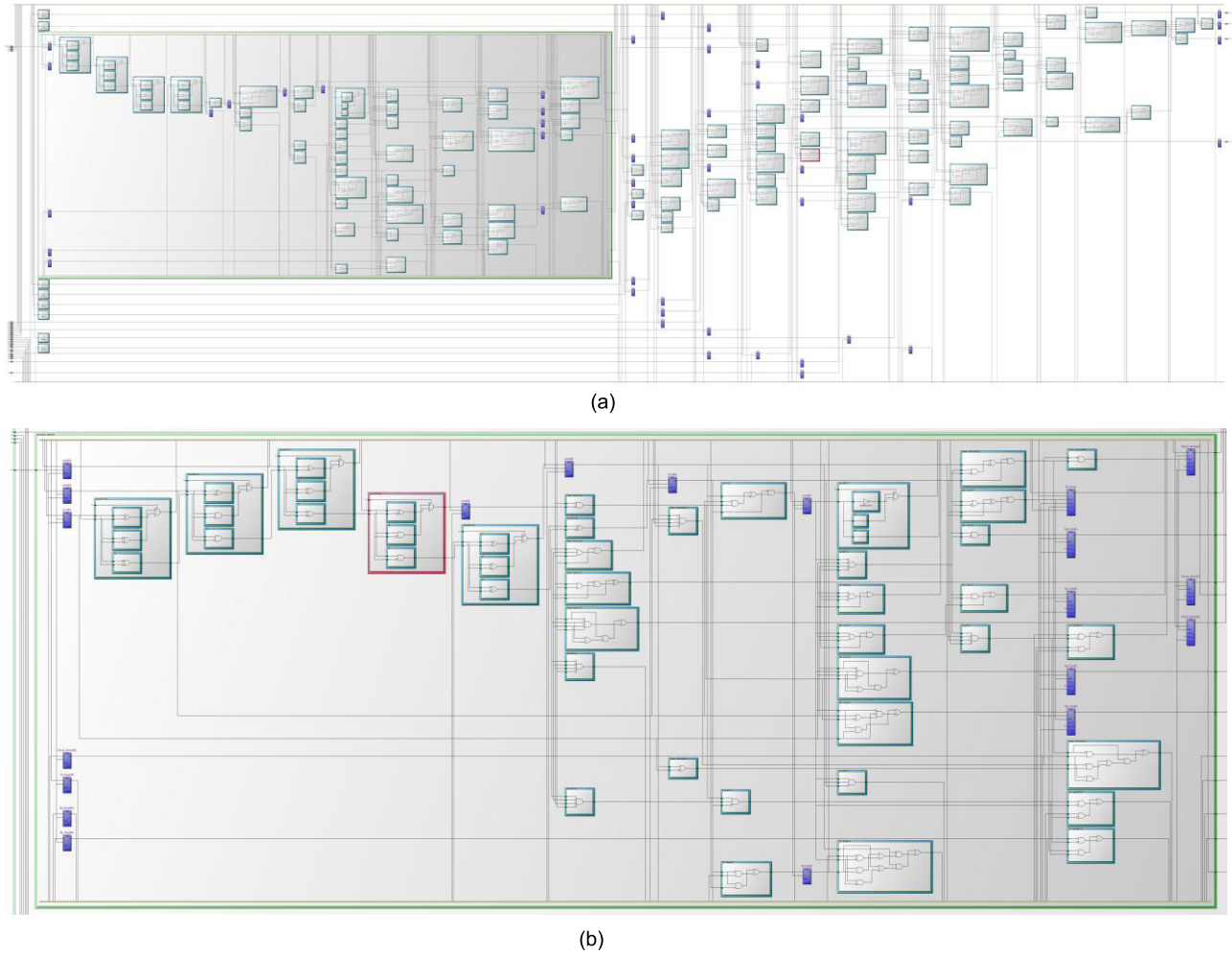

FIGURE 5. **(a) Randomly extracted portion from the gate-level structure of the demonstrated FIR design after syhthesis.** ***Note: Cells in blue color indicate the D-FFs of entire design comprising of DSP core, AES unit and proposed DLCs (D-FFs of proposed DLCs are camouflaged).*** **(b) Another randomly extracted portion from the gate-level structure of the demonstrated FIR design after syhthesis.** ***Note: D-FFs of proposed DLC camouflage with the D-FFs used in DSP design and AES unit.***

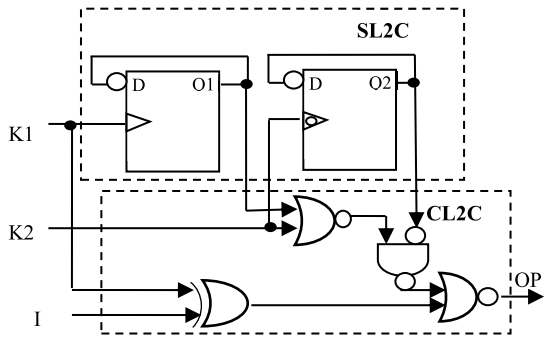

FIGURE 6. **Configuration of a proposed DLC based on key bits "10" obtained from AES output.**

2. **Handling SAT attack**: The motive of the SAT attack is to find the correct key by eliminating wrong key combinations. To eliminate wrong key combinations, distinguishing input-output (DIO) pairs are exploited. Further, to obtain a DIO pair, SAT solver (attacker) first generates assignments to input variables by formulating SAT formula then these inputs are provided to an activated IC to observe correct output. Thus, a DIO pair is obtained by observing correct output for an input assignment. SAT attack algorithm iteratively finds out the DIO pairs until all the wrong keys are eliminated where each DIO pair eliminates a subset of wrong key combinations [14].

However, proposed logic locking has been employed for DSP IP cores for which SAT attack is not feasible. This is because SAT attack is not scalable for multipliers (resulting CNF is quite large even for small size multiplications) [17], [18] and this paper targets DSP cores which comprises of several large size multiplication operations. Thus, SAT attack is not scalable even for small size DSP designs [8]. Nevertheless, assuming efficient SAT attack algorithms may be evolved for DSP cores in future, proposed logic locking can be considered as a proactive

countermeasure against SAT attack also. This is because of the following reason:

(i) The keys for the proposed DLCs are obtained from the encrypted output of the custom- AES block (with fixed secret key) in which a one-way random function is performed. Therefore, AES inputs cannot be determined from its output by an attacker without the knowledge of secret key.

(ii) Because of the security feature of proposed DLCs (discussed earlier), attacker can find correct output only if he/she applies correct keys at first trial. If attacker applies wrong key at first trial and correct key is applied afterwards then attacker becomes unable to find correct output at correct key. Thereby elimination of wrong key for each DIO pair fails and hence SAT attack algorithm becomes unsuccessful in case of proposed DLCs.

3. **Handling removal attack:**Assuming attacker is having templates of DLC structures and access to locked design netlist, he/she can attempt to perform removal attack on DLCs. However, removal attack on proposed DLCs from the locked design netlist fails because of the following reasons:

(i) Since the encrypted output of AES block is connected to the DLC-keys in the locked DSP designs, therefore DLC gate/FF structure depends on the AES output. More explicitly, internal structure of DLCs are configured by reorganizing/choosing the different combinational logic gates and +ve/−ve edge triggered D-FFs based on the output of AES block. Further, the configured structure of DLCs is only known to the designer because he/she chooses the inputs and fixed secrect keys to AES. Thereby several DLC structures are possible (depending on AES output) which are all not known to the attacker. Thus, attacker becomes unable to detect DLCs fully in the locked design as he/she is unaware of the encrypted output of AES and its corresponding secret key. Hence the attacker cannot match the reconfigured DLC structures with the available templates and his attempts to remove DLCs become unsuccessful.

(ii) After inserting DLCs in the RTL of a DSP design, resulting locked design is re-synthesized along with AES block to produce a gate level structure. Post-synthesis, number of gates and their types are changed in locked netlist because of technology specific mapping. Further, logic gates and D-FFs used in proposed DLCs are camouflaged with the similar resources of DSP designs and AES block (*note: a DSP design and AES block require several D-FFs in the form of registers used for storage purpose*). Thereby, indistinguishablility of the components (at gate-level) of individual designs hinders the attacker in identifying the components of proposed DLCs and thus, removal attack on them is impeded.

(iii) Location of DLCs insertion depends on designer selected variable (μ) and its encoding rule which is decided by the designer. Since the value of μ is not known to the attacker, therefore he/she cannot figureout the exact location and number of DLCs. Hence removal attack on DLCs is further impeded.

Removal attack on AES: The AES block itself may also be subjected to removal attack. This is because, AES architecture is publicly available and attacker can avail it to match with the AES block integrated with locked design. Thus, removal attack can be performed on it. However, in case of proposed work, it is not feasible because of the following reasons:

- In proposed work, custom-designed AES architecture (which is not publicly available) is used to integrate with locked DSP design.
- After combined synthesis of locked design integrated with AES block, components of individual designs become indistinguishable post synthesis. This hinders the identification of AES block and thus, any attempt of removal attack on it is impeded.

4. **Handling IP piracy, Overbuilding, Tojan insertion and RE attacks:** In this paper, IP cores are secured against IP piracy, Trojan insertion, overbuilding and RE attacks by employing logic locking. However, security against aforesaid attacks cannot be achieved unless the logic locking is robust. This is because the logic locking may be vulnerable to some key based attacks, removal attack etc. However, resiliency of proposed logic locking against these attacks has been discussed earlier in this subsection which shows the robustness of proposed logic locking. Therefore IP piracy, Trojan insertion, overbuilding and RE attacks are highly challenging in case of proposed work and handling of these attacks is disucussed as follows: (a) Trojans are needed to be inserted at safe places in a design which requires the understanding of functionality of the design [19]. Since functionality of the design is locked through robust DLCs, hence Trojan insertion is not possible in proposed logic locking. (b) Without knowing the correct key of locked netlist, IP piracy and overbuilding of the IP core become useless. Since proposed approach renders the probability of finding correct key in exhaustive trials almost nominal, therefore logic locking becomes very robust and IP piracy and overbuilding becomes infeasible. (c) Even if the attacker may reverse engineer the design, nevertheless he/she cannot make the unauthorized use of it without knowing the correct key.

IV. RESULTS AND ANALYSIS

The proposed approach and similar prior arts [4], [8] have been executed on Intel®Core™i3-3110M CPU with 2GB RAM and processor frequency of 2.4 GHz. Results have been evaluated and analysed for various DSP benchmarks each of size 32-bit (i.e. size of each input, output and FU is 32-bit). Resource count of designs due to proposed logic

TABLE 1. Comparison of the strength of the proposed logic locking with [4], [8] in terms of the probability of obtaining correct key in exhaustive trials (Pck) using eq. (3) and (4).

DSP Core Benchmarks	Number of key bits in locked design of related work [4]	Number of key bits in locked design of related work [8]	Number of key bits in locked design of proposed work	P_{CK} using exhaustive (2^{K_t}) trials in related work [4] [8]	P_{CK} using exhaustive (2^{K_t}) trials in proposed work	# of times security enhancement in proposed approach w.r.t. [4] [8]
IIR	384	768	192	1	1.6E-58	6.3E+57
Mesa Horner	384	768	192	1	1.6E-58	6.3E+57
DWT	384	512	128	1	2.9E-39	3.4E+38
ARF	448	1024	256	1	8.6E-78	1.2E+77
FIR	576	1280	320	1	4.7E-97	2.1E+96
JPEG IDCT	1728	5376	1344	1	2.6E-405	3.8E+404
Mesa Interpolate	1856	3328	832	1	6.2E-232	1.6E+231

locking and similar prior works were calculated using 15nm technology scale NanGate library [26]. Results of proposed work have been obtained for the following: (a) security analysis (b) encyption strength analysis (c) design area analysis. In these analyses, results of proposed work have been compared with similar prior arts [4], [8]. These comparisons show that the proposed work is capable to achieve higher strength of logic locking with lower overhead. Following subsections discuss the aforementioned analysis in details.

A. SECURITY ANALYSIS

Security of the proposed logic locking is assessed in terms of the probability of obtaining correct key in exhaustive trials (P_{ck}). The P_{ck} is evaluated using (2) which further converges into (3) for [4], [8] and (4) for proposed work (as described in section 3.3.3). Therby, (3) determines the security of [4], [8] and (4) determines the security of proposed work. Table 1 shows the comparison of P_{ck} of proproposed approach and related works [4], [8]. It is evident from the table that P_{ck} is obtained to be very less in case of proposed approach than related works. Thus, despite of encoding lesser key bits than [4], [8], proposed approach achieves higher security in terms of P_{ck} because of very low probability of obtaining correct key in exhaustive trials. The minimum value of the probability of obtaining correct key in exhaustive trials using proposed work is obtained to be 2.9E-39 (refering Table 1; DWT benchmark) which is significantly lesser than 1. Since, lower is the value of P_{ck}, higher is the security therefore proposed approach offers higher security than related works [4], [8]. Further, in case of related works, it is possible to obtain correct key in exhaustive trials therefore estimated attack time to find the correct key is finite. Where as in case of proposed work, correct key cannot be obtained in exhaustive trials except first (which is trivial) therefore, attack time required to find the correct key is estimated to be infinite *(i.e. the time consumed will be much higher than the time taken to obtain the correct key using brute-force attack). This is because if attacker missed the correct key in the first trial, he/she can**never**obtain the same in exhaustive trials and the probability of applying correct key in the first trial is almost null*. Table 2 reports the attack time comparison of proposed work and [4], [8] based on the assumption that 1 billion (10^9) keys can be applied per second (adopted from [8]).

TABLE 2. Attack time comparison of the proposed logic locking with related works [4], [8].

DSP Benchmark	Estimated attack time through proposed work	Estimated attack time through [4] (in years)	Estimated attack time through [8] (in years)
IIR	∞	1.2E+99	4.9E+214
Mesa Horner	∞	1.2E+99	4.9E+214
DWT	∞	1.2E+99	4.2E+137
ARF	∞	2.3E+118	5.7E+291
FIR	∞	7.8E+156	6.6E+368
JPEG IDCT	∞	4.8E+503	6.9E+1601
Mesa Interpolate	∞	1.6E+542	2.13E+985

Note: In proposed logic locking, correct key cannot be obtained in exhaustive trials except first and probability of obtaining correct key at first trial is $1/(2^{K_t})$

B. ENCRYPTION STRENGTH ANALYSIS

Encryption strength of proposed logic locking is compared with related works [4], [8] in terms of % of DLC keys encrypted using an AES. The encryption strength is evaluated as:

$$Enc^{strength} = \frac{N_{out}^{AES}}{N_{keybits}^{total}} \tag{5}$$

where $Enc^{strength}$ stands for the encryption strength which represents the % of key bits encrypted using one AES128, N_{AES}^{out} stands for the number of encypted output bits obtained

TABLE 3. Comparison of the encryption strength of the proposed logic locking with [4], [8].

DSP Core Benchmarks	Total # of key bits in the design of [4]	% of key bits encrypted of [4] *(using (5))*	Total # of key bits in the design of [8]	% of key bits encrypted of [8] *(using (5))*	Total # of key bits in the design using proposed	% of key bits encrypted of proposed *(using (5))*
IIR	384	33.3%	768	16.7%	192	66.7%
Mesa Horner	384	33.3%	768	16.7%	192	66.7%
DWT	384	33.3%	512	25.0%	128	100%
ARF	448	28.6%	1024	12.5%	256	50.0%
FIR	576	22.2%	1280	10.0%	320	40.0%
JPEG IDCT	1728	7.4%	5376	2.4%	1344	9.5%
Mesa Interpolate	1856	6.9%	3328	3.8%	832	15.4%

TABLE 4. Comparison of the resource count of the proposed work and [4], [8].

DSP Core Benchmarks	Resource Count of [4]		Resource Count of [8]		Resource Count of proposed work	
	NAND gates	D-FFs	NAND gates	D-FFs	NAND gates	D-FFs
IIR	46416	5216	29712	5216	27504	5408
Mesa Horner	46416	5184	33488	5184	31280	5376
DWT	46032	5216	34928	5216	33456	5344
ARF	50544	5440	41360	5440	38416	5696
FIR	59184	5408	43120	5408	39440	5728
JPEG IDCT	137328	7040	111024	7040	95568	8384
Mesa Interpolate	145968	8224	94224	8224	84656	9056

TABLE 5. % Reduction in resource count using proposed work w.r.t. [4], [8].

DSP Core Benchmarks	Proposed work					
	% reduction in NAND gates w.r.t. [4]	% increase in D-FFs w.r.t. [4]	**Overall % reduction in resource count** w.r.t. [4]	% reduction in NAND gates w.r.t. [8]	% increase in D-FFs w.r.t. [8]	**Overall % reduction in resource count** w.r.t. [8]
IIR	40.7%	3.7%	37.0%	7.4%	3.7%	3.7%
Mesa Horner	32.7%	3.7%	29.0%	6.5%	3.7%	2.8%
DWT	27.3%	2.5%	24.8%	4.2%	2.5%	1.7%
ARF	24.0%	4.7%	19.3%	7.1%	4.7%	2.4%
FIR	33.4%	5.9%	27.5%	8.5%	5.9%	2.6%
JPEG IDCT	30.4%	19.1%	11.3%	13.9%	19.1%	0%
Mesa Interpolate	42.0%	10.1%	31.9%	10.2%	10.1%	0.1%

from AES128 and $N_{keybits}^{total}$ stands for the total number of key bits in a design. Table 3 presents the encryption strength of related works [4], [8] and proposed approach. Results presented in the table shows that proposed approach is capable to encrypt more key-bits w.r.t. related works. This is because proposed work encodes lesser number of key bits for a design than related works while concurrently enhancing the security of proposed logic locking. Thus $N_{keybits}^{total}$ becomes lesser for proposed approach and $Enc^{strength}$ increases significantly compared to related works.

C. DESIGN AREA ANALYSIS

Design area of proposed and related works [4], [8] is anaysed in terms of resource count. The resource count of related works and proposed work is calculated in terms of NAND

gates and D-FFs. Table 4 compares the resource count of locked DSP designs using proposed approach with that of related works [4], [8]. As shown in Table 4, the NAND gates count of proposed work is reduced whereas D-FFs count is increased w.r.t [4], [8]. Increase in D-FFs count in case of proposed work is because of incorporation of D-FFs as logic locking cell in the proposed DLCs. Further, Table 5 shows the % reduction in NAND gates and % increase in D-FFs through proposed approach compared to related works. Furthermore, table 5 presents the overall % reduction in resource count achieved through proposed approach. As resource count consists of NAND gates count and D-FFs count, % reduction in resource count is calculated as an algebraic summation of '% reduction in NAND gate count' and '% increase in D-FF count'. It is evident from the Table 5 that proposed work achieves on average 25.8% reduction in resource count w.r.t. [4] and on average 1.9% reduction w.r.t. [8]. Theses results signify that proposed approach incurs lower resource count and hence lower area than related works [4], [8]. The proposed approach achieves lower overhead than state-of-art [4], [8] because the proposed approach uses particle swarm optimization based design space exploration (PSO-DSE) to generate the resource configuration required to build the DSP circuits. However, [4] does not use PSO-DSE in their approach. Further [8] uses 8-bit keys per output bit which makes the locking block complex in terms of gate count, while proposed DLC uses 2-bit keys per output bit. Thus [8] incurs higher overhead than proposed approach.

V. CONCLUSION

This paper proposes the novel structures of DLCs to employ logic locking for the security of DSP IP cores. Strength of both the proposed logic locking and the similar prior works is assessed in terms of the probability of obtaining correct key in exhaustive trials. Results and analysis show that proposed work achieved higher security at lower overhead w.r.t. similar prior works.

REFERENCES

[1] R. Schneiderman, "DSPs evolving in consumer electronics applications," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 6–10, May 2010.

[2] A. Sengupta, "Hardware security of CE devices [hardware matters]," *IEEE Consum. Electron. Mag.*, vol. 6, no. 1, pp. 130–133, Jan. 2017.

[3] S. M. Plaza and I. L. Markov, "Solving the third-shift problem in ic piracy with test-aware logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 6, pp. 961–971, Jun. 2015.

[4] M. Yasin, J. Rajendran, O. Sinanoglu, and R. Karri, "On improving the security of logic locking," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 9, pp. 1411–1424, Sep. 2016.

[5] R. Torrance and D. James, "The state-of-the-art in IC reverse engineering," in *Proc. 11th Int. Workshop CHES*, vol. 5747, 2009, pp. 363–381.

[6] K. Shamsi, M. Li, T. Meade, Z. Zhao, D. Z. Pan, and Y. Jin, "AppSAT: Approximately deobfuscating integrated circuits," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, McLean, VA, USA, May 2017, pp. 95–100.

[7] J. Rajendran, M. Sam, O. Sinanoglu, and R. Karri, "Security analysis of integrated circuit camouflaging," in *Proc. ACM SIGSAC*, Berlin, Germany, Nov. 2013, pp. 709–720.

[8] A. Sengupta, D. Kachave, and D. Roy, "Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 4, pp. 604–616, Apr. 2019. doi: 10.1109/TCAD.2018.2818720.

[9] Y.-W. Lee and N. A. Touba, "Improving logic obfuscation via logic cone analysis," in *Proc. Latin-Amer. Test Symp. (LATS)*, Puerto Vallarta, Mexico, Mar. 2015, pp. 1–6.

[10] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending piracy of integrated circuits," in *Proc. Design, Autom. Test Eur.*, Munich, Germany, Mar. 2008, pp. 1069–1074.

[11] J. Zhang, "A practical logic obfuscation technique for hardware security," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 3, pp. 1193–1197, Mar. 2016.

[12] A. Sengupta and S. P. Mohanty, "Functional obfuscation of DSP cores using robust logic locking and encryption," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Hong Kong, China, Jul. 2018, pp. 709–713.

[13] A. Sengupta, R. Sedaghat, and Z. Zeng, "A high level synthesis design flow with a novel approach for efficient design space exploration in case of multi-parametric optimization objective," *Microelectron. Rel.*, vol. 50, no. 3, pp. 424–437, Mar. 2010.

[14] P. Subramanyan, S. Ray, and S. Malik, "Evaluating the security of logic encryption algorithms," in *Proc. IEEE Int. Symp. Hardw. Oriented Secur. Trust (HOST)*, Washington, DC, USA, May 2015, pp. 137–143.

[15] M. Yasin, B. Mazumdar, O. Sinanoglu, and J. Rajendran, "Removal attacks on logic locking and camouflaging techniques," *IEEE Trans. Emerg. Topics Comput.*, to be published. doi: 10.1109/TETC.2017.2740364.

[16] Y. Xie and A. Srivastava, "Anti-SAT: Mitigating SAT attack on logic locking," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 2, pp. 199–207, Feb. 2019. doi: 10.1109/TCAD.2018.2801220.

[17] S. Chakraborty, A. Gupta, and R. Jain, *Matching Multiplications in Bit-Vector Formulas*. Paris, France: Springer, 2017, pp. 131–150.

[18] F. Azevedo and P. Barahona, "Modelling digital circuits problems with set constraints," in *Computational logic-CL*. London, U.K.: Springer. 2000, pp. 414–428.

[19] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "TL-HLS: Methodology for low cost hardware trojan security aware scheduling with optimal loop unrolling factor during high level synthesis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 4, pp. 655–668, Apr. 2017.

[20] M. Yasin and O. Sinanoglu, "Evolution of logic locking," in *Proc. IFIP/IEEE Int. Conf. Very Large Scale Integr. (VLSI-SoC)*, Abu Dhabi, UAE, Oct. 2017, pp. 1–6.

[21] A. Sengupta and M. Rathor, "Protecting DSP kernels using robust hologram-based obfuscation," *IEEE Trans. Consum. Electron.*, vol. 65, no. 1, pp. 99–108, Feb. 2019.

[22] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "Low-cost obfuscated JPEG CODEC IP core for secure CE hardware," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 365–374, Aug. 2018.

[23] A. Sengupta, S. P. Mohanty, F. Pescador, and P. Corcoran, "Multi-phase obfuscation of fault secured DSP designs with enhanced security feature," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 356–364, Aug. 2018.

[24] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "DSP design protection in CE through algorithmic transformation based structural obfuscation," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 467–476, Nov. 2017.

[25] Y. Lao and K. K. Parhi, "Obfuscating DSP circuits via high-level transformations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 5, pp. 819–830, May 2015.

[26] *NanGate 15 nm Open Cell Library*. Accessed: Jan. 2019. [Online]. Available: http://www.nangate.com/?pageid=2328

MAHENDRA RATHOR (M'18) received the M.E. degree in electronics engineering, in 2014. He is currently pursuing the Ph.D. degree in computer science and engineering with IIT Indore.

ANIRBAN SENGUPTA (M'09–SM'17) received the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from Ryerson University, Toronto, Canada.

He has more than 200 publications and patents. He has supervised more than 30 candidates. He is the author of a Book from IET on IP core protection and hardware-assisted security for consumer electronics published in U.K., in 2019. He is currently an Associate Professor of computer science and engineering with IIT Indore, where he directs the Research Lab on CAD for Consumer Electronics Hardware Device Security and Reliability. He is also an elected Fellow of IET and a Fellow of the British Computer Society (FBCS), U.K. He has been awarded the prestigious IEEE Distinguished Lecturer by the IEEE Consumer Electronics Society, in 2017, and the IEEE Distinguished Visitor by the IEEE Computer Society, in 2019. More than a dozen of his IEEE publications have appeared in "Top 50 Most Popular Articles" with a few in "Top 5 Most Popular Articles" from the IEEE Periodicals. He was the General/Conference Chair of the 37th IEEE International Symposium on Consumer Electronics (ICCE) 2019, Las Vegas, and the Technical Program Chair of the 36th IEEE International Conference on Consumer Electronics (ICCE) 2018, Las Vegas, the 9th IEEE International Conference on Consumer Electronics (ICCE), Berlin, in 2019, the 15th IEEE International Conference on Information Technology (ICIT) 2016, and the Third IEEE International Symposium on Nanoelectronic and Information Systems (iNIS) 2017. He is also the Deputy Editor-in-Chief of the *IET Computers* and *Digital Techniques* journal that has a publishing history of over 40 years and the Editor-in-Chief of the IEEE VLSI Circuits and Systems Letter of the IEEE Computer Society TCVLSI. He is also the Chairman of the IEEE Computer Society TCVLSI. He currently serves/served in several editorial positions as a Senior Editor, an Associate Editor, an Editor, and a Guest Editor for several IEEE Transactions/journals, IET, and Elsevier journals, including the IEEE Transactions on Aerospace and Electronic Systems (TAES), IEEE Transactions on VLSI Systems, IEEE Transactions on Consumer Electronics, IEEE Access journal, *IET Journal on Computer* and *Digital Techniques*, IEEE Consumer Electronics, IEEE Canadian Journal of Electrical and Computer Engineering, IEEE VLSI Circuits and Systems Letter, and *Microelectronics Journal* (Elsevier). He also serves as the Guest Editor of the IEEE Transactions on VLSI Systems, IEEE Access, and *IET Computers* and *Digital Techniques*. He has successfully commissioned special issues in the IEEE TVLSI, IEEE TCAD, IET CDT, IEEE Access, and IEEE CEM. He is also a registered Professional Engineer of Ontario (P.Eng.).