HCLL-Lock：機能攻撃への高い耐性を持つハードウェアIP保護手法

森下 彩音[1,a)]　一岡 知佑[1]　楊 明宇[1]　原 祐子[1]

概要：本論文では，Look-Up Table (LUT) とダミーループ構造を組み合わせ，高い SAT 攻撃耐性と多くの主出力の保護を両立可能な新たな IP 設計手法である HCLL-Lock を提案する．これは集積回路（IC）の製造を請け負う企業による設計情報（IP）の盗用や IC の過剰生産を防ぐための手段として注目を集めているロジックロッキングの手法の 1 つである．近年の IP 保護手法の多くは，充足可能性問題 (SAT) ソルバを応用した SAT 攻撃への耐性を高めるために，ロックによって影響される出力数を減らしている．しかし，この方針では，正しい鍵を入力しなくても回路がある程度正しく動く可能性が高い．この問題を解決するため，HCLL-Lock ではできる限り多くの主出力を保護するように LUT を置換する．LUT は任意の論理関数を実現可能な高い柔軟性を有する一方で，一般に回路規模や遅延の観点でオーバヘッドが大きいという欠点がある．HCLL-Lock ではさらにダミーループを併用することにより，回路オーバヘッドあたりの SAT 攻撃耐性を高めることを目指す．実験では提案手法を RISC-V プロセッサに適用し，回路面積，遅延，セキュリティのトレードオフを評価した．その結果，既存手法と比較して，SAT 攻撃耐性を維持したまま保護主出力数を改善できており，HCLL-Lock の有用性を確認できた．

HCLL-Lock: Hardware IP protection method with high resilience to functional attacks

Ayane Morishita[1,a)]　Tomosuke Ichioka[1]　Mingyu Yang[1]　Yuko Hara[1]

Abstract: In this paper, we propose HCLL-Lock, a new hardware design method that combines look-up tables (LUTs) and dummy loops to protect the hardware design (intellectual property; IP) against various kinds of attacks (IP theft, overproduction, etc.). This is one of logic locking techniques gaining attention as an effective security-for-design means. Most recent IP protection methods try to reduce the number of outputs affected by the "lock" in order to increase resilience to input/output query attacks using Satisfiability (SAT) solvers. However, this approach is not secure enough in the sense that the locked circuit still functions almost correctly even without the correct key. To address this problem, HCLL-Lock protects as many primary outputs as possible by LUTs while achieving high SAT attack resilience. To mitigate the circuit overhead incurred by LUTs, we also integrate dummy loops enhance SAT attack resilience per circuit overhead. In our experiments, we applied HCLL-Lock to a RISC-V processor and evaluated the trade-off between circuit area, delay, and security. The results demonstrated that HCLL-Lock is superior to existing methods in terms of both high SAT attack resilience and protection of multiple primary outputs.

1. 序論

近年，半導体技術の進展による集積回路 (IC) 製造コストの増大に伴い，設計と製造の分業化が急速に進展している．これにより，ハードウェア知的財産（IP）の保護は，これまで以上に重要な課題として注目されている．とりわけ，信頼性に乏しいファウンドリやユーザによるリバースエンジニアリング，設計情報の不正取得，過剰生産といった脅威に対抗するため，設計段階における IP 保護技術の確立が求められている [1].

こうした背景の下，ロジックロッキング [2] は，秘密鍵

[1] 東京科学大学，東京都目黒区大岡山 2-12-1，
Institute of Science Tokyo, 2-12-1 Ookayama, Meguro-ku, Tokyo
[a)] morishita.a.fe2b@m.isct.ac.jp

に依存する回路構造を挿入し，正しい鍵を入力しない限り回路が正常に動作しないよう設計することで，IP の不正利用を防止する手法として広く注目を集めている．一方で，充足可能性問題 (SAT) のソルバを用いた SAT 攻撃 [3] に代表される機能攻撃によって，ロジックロッキングの効果が破られるリスクが報告されている．近年のロック手法では，誤った鍵を入力した際にも出力が正しい値に近づくように設計し，SAT 攻撃による鍵候補の削減を困難にするアプローチが主流となっている [4][5][6][7][8]．これは，ロジックロッキングの影響を受ける出力の数を意図的に減らすことにより，DIP（区別可能な入力パターン）によって排除される誤った鍵の数を抑え，攻撃の探索回数を増加させるものである．しかしこの方針は，保護できる主出力が限定されるという欠点を持ち，誤った鍵でも回路がある程度正常に動作する可能性を残してしまう．このように，多くの主出力の保護と SAT 攻撃耐性の両立は困難であり，両者はしばしばトレードオフの関係にあるとされる．

本研究では，この課題に対して，Look-Up Table（LUT）による回路ゲートの置換と，ループ構造の複雑化を組み合わせることで，多くの主出力の保護と高い SAT 攻撃耐性を両立可能な新たなロジックロッキング手法 HCLL-Lock（High Corruptibility using LUT Loop Lock）を提案する．本手法では，できる限り多くの主出力を保護するような LUT 配置戦略を採用するとともに，ダミーループ構造を導入することで SAT 攻撃の計算コストを高め，耐性の向上を図る．実験では，提案手法を RISC-V プロセッサに適用し，回路面積や遅延，セキュリティのトレードオフを評価した．その結果，既存手法と比較して，同等または少ない回路オーバヘッドでより多くの出力を保護でき，さらに，SAT 攻撃への高い耐性を維持できることを実証した．

本論文の構成は以下の通りである．第 2 章では，本論文の根底となる技術であるロジックロッキングおよび SAT 攻撃の基礎知識や既存研究を紹介する．第 3 章では，LUT 置換とループ構造を組み合わせたロジックロッキングの手法を提案する．第 4 章では，提案手法を RISC-V プロセッサに適用し，面積オーバヘッドや遅延オーバヘッドに対するセキュリティ効果を評価する．第 5 章では，本研究の総括と今後の課題について述べる．

2. 前提知識

本章では，本研究の先行研究であるロジックロッキング及び攻撃手法について説明する．2.1 節では，ロジックロッキングの概要について説明する．2.2 節，2.3 節では，ロジックロッキングへの代表的な攻撃手法である機能攻撃，特に SAT 攻撃と，それらへの対策について説明する．2.4 節，2.5 節では，ロジックロッキングの手法の中でも特に本研究の提案手法との関連性が大きい手法について説明する．

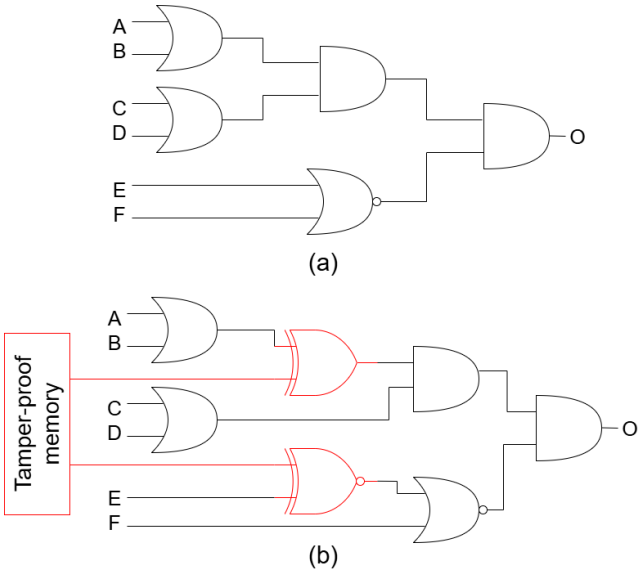

図 1 XOR/XNOR Insertion,
(a) オリジナルの回路，(b) ロック後の回路

Fig. 1 XOR/XNOR Insertion,
(a)original circuit, (b)circuit after logic locking

2.1 ロジックロッキング

ロジックロッキングとは，元の回路に XOR/XNOR ゲートや MUX といった鍵入力に依存する要素を追加することで，正しい鍵を入力しない限り回路が正しく機能しないようにする IP 保護手法である．誤った鍵が入力された場合には回路が誤動作する，もしくは完全に動作しなくなるように設計される．

図 1 は，XOR/XNOR ゲートを用いたロジックロッキングの例を示しており，元の回路に対して 2 ビットの秘密鍵を導入している．図 1(b) の回路は，XOR の鍵に 0，XNOR の鍵に 1 を入力すれば図 1(a) と同じように動作するが，それ以外の鍵を入力すると回路が正しく機能しない．この鍵は耐タンパーメモリに格納され，攻撃者による不正取得が困難な設計となっている．鍵のビット数を増加させることで，正しい鍵を推定するための探索空間は指数関数的に拡大し，攻撃が難しくなる．

ここで，本研究において重要な指標である保護主出力について説明する．本論文においての保護主出力とは，誤った鍵を入力した際に壊れる（ビット反転する）可能性のある主出力を指す．前述した XOR/XNOR Insertion や 2.4 節で説明する LUT を用いたロジックロッキングであれば，挿入された XOR や XNOR，LUT のファンアウトコーン（その論理ゲートから出力に向かってトレースしたときに通過し得る配線や論理ゲートの集合）に含まれる主出力が保護主出力にあたる．回路の全主出力に対して保護主出力の割合が大きければ大きいほど，誤った鍵を入力した際の回路の出力を予測することが困難となる．

2.2 機能攻撃

ロジックロッキングを施された回路は，正しい鍵を入力しない限り本来の機能を発揮しない設計となっている．しかしながら，攻撃者が鍵を解析し，回路の正しい動作を再現可能であれば，ロジックロッキングの効果は失われる．このように，鍵や内部状態を出力挙動から推定する攻撃手法は，機能攻撃と呼ばれる．

この種の攻撃の中でも代表的なものが SAT 攻撃（SAT attack）[3] である．SAT 攻撃は，ロックされた回路の鍵候補を削減しながら正解を探索する反復的な攻撃手法であり，SAT ソルバによって充足可能性（SAT）問題を効率的に解くことにより実現される．

SAT 攻撃では，まずロックされた回路を SAT ソルバに入力可能な形式へ変換し，そこから DIP（Distinguishing Input Pattern）と呼ばれる入力パターンを求める．得られた DIP をオラクル回路に適用することで正しい出力が得られ，その出力と一致しない候補鍵は排除される．この操作を繰り返すことで鍵空間を段階的に削減し，最終的に正しい鍵を特定することが可能となる．

SAT 攻撃は以下の前提の上で成立する [9].

- 設計者は信頼できる
- ファウンドリを含む IC 製造に関わる第三者およびユーザーは信頼できない
- 攻撃者はリバースエンジニアリングされたネットリスト，正しい鍵が埋め込まれた動作可能な回路（オラクル），ロックが施された場所とその内容を知っている

2.3 機能攻撃への対策

SAT 攻撃の実行時間 T は，λ をイタレーションの回数，t_i を i 番目のイタレーションにかかる時間とすると，以下のように表される [9].

$$T = \sum_{i=1}^{\lambda} t_i$$

この実行時間を延長するためには，イタレーション回数 λ を増やす [4][5][6][7][8]，または各イタレーションの実行時間 t_i を引き延ばす [10][11][12]，という 2 つのアプローチが考えられる．

従来手法の多くは前者を採用しており，正しい鍵と誤った鍵に対する出力を類似させる，すなわち保護主出力を少なくすることで，DIP によって区別できる誤った鍵の数を減らすよう設計されている．これにより，攻撃に必要なイタレーション数が増加し，SAT 攻撃の実行時間が長くなる．しかしながらこの手法は，出力の保護範囲が限定されやすく，結果として誤った鍵でも回路が部分的に正しく動作する可能性を残す．さらに，このデメリットを補うために，保護主出力の数が多い他手法を併用したとしても，AppSAT[13] や Double DIP[14] といった近似攻撃に対しては脆弱である．このように，多くの主出力の保護と高い SAT 攻撃耐性の両立は技術的に難しく，トレードオフの関係にある．

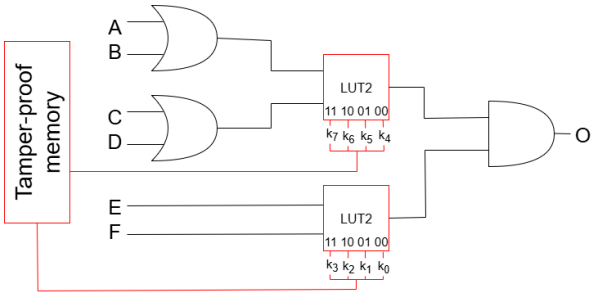

図 2 LUT を用いたロジックロッキング
Fig. 2 Logic Locking using LUT

2.4 LUT を用いたロジックロッキング

Look-Up Table（LUT）とは，IC 製造後であっても任意の入力に対して出力を再構成可能な素子である．たとえば，2^i ビット入力，2^o ビット出力の LUT は，$(2^o)^{2^i}$ 通りの論理関数を実現できる．

LUT を用いたロジックロッキング [4][15][16] では，元の回路の論理ゲートを LUT に置換し，その設定値を秘密鍵として活用することで回路情報を秘匿する．攻撃者が元の論理を知らなければ，LUT を正しく設定することができず，回路を意図通りに動作させることが困難となる．

図 2 は，図 1(a) で示されたオリジナルの回路に対し，2 つの 2 入力 LUT を導入したロジックロッキングの例である．正しい鍵を入力しなければ，元の機能を再現することはできず，たった 2 つの LUT であっても 256 通りもの鍵候補が存在することになる．このように，LUT を用いた手法は鍵空間を容易に拡大可能であり，SAT 攻撃に対して一定の耐性を持つ点が評価されている．

2.5 ダミーループを用いたロジックロッキング

ロジックロッキングの中には，ゲートの追加や置換ではなく，回路内にダミーループ構造を挿入することで SAT 攻撃耐性を高める手法も存在する．このアプローチは，前節で述べた 2 つの方針のうち，各イタレーションの実行時間 t_i を増加させることを目的とするものである．

基本的なループ構造は，あるゲートの出力をそのファンインコーン（その論理ゲートから入力に向かってトレースしたときに通過し得る配線や論理ゲートの集合）に含まれるゲートのうちの 1 つの入力に接続し，MUX によってループの有効・無効を鍵に応じて切り替えることで構成される．ループが有効な状態では，回路は有向非巡回グラフ（DAG）として表現できなくなり，SAT ソルバでの解析が困難になる．

図 3 は，図 1(a) のオリジナルの回路に対してループ構造

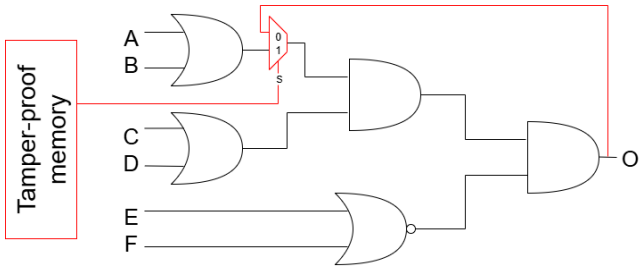

図 3 ループ構造を用いたロジックロッキング
Fig. 3 Logic Locking using loop structure

を用いたロジックロッキングを適用した例を示している．この例では，セレクタ信号が正しい鍵である「1」であるときのみループが無効化され，正常な動作が可能となる．

ただし，単純なループ構造では攻撃者による解除が容易であるため，ループを解消する方法が複数考えられるような複雑化ループの構築が有効である [10]．この場合，ループ経路上の各ゲートに対して，ランダムに選んだゲートを MUX で接続し，ループを解除する方法を一意に定められないようにする工夫が施される．

これに対しても，CycSAT[17] や IcySAT[18] といった対抗手法が開発されているが，ループ生成鍵の検出と通常の SAT 攻撃の実行の両方を要するため，依然として高い計算コストを要する．本研究では，実験における SAT 攻撃評価のために IcySAT-II を使用する．

3. 機能攻撃への高い耐性を持つハードウェア IP 設計手法

本章では，提案手法である HCLL-Lock の設計思想および具体的なアルゴリズムについて述べる．まず 3.1 節で本手法の目的および想定する脅威モデルを明確にし，3.2 節にて手法の全体構成を詳述する．

3.1 目的・脅威モデル

本研究では，回路オーバヘッドを抑制しつつ，機能攻撃に対する強固な耐性を実現することを目的として，LUT によるロジックゲートの置換と複雑なループ構造の導入を組み合わせたロジックロッキング手法「HCLL-Lock」を提案する．

従来手法の一例である LUT-Lock[4] は，回路ゲートを LUT に置換することで保護を実現するが，十分な SAT 攻撃耐性を得るためには多数のゲートを置換する必要があり，結果として面積・遅延オーバヘッドが大きくなりやすい．一方，LUT 置換とフォールスループを併用する FIFOL-Lock[19] では，SAT 攻撃耐性を重視するあまり，LUT の影響を受ける出力が極端に少なくなるという問題がある．

本研究の HCLL-Lock は，これらの課題を克服し，回路オーバヘッドを抑えつつ，高い SAT 攻撃耐性と多くの主出力の保護の両立を目指すものである．

想定する脅威モデルは，2.2 節で説明した SAT 攻撃の脅威モデルに準ずる．

Algorithm 1 HCLL-Lock

Require: $c, target\_num$
Ensure: c_lock
1: $candidates \leftarrow c.outputs()$
2: $sort(candidates, fanincone\_size, "descending\_order", c)$
3: $num\_of\_replaced \leftarrow 0$
4: $affected\_outputs \leftarrow \emptyset$
5: $c\_lock \leftarrow c.copy()$
6: **while** $num\_of\_replaced < target\_num$ **do**
7: $candidate = candidates.pop(0)$
8: $c\_lock.replace\_with\_lut(candidate)$
9: $num\_of\_replaced \leftarrow num\_of\_replaced + 1$
10: **for all** $gate \leftarrow c.fanin(candidate)$ **do**
11: **for all** $non\_adjacent\_gate \leftarrow c.fanin(gate)$ **do**
12: $candidates \leftarrow candidates \cup non\_adjacent\_gate$
13: **end for**
14: **end for**
15: **for all** $gate \leftarrow c.fanout(candidate)$ **do**
16: **for all** $non\_adjacent\_gate \leftarrow c.fanout(gate)$ **do**
17: $candidates \leftarrow candidates \cup non\_adjacent\_gate$
18: **end for**
19: **end for**
20: $sort(candidates, num\_of\_new\_affected\_outputs,$ "descending_order", c)
21: **end while**
22: **return** c_lock

3.2 機能攻撃耐性を持つハードウェア IP 設計手法

提案手法 HCLL-Lock は，次の 2 段階から構成される．

(1) LUT 置換による主出力の保護
(2) その LUT をフィードバック起点とする複雑化ループの挿入による SAT 攻撃耐性の向上

まず，LUT 置換の手順をアルゴリズム 1 に示す．オリジナルの回路 c と LUT に置換する論理ゲートの数 $target\_num$ を入力として受け取り，ロック後の回路 c_lock を返す．このアルゴリズムではまず，$candidates$ に回路の主出力を代入し，ファンインコーンが大きい順に並べ替える．次に，LUT への置換が完了した論理ゲートの数 $num\_of\_replaced$ を 0 に，置換したゲートのファンアウトコーンに含まれる主出力の集合 $affected\_outputs$ を空集合に初期化し，c_lock に c をコピーする．そして，引数 $target\_num$ で指定された数の論理ゲートを置換し終えるまで，以下の操作を繰り返す．まず，$candidates$ の先頭のゲートを LUT に置き換え，$num\_of\_replaced$ をインクリメントし，置換したゲートとは連続していないゲートを $candidates$ に追加する．そして $candidates$ を，$num\_of\_new\_affected\_outputs$（ファンアウトコーンに含まれる主出力のうち，まだ LUT への置換によって保護されていない主出力の数）が大きい順に並べ替える．

連続するゲートを LUT に置換することを防ぐという戦

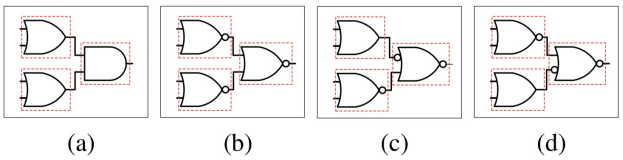

図 4 ド・モルガンの法則の適用に基づくゲート変換 (a)OR-AND (b)NOR-NOR (c) カスタム 1 (d) カスタム 2[4]

Fig. 4 Gate conversion based on the application of De-Morgan's law (a)OR-AND (b)NOR-NOR (c)Custom1 (d)Custom2[4]

略は，既存研究 [4] において NB2（Not Back-to-Back）という名前で提案されている．例えば，**図 4** に示すように，関数 $(A \vee B) \wedge (C \vee D)$ を表現している 3 つの論理回路をすべて 2 入力 LUT を用いてロックする場合，ド・モルガンの法則に基づいた 4 通りの論理構成が可能である．よって，元々は 1 通りだった正しい鍵の数が 4 通りとなってしまう．このように，連続するゲートを LUT で置換すると正しい鍵の候補が指数関数的に増加し，SAT 攻撃耐性が低下する危険性がある．そのため HCLL-Lock では，LUT に置換するゲートの確定後，置換するゲートの 1 段手前のゲート（すなわちファンイン）をスキップし，そのスキップしたゲートの 1 段手前のファンインを置換候補に追加する．ファンアウトについても同様である．これによって，連続するゲートを LUT に置換してしまうことを防いでいる．

また，20 行目のソートについて，$num\_of\_new\_affected\_outputs$ の数が等しいゲート同士では，出力の SPS（Signal Probability Skew）の高いゲートを優先するようにしている．SPS とは，あるゲートの出力が「0 になる確率」と「1 になる確率」の差の絶対値である．SPS が高いほど制御が難しいため，LUT に置換すれば SAT ソルバが DIP を見つけにくくなる．

LUT に置換するゲートが決定したら，次にループ構造を追加する．まず，ループの長さ L（ループに含まれる論理ゲートの数）を決める．次に，置換された LUT のファンインコーンに含まれるゲートのうち，LUT への長さ L のパスが存在するものの中から任意の 1 つを選び，置換された LUT の出力をそのゲートの入力の 1 つに接続することで，MUX を用いてループを作る．ループパスが決定したら，2.5 節で説明したような方法でループを複雑化する．なお，LUT に置換されたゲートが回路の入力に近い場合，指定する長さのループを作ることができないことがあるため，実際に作られるループの数は LUT 置換数よりも少なくなる．

図 5 は HCLL-Lock の適用例を示す．この例では 2 入力の LUT を 3 つ用い，長さ 2 のループを 1 つ作るとする．図 5(a) がオリジナルの回路，図 5(b) が LUT 置換後の回路，図 5(c) が単純なループを作った回路，図 5(d) がループを複雑化した回路，すなわち提案手法 HCLL-Lock の適用が完了した回路である．

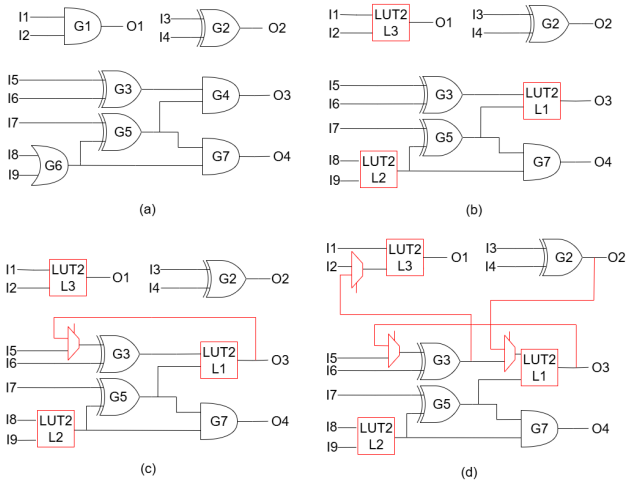

図 5 HCLL-Lock の適用例，(a) オリジナルの回路，(b)LUT 置換後の回路，(c) 単純なループ作成後の回路，(d) ループ複雑化後の回路

Fig. 5 Examples of HCLL-Lock application, (a) original circuit, (b) circuit after LUT replacement, (c) circuit after simple loop creation, (d) circuit after loop complication

まず，主出力の中で最も大きなファンインコーンを持つ O3 の直前のゲート G4 を LUT に置換する（L1）．NB2 の戦略により，L1 のファンインに含まれるゲート G3，G5 は置換候補から外れる．この時点での $candidates$ の要素は G1，G2，G6，G7 であり，どのゲートを置換したとしても $num\_of\_new\_affected\_outputs$ の数は 1 であるため，最も SPS の高いゲートが置換される．ここでは G6 とした（L2）．NB2 によって出力 O4 の直前のゲート G7 は置換候補から外れるため，次の置換候補は出力 O1 を持つゲート G1 か O2 を持つゲート G2 のうち，SPS の高い方となる．今回は G1 を置換している（L3）．この一連の置換によって図 5(b) の回路ができる．

次に，長さ 2 のループを 1 つ作る．置換した 3 つの LUT のうち，ループのフィードバック起点にできるのは L1 のみである．よって，L1 の出力から前段のゲートの入力へのフィードバックを作り，図 5(c) のようにする．そして最後に，このループを 2.5 節で述べた手順で複雑化する．具体的には，L1 の入力の 1 つに MUX を接続し，ランダムに選ばれたゲート G2 の出力を伝えるのか元から接続されているゲート G3 の出力を伝えるのか選べるようにする．また，ゲート G3 のファンアウトが 1 であるので，ランダムに選んだゲート L3 の入力の 1 つに MUX を接続し，G3 の出力を伝えるのか元の入力 I2 を伝えるのかを選べるようにする．これにより，図 5(d) の回路が得られる．

4. 実験

本章では，提案手法といくつかの既存手法を実際の回路

に適用し，セキュリティやオーバヘッドについて評価した．4.1 節で実験環境や設定について説明し，4.2 節と 4.3 節で各種オーバヘッドあたりのセキュリティを評価する．4.4 節では，実験結果を踏まえた提案手法の総合的な評価を述べる．

4.1 実験設定

比較対象として，以下の 3 つの手法を評価した．

- LUT-Lock[4]：LUT 置換のみで回路をロックする手法．充分なセキュリティを担保しようとすると回路オーバヘッドが大きくなりやすい．
- FIFOL-Lock[19]：LUT 置換とダミーループを併用する手法．保護主出力数のみが極端に少なくなるような LUT 置換アルゴリズムを用いる．
- HCLL-Lock（提案手法）：LUT 置換とダミーループを併用する手法．保護主出力数を大きくするような LUT 置換アルゴリズムを用いる．

各手法について，回路面積および遅延のオーバヘッドあたりのセキュリティ効果（SAT 攻撃への耐性，および保護主出力数）を定量的に評価した．LUT に置換する論理ゲートの数は 5 個から開始し，5 個ずつ増加させて比較を行った．FIFOL-Lock と HCLL-Lock のループの長さは 5 とし，2〜4 入力の LUT を使用した．なお，NOT や BUF のような 1 入力ゲートは，LUT によって実装可能な論理関数の数がわずか 2 通り（恒等関数と反転）のみであり，セキュリティ対効果が低いため，置換対象から除外する．

本研究では，評価対象として，2 段階のパイプラインを備える RISC-V プロセッサ Ibex[20] のうち，2 段目のパイプラインステージ（デコードから実行まで）のデコーダモジュールをロック対象とした．デコーダはプロセッサの制御に大きく関与するため，この部分をロックすることで，誤った鍵による誤動作がプロセッサ全体に伝播しやすくなる．また，LUT は高い柔軟性を持つ一方で，回路規模や遅延の面でオーバヘッドが大きくなる傾向があるが，適用範囲をデコーダモジュールに限定することで，過剰なオーバヘッドを抑制できる．

論理合成には Synopsys 社の Design Compiler を用い，セルライブラリは Nangate 15nm Open Cell Library[21] を使用した．動作周波数は 200MHz に設定し，LUT に対して最適化がかからないよう制約を加えている．面積・遅延オーバヘッドは，デコーダ単体ではなく，Ibex 全体を基準として評価した．すべての手法において，ロック後の回路はオリジナルの回路に対して遅延増加はあったが，制約の 5ns (200MHz) を違反することはなかった．

SAT 攻撃は，Ubuntu 20.04 LTS（WSL 環境）上において，Intel Xeon w7-3465X（28 コア，2.49GHz，75MB L3 キャッシュ）と 251GiB のメモリを搭載した単一ソケット構成で実施した．SAT 攻撃には NEOS（Netlist Encryption and Obfuscation Suite）[22] に含まれる IcySAT-II を用い，タイムアウトを 24 時間と設定し，正しい鍵を特定するまでの時間を測定した．グラフには，2 回の平均実行時間をプロットしている．

4.2 保護主出力割合

図 6，**図 7** は各種オーバヘッドと保護主出力の割合の関係を示している．横軸は Ibex 全体での面積オーバヘッドや遅延オーバヘッド（%），縦軸は保護主出力の割合（%）を表している．

LUT-Lock では，LUT 置換数の増加に伴い保護主出力の割合は一定程度までは増えるものの，ある閾値を超えると効果は頭打ちになる．この飽和は，連続するゲートを LUT に置換しない NB2 戦略の影響で置換候補が限定されるためと考えられる．一方，FIFOL-Lock は，LUT 置換の影響を受ける出力を意図的に減らすアルゴリズムであるため，LUT 置換数を増やしていっても保護主出力の割合は極めて低い．実際，デコーダの主出力 269 個中，影響を受ける出力は 1〜3 個に留まる．さらに論理ゲート 524 個中 150 個（約 29%）を LUT に置換しても，主出力 269 個中 14 個（約 5%）しか保護できず，その時点で面積オーバヘッドは約 25%，遅延オーバヘッドは約 30%に達する．このように，保護主出力数が犠牲になっていることが明らかである．提案手法である HCLL-Lock は，同程度の面積オーバヘッドで比較すると最も高い保護主出力割合を示す．NB2 戦略のため，LUT-Lock と同様にある閾値で効果が鈍化する傾向も読み取れるが，飽和に至るまでの立ち上がり（コスト対効果）は HCLL-Lock が優位である．

遅延オーバヘッドで整理しても傾向は概ね同様である．LUT-Lock はオーバヘッド増に対して保護主出力数は早期に頭打ちとなり，FIFOL-Lock は保護主出力割合が極端に小さいのに対して，HCLL-Lock は同程度のオーバヘッドでより多くの主出力を保護できている．また，LUT-Lock と HCLL-Lock はいずれも NB2 の影響で飽和が生じている．

総じて，面積・遅延いずれの指標でも，HCLL-Lock は同等のオーバヘッドで最大の保護主出力割合を達成する．FIFOL-Lock は高い耐性を狙う設計方針ゆえに保護主出力割合が著しく低く，LUT-Lock は保護主出力割合向上のために置換を増やすとオーバヘッドが急増する．HCLL-Lock は，保護主出力割合とコストの両面で実用的なバランスを示すことが確認できる．

4.3 SAT 攻撃耐性

図 8，**図 9** は各種オーバヘッドと SAT 攻撃時間の関係を示している．横軸は Ibex 全体での面積オーバヘッド・遅延オーバヘッド（%），縦軸は SAT 攻撃時間（秒）を対数スケールで表示している．

LUT-Lock は，面積オーバヘッドの増加に伴って SAT

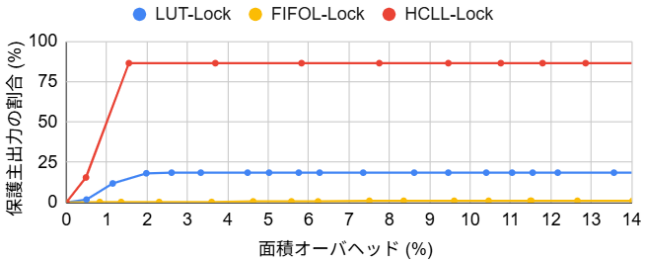

図 6 面積オーバヘッドあたりの保護主出力の割合

Fig. 6 Percentage of primary output protected per area overhead

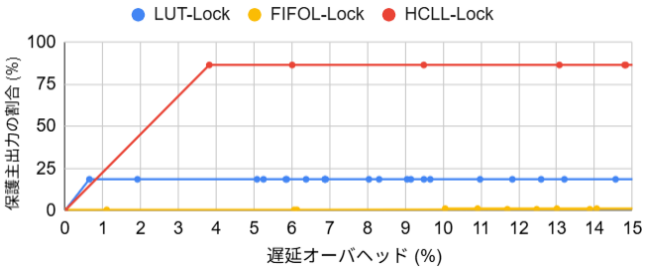

図 7 遅延オーバヘッドあたりの保護主出力の割合

Fig. 7 Percentage of primary output protected per delay overhead

攻撃時間は増加するが，単独で大幅な耐性を得るには置換数の増加すなわちオーバヘッドの増大が避けられない．FIFOL-Lock は，同程度の面積オーバヘッドで比較した場合，3 つの手法中で最も長い攻撃時間を示している．ただし，このメリットは LUT 置換の影響を受ける主出力を極端に減らす設計方針に依存しており，正しい鍵を入力しなくても多くの入力パターンで元回路と同一出力となり得るため，IP 保護としての実用性には疑問が残る．提案手法である HCLL-Lock は，FIFOL-Lock に次ぐ高い攻撃時間を示しつつ，保護主出力割合を大きく確保している．したがって，面積オーバヘッド当たりの「実効的な」耐性という観点では，最もバランスが良い．

遅延オーバヘッドで評価しても，全体の傾向は同様であり，オーバヘッドの増加に対して SAT 攻撃時間は指数関数的に増加する．同程度の遅延オーバヘッドでは，耐性は FIFOL-Lock ＞ HCLL-Lock ＞ LUT-Lock の順に高い．ただし FIFOL-Lock の優位は保護主出力割合を犠牲にした結果であり，HCLL-Lock が実用性を伴った耐性の面で優位と評価できる．

総じて，SAT 攻撃時間は面積・遅延いずれのオーバヘッドに対しても指数関数的に増加するが，HCLL-Lock は保護主出力割合と攻撃耐性を同時に高水準で満たす．LUT-Lock は耐性向上に比例してオーバヘッドが嵩み，FIFOL-Lock は高耐性の代償として保護主出力数が不足する．HCLL-Lock は，実運用を想定したトレードオフの観点で最も優れたコストパフォーマンスを示している．

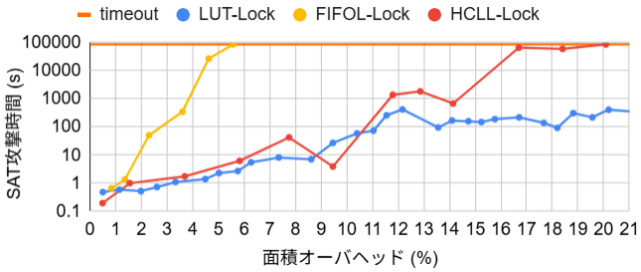

図 8 面積オーバヘッドあたりの SAT 攻撃時間

Fig. 8 SAT attack execution time per area overhead

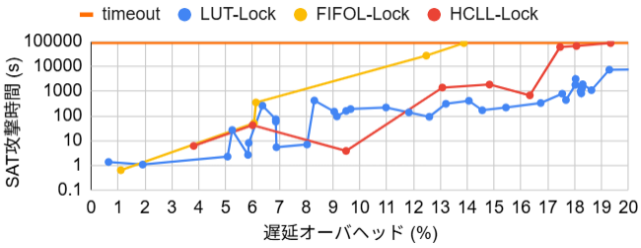

図 9 遅延オーバヘッドあたりの SAT 攻撃時間

Fig. 9 SAT attack execution time per delay overhead

4.4 総評

実験結果から，提案手法 HCLL-Lock は，面積・遅延のオーバヘッドを抑えながら，多数の主出力を効果的に保護できるという点で，既存手法より優れていることを明らかにした．

なお，LUT 置換数が同じ条件で比較すると，遅延・面積オーバヘッドは HCLL-Lock および FIFOL-Lock の方が LUT-Lock より大きくなる傾向が見られる．これは，HCLL-Lock および FIFOL-Lock がループ構造のために複数の MUX（2 入力）を追加しているためであると考えられる．しかしながら，これまでの評価から明らかなように，同程度のオーバヘッドで比較した場合，HCLL-Lock と FIFOL-Lock は LUT-Lock よりも高いセキュリティ効果を持つ．特に HCLL-Lock は，主出力の保護と SAT 攻撃耐性の両立という観点において最も高いコストパフォーマンスを有しており，実用的かつ堅牢な IP 保護手法として有用であることが確認された．

5. 結論

本研究では，SAT 攻撃をはじめとする機能攻撃に対して高い耐性を持ちつつ，回路オーバヘッドを抑制するハードウェア IP 保護手法 HCLL-Lock を提案した．本手法は，回路内の一部の論理ゲートを Look-Up Table（LUT）に置換することに加え，複雑なループ構造を導入することで，攻撃者による鍵探索を困難にするとともに，できる限り多くの主出力を保護することを目指している．

評価実験では，RISC-V プロセッサ Ibex のデコーダモジュールを対象として，従来手法である LUT-Lock および FIFOL-Lock との比較を行った．その結果，HCLL-Lock

は，同等または少ないオーバヘッドでより多くの主出力を保護できること，および SAT 攻撃に対しても高い耐性を維持できることが確認された．特に，多くの主出力の保護と SAT 攻撃耐性という 2 つの重要な指標の両立を実現している点は，本手法の有効性を示す重要な成果である．

今後の課題としては，ループ構造の挿入のアルゴリズムの改善，構造解析や近似攻撃といった他の攻撃手法に対する耐性の評価，遅延オーバヘッドをより小さく抑えるような LUT 置換アルゴリズムの考案などが挙げられる．

謝辞

本研究は，JSPS 科研費 JP25K03091，および，JST CREST S5 基盤 JPPMJCR23M2 の支援を受けたものである．また，東京大学 VDEC 活動を通して，日本シノプシス合同会社の協力で行われたものである．

参考文献

[1] Roy, Jarrod A and Koushanfar, Farinaz and Markov, Igor L. EPIC: Ending piracy of integrated circuits. In *Proceedings of the conference on Design, automation and test in Europe*, pp. 1069–1074, 2008.

[2] Kamali, Hadi Mardani and Azar, Kimia Zamiri and Farahmandi, Farimah and Tehranipoor, Mark. Advances in logic locking: Past, present, and prospects. *Cryptology ePrint Archive*, 2022.

[3] Subramanyan, Pramod and Ray, Sayak and Malik, Sharad. Evaluating the security of logic encryption algorithms. In *2015 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 137–143. IEEE, 2015.

[4] Kamali, Hadi Mardani and Azar, Kimia Zamiri and Gaj, Kris and Homayoun, Houman and Sasan, Avesta. Lut-lock: A novel lut-based logic obfuscation for fpga-bitstream and asic-hardware protection. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 405–410. IEEE, 2018.

[5] Muhammad Yasin, Bodhisatwa Mazumdar, Jeyavijayan JV Rajendran, and Ozgur Sinanoglu. Sarlock: Sat attack resistant logic locking. In *2016 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 236–241. IEEE, 2016.

[6] Yang Xie and Ankur Srivastava. Anti-sat: Mitigating sat attack on logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 38, No. 2, pp. 199–207, 2018.

[7] Meng Li, Kaveh Shamsi, Travis Meade, Zheng Zhao, Bei Yu, Yier Jin, and David Z Pan. Provably secure camouflaging strategy for ic protection. *IEEE transactions on computer-aided design of integrated circuits and systems*, Vol. 38, No. 8, pp. 1399–1412, 2017.

[8] Muhammad Yasin, Abhrajit Sengupta, Mohammed Thari Nabeel, Mohammed Ashraf, Jeyavijayan Rajendran, and Ozgur Sinanoglu. Provably-secure logic locking: From theory to practice. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1601–1618, 2017.

[9] Muhammad Yasin and Jeyavijayan Rajendran and Ozgur Sinanoglu. *Trustworthy Hardware Design: Combinational Logic Locking Techniques*. Analog Circuits and Signal Processing. Springer Nature Switzerland AG, Cham, Switzerland, 2020.

[10] Shamsi, Kaveh and Li, Meng and Meade, Travis and Zhao, Zheng and Pan, David Z and Jin, Yier. Cyclic obfuscation for creating SAT-unresolvable circuits. In *Proceedings of the Great Lakes Symposium on VLSI 2017*, pp. 173–178, 2017.

[11] Hadi Mardani Kamali, Kimia Zamiri Azar, Houman Homayoun, and Avesta Sasan. Full-lock: Hard distributions of sat instances for obfuscating circuits using fully configurable logic and routing blocks. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pp. 1–6, 2019.

[12] Muhammad Yasin, Jeyavijayan JV Rajendran, Ozgur Sinanoglu, and Ramesh Karri. On improving the security of logic locking. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 35, No. 9, pp. 1411–1424, 2015.

[13] Shamsi, Kaveh and Li, Meng and Meade, Travis and Zhao, Zheng and Pan, David Z and Jin, Yier. AppSAT: Approximately deobfuscating integrated circuits. In *2017 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp. 95–100. IEEE, 2017.

[14] Shen, Yuanqi and Zhou, Hai. Double DIP: Re-evaluating security of logic encryption algorithms. In *Proceedings of the Great Lakes Symposium on VLSI 2017*, pp. 179–184, 2017.

[15] Alex Baumgarten, Akhilesh Tyagi, and Joseph Zambreno. Preventing ic piracy using reconfigurable logic barriers. *IEEE design & Test of computers*, Vol. 27, No. 1, pp. 66–75, 2010.

[16] Gaurav Kolhe, Sai Manoj PD, Setareh Rafatirad, Hamid Mahmoodi, Avesta Sasan, and Houman Homayoun. On custom lut-based obfuscation. In *Proceedings of the 2019 Great Lakes Symposium on VLSI*, pp. 477–482, 2019.

[17] Zhou, Hai and Jiang, Ruifeng and Kong, Shuyu. CycSAT: SAT-based attack on cyclic logic encryptions. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 49–56. IEEE, 2017.

[18] Shamsi, Kaveh and Pan, David Z and Jin, Yier. IcySAT: Improved SAT-based attacks on cyclic locked circuits. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7. IEEE, 2019.

[19] 小棚木 凛太朗, 一岡 知佑, 楊 明宇, 原 祐子. eASIC を用いたハードウェア IP 保護手法：RISC-V への適用事例. 2025 年 暗号と情報セキュリティシンポジウム, 2025.

[20] Ibex. https://github.com/lowRISC/ibex.

[21] Mayler Martins, Jody Maick Matos, Renato P Ribas, André Reis, Guilherme Schlinker, Lucio Rech, and Jens Michelsen. Open cell library in 15nm freepdk technology. In *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, pp. 171–178, 2015.

[22] Netlist Encryption and Obfuscation Suite. https://bitbucket.org/kavehshm/neos/src/master/.