ISSN 2186-5647

コントローラ拡大と鍵系列を用いたIPコアの論理暗号化法

日大生産工(院)　○橋立 英実　　日大生産工　細川 利典
京産大　　吉村 正義

1. まえがき

近年の半導体微細化技術の進歩に伴って，大規模集積回路(Very Large Scale Integrated circuits:VLSI)は，社会情報基盤技術や，機密情報に関する情報を保持するシステムにおいて使用されている．このため，VLSIの信頼性が社会に与える影響は増大している．また，VLSIは高機能化や高集積化のため、回路規模が増大している．これにより VLSI 設計には人件費や設計/製造コストなどの開発コストが増加している．

この課題を解決するために，VLSIの設計において，一部の機能は，外部から IP コア(Intellectual Property core)を購入している．具体的には，自社ですべてのVLSI上の機能ブロックの設計を行うのではなく，IPベンダからIPコアを購入し，IP コアを用いて VLSI の一部の機能ブロックを実現することにより，設計コストの削減を図っている．しかしながら，その結果としてIPコアの著作権侵害が懸念されている．具体例として，IPコアの不正な再利用，マスクの窃盗や，IPベンダが許可していない過剰な数のVLSIの生産などが挙げられる[2]．最近の研究では，VLSIの著作権侵害は電子産業や防御産業において主要な攻撃となっていることが報告されている[1].

これに対しEPIC(Ending Piracy of Integrated Circuits)[2]は，VLSIを論理暗号化することにより VLSI の過剰生産を防ぐことを提案している[2]. VLSIを機能動作させるためには，IPベンダのみが生成することができる入力鍵が必要となる．EPICの手法は(1)チップID生成，(2)組合せ回路部の論理暗号化，(3)公開鍵暗号方式による鍵生成に基づいている．この手法によってVLSIの過剰生産に耐性のある VLSI を設計する．EPIC は，論理暗号化のオーバーヘッドによるVLSI の遅延や消費電力の増加が非常に小さく，その検証にために標準設計フローやテスト方式を変更する必要がない．そのため，EPICは過剰生産に対して頑強な防御策であることが示された．

EPICはXORゲートやXNORゲートなどを組合せ回路部に挿入することで論理暗号化を行っている．XORゲートやXNORゲートを挿入することで誤った鍵を印加したときに，誤った値が外部出力まで伝搬する可能性がある．

現在，前述した手法を基に故障辞書を用いて論理暗号化箇所を決定する FLE(Faultsimulation Logic Encryptin)[3]や，回路構造に着目して論理暗号化箇所を決定する SLE(Structure Logic Encryption)[4]など論理暗号化手法が提案されている．しかしながら，SAT 攻撃[5]によって鍵を特定するという手法が提案されている[5]．この攻撃手法は効率良く鍵探索空間を狭めて，最終的に現実的な時間で，正しい鍵を特定するという手法を用いている．SAT攻撃に対し，文献[6][7]では，鍵探索空間を1個ずつ狭めさせる論理暗号化手法が提案されている．

本論文では，前述した論理暗号化法[6][7]とは異なり，コントローラ拡大と鍵系列を用いることで，SAT 攻撃時の時間展開数を増加させることで，現実時間内で鍵系列の探索が困難な論理暗号化手法を提案する．

第2章ではIPコア，鍵の種類の定義を行う．第 3 章では入力鍵を生成するための論理暗号化の鍵生成フロー，論理暗号化を組込んだ設計/製造フロー及び入力鍵を印加したときの復号化フローを説明する．第4章では論理暗号化回路の攻撃手法の概要とSAT攻撃[5]の説明を行う．第5章では，SAT 攻撃の対策手法として鍵系列とコントローラ拡大を用いた論理暗号化手法を提案する．第6章では，実験の評価回路と評価基準を述べ，実験結果を示し，結果に対する考察を述べ

A Logic Encryption Method of IP Core Using Controller Augmentation and Key Sequence

Hidemi HASHIDATE, Toshinori HOSOKAWA and Masayoshi YOSHIMURA

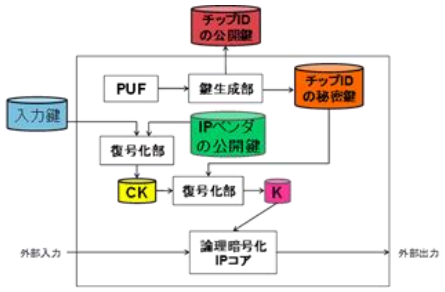

図 1 公開鍵暗号方式を用いた論理暗号化回路

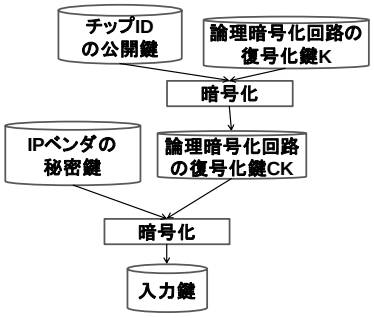

図 2 入力鍵生成フロー

る．第 7 章では結論を述べたあと，今後の課題を記す．

2. 諸定義

2.1. VLSIの設計/製造フロー

VLSIの設計/製造フローには，VLSI設計者，IPベンダ，消費者が関わっている．鍵の種類は，チップIDの公開鍵と秘密鍵，マスターの公開鍵と秘密鍵，論理暗号化回路の復号化鍵K，IPコアの復号化鍵CK, IPコアの入力鍵の7種類の鍵が存在する．チップIDの秘密鍵と公開鍵はチップごとに異なる性質を持ち，秘密鍵は誰もその値を知らない．マスターの公開鍵と秘密鍵はIPコアに対してそれぞれ1個だけ存在し，秘密鍵はIPベンダしか知らない．論理暗号化回路の復号鍵Kは論理暗号化された回路を正常に動作させ，IPコアによって一意に決まる性質がある．IPコアの復号化鍵CKはチップごとに異なる性質がある．IPコアの入力鍵はIPベンダが設計した論理暗号化IPコアを含んだ設計データを正常に動作させ，チップごとに異なる性質がある．チップIDの公開鍵と秘密鍵は，IPコア内にある物理的にクローン化が不可能な機能PUF(Physically Unclonable Function)で生成され，チップIDの公開鍵を外部に出力し，チップIDの秘密鍵はIPコア回路に追加される．IPベンダの公開鍵と秘密鍵はIPベンダが生成し，IPベンダの公開鍵はIPコアに追加され，マスターの秘密鍵はIPベンダ以外に知られることがないようにする．論理暗号化回路の復号化鍵は論理暗号化したときに得られる鍵のビット列である．IPコアの入力鍵CKは，チップIDの公開鍵とIPベンダの秘密鍵で公開鍵暗号方式を用いて暗号化し，生成される．

2.2. 過剰生産

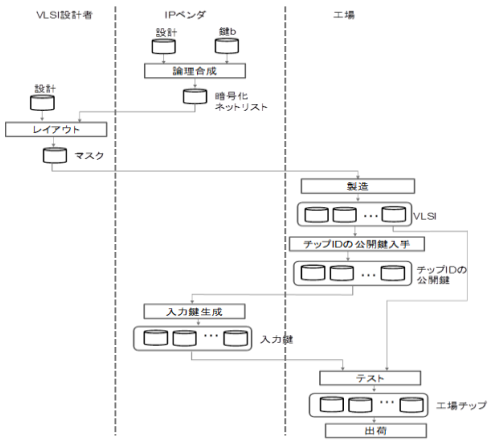

図 3 設計/製造フロー

2.3. IPベンダが設計会社にIPコアを販売するときに，そのIPコアを搭載するVLSIの生産数を決めて契約する．過剰生産とは契約した生産数を越えてIPコアとしてVLSIに搭載し生産することである．暗号化されていない回路をIPコアとして購入した場合，際限なくVLSIに搭載することができ，IPベンダの許可なく，過剰にVLSIを製造し販売することができる．

2.4. 論理暗号化と公開鍵暗号方式を用いた過剰生産防御法

本節では，文献[2]で提案されている論理暗号化と公開鍵暗号を用いた過剰生産に対する防御法を述べる．図1にIPベンダが提供する論理暗号化IPコアを含んだ設計データを示す．Kは論理暗号化回路の復号鍵を示し，CKとは復号鍵Kをチップの公開鍵で暗号化した鍵である．PUFと鍵生成部によって，チップIDの公開鍵と秘密鍵が生成され，その公開鍵のみ外部に出力される．

3. 論理暗号化の設計/製造フロー鍵生成フロー

本章では，論理暗号化の鍵生成フロー，文献[2]の設計/製造フローについて説明する．図2に論理暗号化の鍵生成フローを示す．図2では、まず始めに論理暗号化された回路の復号化鍵Kをチップの公開鍵で暗号化し，IPコアの復号化鍵CKを生成する．暗号化されたIPコアの復号化鍵CKをIPベンダの秘密鍵で暗号化する．これが最終的に回路に印加する入力鍵となる．

図3に文献[2]の設計/製造フローを示す．IPベンダはIPコアを論理暗号化する．そのとき，論理暗号化回路の復号化鍵Kを生成する．またPUFと鍵生成部，マスター鍵の公開鍵，二つの復号化部をIPコアに追加する．VLSI設計者はIPベンダが論理暗号化した回路を受け取り，マスクを生成する．製造されたチップは鍵生成部を用いてチップIDとその秘密鍵と公開鍵を生成する．チップIDの秘密鍵は回路に記憶され，公開鍵は外部に出力される．工場はIPベンダにチップ鍵の公開鍵を送信

する. 工場は製造テストを行い製造した回路を出荷する.

4. 論理暗号化回路に対する攻撃

本章では, 論理暗号化された回路に対するブルートフォース攻撃手法[2]を説明する. 初めに論理暗号化された回路に対する攻撃方法の課題点を説明した後, 課題点を解決するSAT攻撃[5]を説明する.

4.1. 問題設定

本論文では, 攻撃者は以下の物を所持していると仮定する.

1. 論理暗号化済みネットリストC
2. 活性化済みVLSI

攻撃者は, IPベンダから提供される論理暗号化済みネットリストを所持することができる. また, 市場に出回っている活性化済み(暗号化が解除されている)VLSIを入手することができる.

4.2. ブルートフォース攻撃の問題点

論理暗号化回路において, p個の外部入力を持つ入力ベクトル$(X_1, X_2, \cdots X_p)$と, 対応するq個の外部出力を持つ出力ベクトル$(Y_1, Y_2, \cdots Y_q)$が与えられたとき, 出力ベクトルと一致する鍵を探索することは容易である. その方法は, $C(\wedge_{i=1}^{p} \vec{X}_i, \vec{K}, \wedge_{j=1}^{q} \vec{Y}_j)$という式をSATSolverに入力し, 式を満足する鍵ベクトル$\vec{K}$を探索するものである. しかしながら, 新たな入力ベクトル$(X'_1, X'_2 \cdots X'_p)$と対応する出力ベクトル$(Y'_1, Y'_2, \cdots Y'_q)$が与えられたとき, $C(\wedge_{i=1}^{p} \vec{X}_i, \vec{K}, \wedge_{j=1}^{q} \vec{Y}_j)$を満足する鍵ベクトル$\vec{K}$が, $C(\wedge_{i=1}^{p} \vec{X}'_i, \vec{K}, \wedge_{j=1}^{q} \vec{Y}'_j)$を満足するという保証はない. SATSolverはこれまでの観測値と一致する鍵を探索することが可能であるが, これを検証するためには2^M個(M:鍵ビット数)の鍵で入出力パターンを確認する必要がある. そのため, 前述した手法で正しい鍵を探索する時間は指数関数的に増加する. これらの課題点を解決する手法が文献[5]で提案されている.

4.3. SAT攻撃

本節では, 文献[5]で提案されているSAT攻撃を説明する. 文献[5]は2個の洞察を通じて課題点を解決している.

1個目の洞察は, 鍵ベクトルを個別に考えるのではなく, 鍵ベクトルの等価クラスを考える. 任意の入力ベクトル$\vec{X}_i$についてのみ, 2個の暗号化回路が両方の鍵ベクトル$\vec{K}_1$と$\vec{K}_2$に対して, 同じ出力ベクトル$\vec{Y}_j$を生成する場合, $\vec{K}_1$と$\vec{K}_2$を等価クラスであると定義する. $C(\vec{X}_i, \overrightarrow{K_1}, \vec{Y}_j) \wedge C(\vec{X}_i, \overrightarrow{K_2}, \vec{Y}_j)$という式は, 正しい鍵ベクトルを探索する代わりに, 任意の入力ベクトルに対して正しい出力ベクトルを生成する鍵の等価クラスを探索する. 誤った鍵ベクトルの等価クラスを鍵探索空間から削除するために, 少なくとも1個の入力ベクトルに対して誤った出力ベクトルを生成する等価クラスを繰り返し除外する.

2個目の洞察は, 2個の暗号化回路に対して, 鍵ベクトル$\overrightarrow{K_1}$と$\overrightarrow{K_2}$, がそれぞれ設定されており, 1個の入力ベクトルに対して異なる出力ベクトル$\overrightarrow{Y_1^d}$と$\overrightarrow{Y_2^d}$を出力する場合, その入力ベクトルを識別入力ベクトル$\overrightarrow{X_d}$と定義する. 正確には, $C(\overrightarrow{X_d}, \overrightarrow{K_1}, \overrightarrow{Y_1^d}) \wedge C(\overrightarrow{X_d}, \overrightarrow{K_2}, \overrightarrow{Y_2^d}) \wedge (\overrightarrow{Y_1^d} \neq \overrightarrow{Y_2^d})$ならば, $\overrightarrow{X_d}$は$\overrightarrow{K_1}$と$\overrightarrow{K_2}$の識別入力ベクトルである. 識別入力ベクトル$\overrightarrow{X_d}$が見つかった場合, $\overrightarrow{X_d}$に対して活性化済みVLSIの出力ベクトルを調べ, これを使用して鍵ベクトル$\overrightarrow{K_1}$と$\overrightarrow{K_2}$の一方または両方を正しい鍵の等価クラスの鍵にすることである.

5. コントローラ拡大を用いた論理暗号化法

本章では, 前章のSAT攻撃に耐性を持つ論理暗号化手法を提案する. 本提案手法は, コントローラ拡大と鍵系列を用いることで, SAT攻撃時に論理暗号化回路の時間展開が必要であり, 現実時間内で正しい鍵系列の探索が困難な論理暗号化手法である. 5.1節において, コントローラ拡大の論理暗号化に関して図を用いて説明する. 5.2節では, 鍵を入力する仕組みや, 入力された鍵がいつ認証されるかを説明する.

5.1. コントローラ拡大

本節では, コントローラ拡大と鍵系列を用いて論理暗号化手法を説明する. 本手法はRTL(Resister Transfer Level)での論理暗号化手法であり, RTL 回路はコントローラとデータパスから構成されていると仮定する. 図 4 に RTL コントローラを示す. 次に, 図 5 に図 4 のコントローラを拡大することにより論理暗号化した図を示す. 図 4 と図 5 は共に, 状態数が 10 である. 図 4 では, state1 から state4 までの状態遷移のコントローラ拡大を行った. コントローラ拡大を行って追加された状態遷移のことを" 暗号化状態遷移"と定義し, コントローラ拡大前の状態遷移のことを" 状態遷移"と定義する. 暗号化状態遷移に遷移するか否かは, コントローラに入

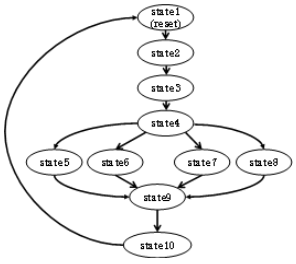

図 4 コントローラ

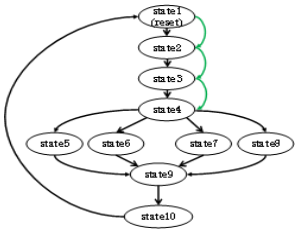

図 5 論理暗号化されたコントローラ

力された鍵によって決定される．コントローラに正しい鍵が入力された場合は，状態遷移を行う．もし，コントローラに入力された鍵が誤っていた場合，暗号化状態遷移を行う．暗号化状態遷移は誤った制御信号をデータパスに供給するので，データパス内の動作は誤った動作を行う．

5.2. 鍵認証部

本節では，暗号化済みコントローラに鍵を入力する仕組みを説明する．5.2.1 節では，2.3 節で説明した公開鍵暗号を用いた順序回路の論理暗号化について説明する．5.2.2 節では，コントローラの鍵系列を処理するための仕組みを説明する．

5.2.1. 公開鍵暗号を用いた順序回路の論理暗号化

公開鍵暗号を用いた論理暗号化手法を説明する．文献[2]では，過剰生産に対する防御法として IP ベンダとチップ ID を用いる公開鍵暗号が提案されている．本提案手法もこれに倣い，コントローラの論理暗号化の鍵系列についても同じ方法を用いる．

図 6 にコントローラの論理暗号化の鍵入力の図を示す．図 6 で示した K 以外のそれぞれの鍵は 2.3 節で説明した役割と同じなので省略する．2.3 節で説明した K は XOR ゲートなどの 1 入力が 1 ビットであるような錠ゲートの入力の集合に対してビット長が 1 の鍵入力で表されていたが，コントローラを論理暗号化した SK は鍵系列として複数のビット長で表される．図 7 と図 8 に論理暗号化前の RTL 回路と論理暗号化後の RTL 回路を示す． 論理暗号化されたコントローラに入力される鍵は図 6 で入力される鍵系列 SK である．

6. 実験結果

実験結果は,EPIC[2]で論理暗号化した回路と本提案手法で論理暗号化した回路とSAT攻撃を行い鍵探索時間の結果を出す予定である．

7. まとめ

本稿では，SAT攻撃に耐性のあるコントローラ拡大と鍵系列を用いた論理暗号化手法を提案した．今後の課題として，様々な論理暗号化手法と

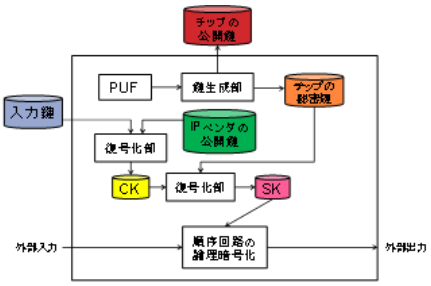

図 6 順序回路の公開鍵暗号

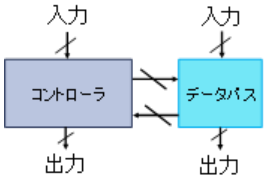

図 7 RTL 回路

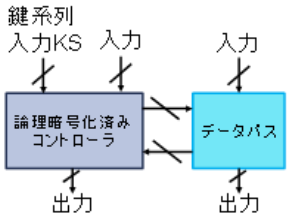

図 8 論理暗号化済み RTL 回路

本提案手法を比較し，SAT攻撃を評価した結果を示すことが挙げられる．

参 考 文 献

[1] W. Schneider, F. Chairman, “Defense Science Board Task Force On High Performance Microchip Supply,” the Defense Science Board https://www.acq.osd.mil/dsb/reports/2000s/ADA435563.pdf, 2005.

[2] J. Roy, F. Koushanfar, and I. Markov, “EPIC: Ending Piracy of Integrated Circuits,” Design, Automation and Test in Europe, pp.1069-1074, 2008.

[3] J. Rajendran, Y. Pino and O. Sinanoglu, “Logic Encryption: A Fault Analysis Perspective,” Design, Automation and Test in Europe Conference and Exhibition, pp.953-958, 2012.

[4] J. Rajendran, Y. Pino, O. Sinanoglu, R. Karri, “Security Analysis of Logic Obfuscation,” Design Automation Conference, pp.83-89, 2012.

[5] P. Subramanyan, S. Ray, S. Malik, “Evaluating the Security of Logic Encryption Algorithms,” Hardware Oriented Security and Trust, pp.137-143, 2015.

[6] M. Yasin , B. Mazumdar, J. Rajendran, o. Sinanoglu, “SARLock: SAT Attack Resistant Logic Locking,” Hardware Oriented and Trust, pp236-241, 2016.

[7] M. Zaman, A. Sengupta, D. Liu, O. Sinanoglu “Towards Provably-Secure Performance Locking,” Design, Automation and Test in Europe ,pp1592-1597, 2018.