ロードされた構成データを修飾して使用する安全性強化型再構成可能ハードウェア

赤井 健一郎[†] 松本 勉[†] 河野 隆二[††]

FPGA（Field Programmable Gate Array）などの再構成可能ハードウェアは，デバイス内部で構成する回路を定義した構成データをロードすることで，内部の回路構成を変更することができる．ソフトウェア無線のように構成データがネットワーク上でやりとりされるような環境では，不正な利用者に構成データが渡り，解析・改変されることが考えられる．そこで我々は，解析行為が難しくなるような構成データの保護方式の基本アイディアを提案する．基本アイディアでは，あらかじめ再構成可能ハードウェア内部に秘密の固定回路を作っておく．構成データ作成者は，構成データを秘密の固定回路に応じて変換して，配布する．もし，不正な利用者が変換された構成データを入手したとしても，保護対象の回路とは等価ではないため，解析が困難になると期待される．本稿では，基本アイディアを FPGA 上で実装する方法について検討し，その安全性について論じる．

Security Enhanced Reconfigurable Hardware Working with Modified Version of Loaded Configuration Data

Kenichiro Akai,[†] Tsutomu Matsumoto[†] and Ryuji Kohno[††]

Reconfigurable Devices such as field programmable gate arrays (FPGAs) can change internal circuit by loading configuration data that defines the circuit. In the environment where configuration data can be exchanged through the network such as Software Defined Radio, there is possibility that an illegal user gets the configuration data and he analyzes and tampers with the data. Then we propose the scheme that makes analyzing configuration data difficult. In this scheme, configuration data is modified with secret circuit inside a reconfiguration device. Original function is implemented by combining modified configuration data with the secret circuit inside the reconfiguration device. If an illegal user gets the modified configuration data, it is difficult for him to analyze it because the modified configuration data is not equal to the original configuration data. In this paper, we discuss the difficulty in analyzing our scheme when it is implemented on FPGAs.

1. はじめに

FPGA（Field Programmable Gate Array）などの再構成可能ハードウェアは，デバイス内部で構成する回路を定義した構成データをロードすることで，内部の回路構成を変更することができる．再構成可能ハードウェアの構成データをネットワーク上でやりとりすることも今後増えてくると考えられる．このような例として，ソフトウェア無線があげられる．ソフトウェア無線は，携帯端末に FPGA を組み込んでおき，通信方式を記述した構成データをネットワーク上で配信し，FPGA にロードすることで，通信方式を柔軟に変更できる通信システムである．しかし，構成データがネットワーク上でやりとりされるような環境では，不正な利用者（以下，不正者）に構成データが渡り，解析・改変が行われて，提供するサービスの不正利用や妨害が行われる可能性がある．したがって，不正者が構成データの解析・改変を行うことを難しくする必要があると考えられる．

再構成可能ハードウェアの構成データ保護方式として，耐タンパーハードウェア内部に再構成可能ハードウェアと暗号の復号関数および復号鍵を内蔵させておき，構成データを暗号化して保護する方式が提案されている[2)~4)]．また，構成データの冗長性を利用して，構成データにフィンガープリンティングを施す方式も提案されている[5)]．我々は，これらとは異なるアプロー

† 横浜国立大学大学院環境情報研究院
Graduate School of Environment and Information Sciences, Yokohama National University
†† 横浜国立大学大学院工学研究院
Graduate School of Engineering, Yokohama National University

☆ 本稿は，文献 1) の内容を含み，さらに充実させたものである．

チで，構成データの解析行為を難しくするような構成データの保護方式を提案する．提案方式では，あらかじめ再構成可能ハードウェア内部に秘密の固定回路を作っておく．構成データ作成者は，構成データを秘密の固定回路に応じて変換して，配布する．もし，不正者が変換された構成データを入手したとしても，保護対象の回路とは等価ではないため，解析が困難になると考えられる．

本稿では，2章で基本アイディアを説明し，3章で基本アイディアのFPGA上での実装に関して述べ，4章でFPGAで実装した場合の安全性に関して検討する．そして，5章でまとめと今後の課題について述べる．

2. 基本アイディア

FPGAなどの再構成可能ハードウェアは，内部で構成する論理回路を定義した構成データをロードすることで，任意の論理回路を構成することができる．ある関数 f を実現する構成データCと，Cを構成する論理ゲートの1カ所を取り除いた構成データC'を考える．Cを再現するためには，C'の回路が欠けた部分に入りうるゲートの数および種類が既知であっても，欠落した部分に入りうる論理ゲートに関してすべての可能性を確かめる必要がある．また，ゲート間の結線も欠けている場合には，さらに複雑になると考えられる．

そこで，構成データ作成者は，再構成可能ハードウェア内部にあらかじめ秘密の固定回路 s を構成して出荷する．構成データ作成者は，関数 f を実現する論理回路の構成データCを作成し，再構成可能ハードウェアに固有の秘密の固定回路 s をもとにCを C^{-s} に変換して配信する．本来実現したかった関数 f の機能は，ハードウェア内部の秘密の固定回路 s と C^{-s} を組み合わせることで実現される（**図1**）．もし不正者が C^{-s} を入手したとしても，秘密の固定回路 s を知らずに解析を行うことは困難であると考えられる．また，不正者が通常の再構成可能ハードウェアや別の固定回路 t を持つ再構成可能ハードウェアに，C^{-s} をロードしても，関数 f の機能は得られない．

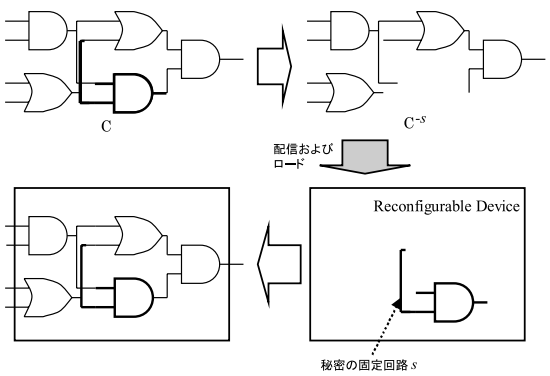

図1 基本アイディア
Fig. 1 The basic idea.

構成データ作成者が C^{-s} に変換する際に，固定回路 s とCの組合せによっては，Cから s を取り除くことができない場合がある．そこで，s をうまく利用するようにCを等価変換してから C^{-s} を作成する．このようにすると，固定回路 s でCを変換した C^{-s} と s とは異なる t でCを変換した C^{-t} では，違いが生じる．したがって，C^{-s} と C^{-t} を単純に比較しただけでは，s および t の情報が漏れにくくなると考えられる．また，Cから s を取り除いた後，s に相当する部分に別の論理回路を挿入して C^{-s} とすることにより，さらに s に関する情報が漏れにくくなると考えられる．

3. FPGAにおける実現法

再構成可能ハードウェアには，PLD（Programmable Logic Device）やFPGAが存在する．どちらに対しても，2章で述べた基本アイディアの適用が可能であると予想できる．しかし，そう主張するためには詳細な検証が必要である．本稿では，FPGAを用いて基本アイディアを適用した場合について述べ，PLDに関しては今後の課題とする．

3.1 FPGAの構造

FPGAの構造は小規模論理ブロックを規則的に配列し，その周辺にプログラム可能な配線を配置したものである．FPGAはベンダごとに様々な種類のものが提供されている．たとえば，FPGAに回路を構成する方法としては，アンチフューズを用いる方法や，SRAM（Static Random Access Memory），EPROM（Erasable and Programmable Read Only Memory），EEPROM（Erectrically Erasable Read Only Memory）などの記憶装置を用いて，記憶されている回路情報に応じてパストランジスタなどを制御し，回路を構成する方法がある．また，論理ブロックの形式にはルックアップテーブル（LUT: Look Up Table）タイプ，マルチプレクサロジックタイプ，ゲートアレイタイプなどがある[6]．このように，さまざまなタイプのFPGAがあるため，本稿で用いるFPGAは単純化して，モデル化したものを用いて説明する．本稿で用いるFPGAのタイプは，SRAMを用いて回路を構成し，論理ブロックとしてLUTタイプを用いたもので論じる．このタイプのFPGAとしては，Xilinx社のVirtexシリーズ[3),4),7),8)]やXC4000シリーズ[9)]があ

げられる．しかし，これらの製品が必ずしも本稿で用いる FPGA のモデルに当てはまるとは限らない．

本稿で用いる FPGA のモデルを**図 2** に示す．この FPGA のモデルでは，プログラム可能なブロックがマトリクス上に配置され，それらをつなぐ導線を配置した構造をしている．以下，それぞれのブロックについて説明する．

IOB（I/O Block）

IOB は，FPGA 内部で構成されている回路と外部とのインタフェースとなるブロックである．

CLB（Configuration Logic Block）

本稿で用いる CLB のモデルは，k ビット入力の LUT（Look Up Table）と D タイプフリップフロップ（D-FF），およびマルチプレクサ（MUX）をそれぞれ 1 つずつ持ち，小規模な論理回路を作ることができる（**図 3**）．LUT は k ビット入力 1 ビット出力の真理値表を実現する．LUT は付属する 2^k ビットの SRAM に値を書き込むことで任意の出力を実現することができる．MUX は，D-FF を使用するかどうか制御する．MUX にも 1 ビットの SRAM が付属しており，これに値を書き込むことで制御する．

CLB の入力 $(\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_k)$ と，その周辺に配置された導線との接続は，パストランジスタにより制御することができる．パストランジスタには 1 ビットの SRAM が付属し，これに値を書き込むことで接続の制御を行う．

SM（Switch Matrix）

本稿で用いる SM のモデルは，CLB の周りに配置された水平方向と垂直方向の導線間の接続を制御する（**図 4**）．水平方向および垂直方向からそれぞれ m 本の導線があり，水平方向の導線を $\mathrm{H}_1, \mathrm{H}_2, \cdots, \mathrm{H}_m$，垂直方向の導線を $\mathrm{V}_1, \mathrm{V}_2, \cdots, \mathrm{V}_m$ とすると，導線の交点のうち $\mathrm{H}_1\mathrm{V}_1, \mathrm{H}_2\mathrm{V}_2, \cdots, \mathrm{H}_m\mathrm{V}_m$ に接続を制御するためのパストランジスタが配置されている．パストランジスタは 1 ビットの SRAM の値により制御される．

3.2 FPGA における実装

2 章で述べた基本アイディアを FPGA 上で実現することを考える．

FPGA のチップには耐タンパー性があると仮定する．耐タンパー性とは，内部の情報を覗き見ることや実現する機能の改変が困難な性質のことである．スマートカードなどのハードウェアデバイスは，この性質を持つように実装されている．この仮定により，不正者は FPGA 内部で実現されている回路を直接覗き見ることはできないものとする．

FPGA では LUT，MUX，パストランジスタに対応して SRAM が付属し，この SRAM に 1 ビットの情報を書き込むことで，FPGA 上で任意の回路を構成することが可能である．したがって，FPGA 内部の SRAM の値を固定することで，内部に固定の回路を作ることができる．

内部に固定回路を持つ FPGA の実現方法の一例をあげる．この例では，チップ内部に SRAM のメモリ領域を 1 つと 1 度だけ情報の書き込みが可能な PROM（Programmable Read Only Memory）のメモリ領域を 2 つ用意する．SRAM の領域には構成データ作成者が作成した構成データ $R = \langle r_1, r_2, \ldots, r_l \rangle$ をロード

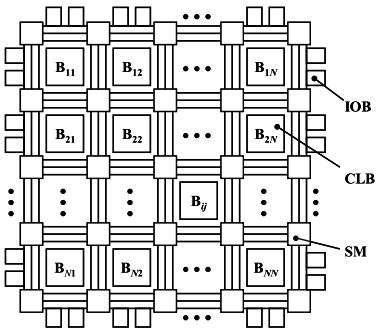

図 2 FPGA の構造
Fig. 2 An architecture of FPGA.

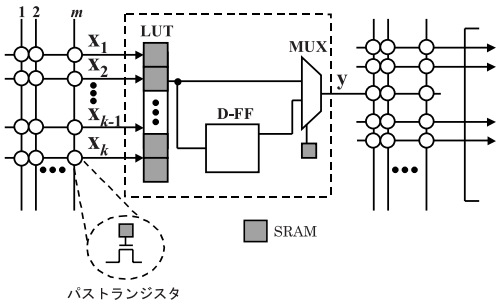

図 3 CLB の構造
Fig. 3 An architecture of CLB.

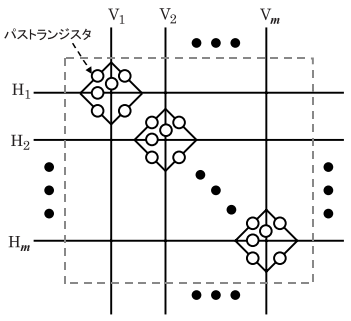

図 4 SM の構造
Fig. 4 An architecture of SM.

する．PROM のメモリ領域には固定回路の情報を記憶する．PROM で構成された領域の一方には固定する位置のデータ $P = \langle p_1, p_2, \ldots, p_l \rangle$ を記憶し，$p_i = 1$ なら固定の位置を示すとする．また，PROM で構成された領域のもう一方には，P に対応して固定する値のデータ $V = \langle v_1, v_2, \ldots, v_l \rangle$ を記憶する．なお，l は固定回路のない FPGA における再構成可能な部分の数であり，$r_i, p_i, v_i \in \{0, 1\}, 1 \leq i \leq l$ である．そして，$p_i = 1$ のとき v_i を選択し，$p_i = 0$ のとき r_i を選択するようなセレクタを用いて回路の再構成を行うことで，内部に固定回路を持つ FPGA を構成することができる．構成データ R は SRAM で構成されたメモリ領域にロードされるため，何度でも書き換えが可能である．また，P および V は PROM で構成されたメモリ領域に記憶するので，工場出荷時に一括で記憶することができる．

このような方法により，内部に固定の回路が作成された FPGA を固定回路入り FPGA と呼び，固定した回路 s を持つ固定回路入り FPGA を FPGA^{+s} とする．SRAM の固定の仕方は，プログラム可能なブロックに対応させる方法や，プログラム可能なブロックに対応させずに複数の CLB や SM にまたがるように固定する方法が考えられる．

3.3 構成データの保護能力を向上させる方法

FPGA では，構成データをロードした際に，回路内部にショートなどの矛盾がないかを検査する機構が付されていることがある．これは，構成データによって物理的に回路を構成するため，回路の一部にショートする部分があると，FPGA 自体が破壊される危険性があるためである．

構成データの保護の観点から，FPGA のこの特性を利用することを考える．FPGA^{+s} の固定回路 s をショートが起きやすいように構成しておく．固定回路 s を導出するために，不正者が自身で作成した構成データを FPGA^{+s} にロードし，実行して解析を行うような場合は，内部の回路がショートする可能性が高い．そのため，不正者がこのような解析を行おうとすると FPGA^{+s} を動作させることができない，または，動作させるとショートして FPGA^{+s} が破壊されるといったことが期待できる．これにより，不正者の FPGA^{+s} に対する解析行為が困難になると考えられる．FPGA においてこのような固定回路 s は，CLB の出力と導線をつなぐパストランジスタの値を固定しておくことにより実現できる．不正者ではない構成データ作成者は，固定回路 s を知っているため，FPGA^{+s} にロードしてもショートを起こさないような構成データを作成することが可能である．

4. 安全性の検討

提案した基本アイディアを FPGA で実現した場合の安全性に関して検討を行う．

サービス提供者は，関数 f を実現する FPGA の構成データ C を，サービス利用者ごとに割り当てられた秘密の固定回路 s を用いて C^{-s} に変換して配布する．サービス利用者は，C^{-s} を内部に秘密の固定回路 s を持つ FPGA^{+s} にロードして C として利用する．

不正者の目的は，f を実現する構成データ C を解析すること，および，別の関数 g を実現する構成データを構成し，FPGA^{+s} にロードすることなどが考えられる．不正者が FPGA^{+s} の固定回路 s を得られたなら，これらの目的を達成できることは自明である．このことから，s を導出することを不正者の目的とする．

不正者が FPGA^{+s} と C^{-s} を持っているとすると，FPGA^{+s} のみを用いて解析を行う場合，FPGA^{+s} と C^{-s} の両方を用いて解析を行う場合，C^{-s} のみを用いて解析を行う場合，が考えられる．

FPGA^{+s} が低い耐タンパー性しか有していないとき，不正者は FPGA^{+s} に探針を直接刺して s を覗き見るような解析を行うと考えられる．このような解析をプローブ解析として検討する．また，FPGA^{+s} は十分な耐タンパー性を有しているが，s を知らない者が作成した構成データを FPGA^{+s} にロードできるような場合には，不正者は FPGA^{+s} に自身で作成した解析用の構成データをロードして入出力を観測することで s の導出を行うような解析を行うと考えられる．このような解析を解析用構成データによる解析として検討する．プローブ解析，解析用構成データによる解析とも FPGA^{+s} のみを用いて行う解析である．

FPGA^{+s} が十分な耐タンパー性を有していて，s を知らない者が作成した構成データを FPGA^{+s} にロードして実行すると FPGA^{+s} が内部でショートして破壊されてしまうような危険性がある場合には，FPGA^{+s} のみを用いて解析を行うことはできない．このような場合，不正者は，FPGA^{+s} に C^{-s} をロードし f の入出力を観測しておき，C^{-s} の一部を変更して D とし，D を固定回路を持たない通常のの FPGA にロード，実行して入出力を観測し，観測しておいた f の入出力と一致するかを確かめる．一致した場合は，$\text{D} = \text{C}$ であるとして，D と C^{-s} を比較することで s に関する情報を入手する．一致しない場合には，f の入出力と一致するまで繰り返すような全数探索的な解析を行うと考えられる．このような解析は，FPGA^{+s} と

C^{-s} の両方を用いた解析である．

他方，C^{-s} は $FPGA^{+s}$ の固定回路 s により変換されたものであることから，$FPGA^{+s}$ を使用せずに C^{-s} から s の導出を試みる場合が考えられる．このような解析として，$FPGA^{+s}$ の基となる FPGA の構造と C^{-s} から，ショート箇所などを特定するような解析を行う場合や，C^{-s} と他の固定回路 t で変換された C^{-t} を比較して s，t を導出する解析が考えられる．

4.1 $FPGA^{+s}$ の解析

4.1.1 プローブ解析

暗号モジュールと鍵を内蔵した耐タンパー性のハードウェアに対して，探針を直接さしてレジスタの値を読み出して鍵を求めるプローブ攻撃が提案されている[10]．プローブ攻撃が成功するためには，暗号の構造を知っていて，かつ，所望のレジスタの値を直接読み出せる必要がある．このようなときに，所望のレジスタの値を数ビット読み出すことができれば，モジュール内部で保護されている鍵の導出に成功する．これは，レジスタの値は，暗号の構造と関連性があることを利用した解析であると考えられる．

これに対して，固定回路 s を内部に持つ $FPGA^{+s}$ は，s を構成する固定 SRAM の位置は任意である．したがって，$FPGA^{+s}$ に直接探針をさし，すべてのメモリ情報を入手する必要がある．たとえば，3.3 節であげた例においては，固定回路の位置を記憶した PROM とその位置に対応した値を記憶している PROM に対して探針を刺し，その内容をすべて取り出す必要がある．

4.2 解析用構成データを使用する解析

FPGA 内部で固定する SRAM を CLB に対応したものに限定して検討を行う．構成を固定された CLB を固定 CLB，プログラム可能なブロックのことを可変 CLB と呼ぶことにする．

不正者が，$FPGA^{+s}$ に対して任意の構成データをロードできる場合，1 つの CLB に外部から入力を与え，出力が得られるように作成された構成データを $FPGA^{+s}$ にロードし，入出力を観測することで，固定 CLB か可変 CLB かの判定が可能である．この操作をすべての CLB に対して行った後，固定 CLB の回路を特定していく解析法が考えられる．以下，固定 CLB 位置の特定の過程と固定 CLB 内部の回路を特定する過程に分けて説明する．

固定 CLB の位置特定

まず，固定 CLB とその周りに配置されている m 本の導線との接続を制御するパストランジスタが固定されていない場合（**図 5** (a)）に関して，固定 CLB の位置を特定する過程について述べる．FPGA の $N \times N$ の CLB に対して，左上の 1 行 1 列目から B_{11} とし，右下の N 行 N 列目を B_{NN} とする（図 2）．

$B_{ij}(1 \le i, j \le N)$ に対して，k ビット入力の AND ゲートを構成し，IOB を通して B_{ij} の入力と出力が観測できるように構成データを作成し，$FPGA^{+s}$ にロードする（**図 6**）．次に IOB を通して B_{ij} に k ビットの可能な入力パターンをすべて入力し，IOB から B_{ij} の出力を観測する．もし，外部で観測できる出力が k ビット入力の AND ゲートの出力と異なる場合には，B_{ij} が固定 CLB であると判断できる．もし，AND ゲートの出力と同じ場合は，固定 CLB が k ビット入力の AND ゲートであるときや，LUT は AND ゲートとは別のものであるが D-FF を使うように固定 CLB が構成されていて偶然に k ビット AND ゲートの出力と同じものが得られるときが存在するため，可変 CLB であるとは特定できない．そこで，B_{ij} に k ビット入力 OR ゲートと同じ出力が得られるように LUT の値をセットし，IOB を通して B_{ij} の入力と出力が観測できるように構成データを作り，$FPGA^{+s}$ にロードする．そして，k ビットのすべての入力パターンを入

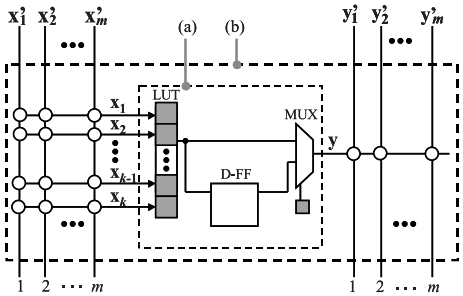

図 5 固定 CLB の範囲
Fig. 5 The architecture of CLB.

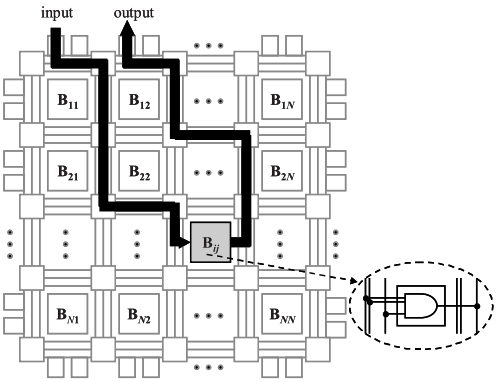

図 6 固定 CLB の位置特定に用いる構成データ
Fig. 6 The configuration data for searching fixed CLB.

力し，出力の値を観測する．B_{ij} のすべての出力が k ビット入力の OR ゲートのものと同じ場合には，B_{ij} は可変ブロックであると判断できる．B_{ij} のすべての出力が k ビット入力の OR ゲートのものと異なる場合には，固定ブロックであると判断できる．この理由は，各ゲートに関して 1 が出力される入力パターンは，AND ゲートは $(x_1, x_2, \cdots, x_k) = (1, 1, \cdots, 1)$ のときの 1 通りに限るのに対して，OR ゲートの場合は全入力パターンのうち $(x_1, x_2, \cdots, x_k) = (0, 0, \cdots, 0)$ を除く $2^k - 1$ 通りに関して 1 が出力されるため，固定 CLB が D-FF を使用していたとしてもすべての出力が一致することはないためである．

この検査ですべての固定ブロックの発見にかかるコストは，1 つの構成データに対して 2^k 通りの入力と，最大 $2N^2$ 回の構成データをロードする必要があるため，最大で $2^k \times 2N^2 = 2^{k+1}N^2$ 回の入力を行う必要がある．

固定 CLB とその周りに配置されている m 本の導線との接続を制御するパストランジスタも固定されている場合の固定 CLB 特定の過程は，上述のものとほぼ同じである．異なる点は，CLB への入力となる $x_1, x_2, \cdots, x_k$ に接続する可能性のあるすべての導線を $x'_1, x'_2, \cdots, x'_m$ として入力を与える点と，CLB の出力 y を直接観測する代わりに対象となる CLB の出力 y が接続する可能性のあるすべての導線を $y'_1, y'_2, \cdots, y'_m$ として観測する点である（図 5 (b)）．これは，CLB の入力 $x_1, x_2, \cdots, x_k$ および出力 y と，CLB の周囲に配置された導線との接続を制御するパストランジスタも固定されていて，直接に観測することができないためである．

この場合に，固定 CLB を特定するためにかかるコストは，$2N^2$ 回の構成データのロードと，各構成データに対して 2^m 回の入力が必要となる．したがって，合計で最大 $2^{m+1}N^2$ 回の入力が必要となると考えられる．

Xilinx XC4000 シリーズでは，N は 4 から 56 の製品がある．また，CLB 内の LUT への入力数は $k = 4$ である．$N = 32$ であるとすると，CLB のみが固定であるときの固定 CLB の位置特定には，$2N^2 = 2^{11}$ 回の構成データのロードと最大で $2^{k+1}N^2 = 2^5 \times 32^2 = 2^{15}$ 回の入力を行う必要がある．また，$m = 8$ であるとしたとき，CLB の回りの配線まで固定されている場合の位置特定には，$2N^2 = 2^{11}$ 回の構成データのロードと最大で $2^{k+1}N^2 = 2^9 \times 32^2 = 2^{19}$ 回の入力を行う必要がある．このことから，CLB のみを固定回路とするだけでは，安全性の面からみて十分ではないと推察される．したがって，SM などのブロックも含めて固定ブロックとすることが望まれる．

固定 CLB の回路特定

固定 CLB であるすべての B_{ij} を見つけた後，その回路構成を調べる．CLB の回路構成を決定するのは LUT の内容と D-FF を使用しているかを決定している MUX に付属した SRAM の値である．したがって，固定 CLB は，LUT の SRAM と MUX に付属した SRAM が同時に固定されている状態である．

IOB を通して B_{ij} の入出力が観測できるように，IOB と B_{ij} の入力および出力をつなぐように構成データを作り，$FPGA^{+s}$ にロードする．B_{ij} が D-FF を使用しない構成をしている場合には，すべての入力パターンを入力することで出力を観測することで，LUT の内容を知ることができる．しかし，B_{ij} が D-FF を使用する構成をしている場合には，出力から観測できる値は，1 クロック遅れることになる．したがって，固定 CLB の構造を解析する際には，この 2 つの場合を見分けることが求められる．

そこで，D-FF の特性からこれら 2 つの場合を識別可能であるか考察する．D-FF は，CLOCK 信号が入力されたときに入力された値を内部に記憶する．そして，保持している値を出力しつづける．したがって，CLOCK 信号を解析に利用できる場合は，次の 2 つのステップ（ステップ 1，ステップ 2）を行うことで，B_{ij} の回路の特定ができる．

ステップ 1 では，固定 CLB である B_{ij} に，CLOCK 信号 1 つに対して 1 つの入力を行いながら，可能な限りの入力パターンを順番に入力していき，最後に $(x_1, x_2, \cdots, x_k) = (0, 0, \cdots, 0)$ を入力し，出力を観測する．出力が恒等的に 0 または 1 のときは，B_{ij} が D-FF を使用しているか否かによらず，恒等的に 0 または 1 を出力する固定 CLB であると判断する．それ以外の場合には，ステップ 2 に進む．

ステップ 2 では，B_{ij} に対して CLOCK の入力は行わず，すべての入力パターンを入力し，出力を観測する．もし，出力が一定の場合には，B_{ij} は D-FF を使用する構成であると判断できるので，ステップ 1 で観測した出力パターンを 1 つずらすことで，LUT の内容を知ることができる．また，もし，出力パターンに変化がある場合には，B_{ij} は D-FF を使用していないような構成であると判断できるので，ステップ 1 で観測した出力パターンにより LUT の内容を決定できる．

4.3 $FPGA^{+s}$ と C^{-s} を用いた解析

固定回路 s が内部に存在する $FPGA^{+s}$ に対して，s を知らない不正者が作成した構成データをロードで

きない場合を考える．FPGA^{+s} の内部では関数 f が構成データ C により構成されており，不正者は C の入出力を観測できるものとする．不正者は C^{-s} を入手している場合，仮定より FPGA^{+s} に対して C^{-s} をロードできないが，内部に固定回路を持たない通常の FPGA には C^{-s} をロードでき，その入出力を観測できるとする．

関数 f を実現する構成データ C と C を固定回路 s を用いて変換した構成データ C^{-s} が組合せ回路で，使用している CLB の位置および CLB 間の接続が同じであるとする．このとき，C と C^{-s} では，1 つまたは複数の LUT の値のみが異なる．この場合には，C^{-s} の 1 つまたは複数の LUT の値を置き換えて構成データを作成する．このようにして作成した構成データを D とする．そして，D を通常の FPGA にロードして，すべての入出力値を観測し，FPGA^{+s} のすべての入出力値と一致するかの判定を行う．一致する場合には，D が C であると判断できる．一致しない場合には，再度同じ処理を行う．すべての可能性のある D に対してこの操作を試せば，必ず C を見つけることができる．

この方法で C^{-s} から C を導出するのにかかるコストを求める．FPGA 内部にある N^2 個の CLB のうち，C および C^{-s} が使用している CLB の集合を Ω として，$\sharp\Omega = \omega$ とする．また，前提から Ω は不正者にとって既知である．C，C^{-s} は K ビットの入力を必要とし，B $\in \Omega$ の入力数を $k_B \leq k$ とする．このとき，B に対して置き換え可能な LUT の総数は，

$$2^{2^{k_{\mathrm{B}}}} \leq 2^{2^k}$$

である．したがって，不正者が作る D の総数は，最大で，

$$\prod_{\mathrm{B}\in\Omega} 2^{2^{k_{\mathrm{B}}}} \leq 2^{2^k\omega}$$

となる．また，各 D に対して最大 2^K 回の入力を行うので，不正者が行う入力の回数は

$$2^K 2^{2^k\omega} = 2^{K+2^k\omega}$$

となる．

CLB の入力数 $k = 4$ としたとき，$\omega \geq 4$ であれば $2^{K+2^k\omega} \geq 2^{64}$ となる．また，$\omega \geq 8$ であれば $2^{K+2^k\omega} \geq 2^{128}$ となる．Xilinx XC4000 シリーズの CLB 数が $N = 4 \sim 56$ であることを考慮すると，$\omega = 4$ または $\omega = 8$ は十分実用的な数値であると考えられる．

4.4 C^{-s} を用いた解析

構成データ C^{-s} から FPGA^{-s} の固定回路 s を導出する解析として，FPGA^{+s} の基となる FPGA の構造と C^{-s} から s を導出する解析と，C^{-s} と s とは異なる固定回路 t で C を変換した C^{-t} を比較することにより s，t を導出する解析が考えられる．

FPGA^{+s} の基となる FPGA の構造と C^{-s} から s を導出する解析としては，FPGA^{+s} 上での C^{-s} の配線イメージを作成し，ショートなどの回路的に矛盾をなす SRAM の位置を特定し，そこから s を導出するような解析が考えられる．これは，C^{-s} において回路的に矛盾を起こしている部分が固定回路 s に関係しているであろうことを利用した解析である．しかし，C^{-s} が回路的に矛盾がないようなものの場合にはこのような解析は適用できなくなると考えられる．ただし，FPGA 上での C^{-s} の配線イメージからどの程度固定回路 s を特定できるか，また，回路的に矛盾のない C^{-s} の作成に関しては今後検討を行う必要がある．

固定回路 s とは異なる固定回路 t で C を変換した C^{-t} と C^{-s} を比較することで s，t を導出する解析は，C^{-s} と C^{-t} では s，t に関連した部分のみが異なっているであろうことに着目した解析である．しかし，C^{-s}，C^{-t} は s，t をうまく利用するように C を変換したあとに作成されるとすると，s，t に関連しない部分に関しても C^{-s}，C^{-t} では違いが生じるようになると考えられる．ただし，どの程度差異が生じるかについては今後検討する必要がある．

5. ま と め

本稿では，内部に秘密の固定回路を持つ再構成可能ハードウェアに，この秘密の固定回路をもとに作成された構成データをロードすることで，本来の機能を実現する基本アイディアを提案した．また，この基本アイディアを FPGA で実現することに関して検討を行った．

今後の課題は，PLD において基本アイディアが実現できるか検討することと，FPGA で基本アイディアを実現した場合の安全性の詳細な検討があげられる．FPGA で基本アイディアを実現した場合の安全性の検討においては，固定ブロックが CLB と SM であるときの解析用構成データを使用する解析の困難さの検討，および，固定回路入り FPGA にロードされる構成データのみを用いた解析に関する検討があげられる．

謝辞 本研究に関して有益な助言をいただいた慶応義塾大学理工学部情報工学科春山真一郎，ソニー（株）Lachlan B. Michael，横浜国立大学大学院吉岡克成の

各氏および査読担当委員の方々に深く感謝する．

参 考 文 献

1) 赤井健一郎，松本　勉，河野隆二：ロードされた構成データを修飾して使用する安全性強化型FPGA，電子情報通信学会技術研究報告，SR02-11, pp.15–21 (2002).
2) Lachlan, B.M., Mihaljevic, M.J., Haruyama, S. and Kohno, R.: A Proposal for Secure Software Download in Software Defined Radio, Technical Report of IEICE, SR01-23, pp.1–8 (2001).
3) Xilinx Inc.: Virtex-II Pro™, Platform FPGAs: Introduction and Overviews, Advance Product Specification, v2.0 (2002).
4) Xilinx Inc.: Virtex-II 1.5V Field Programmable Gate Arrays, Advanced Specification, v1.7 (2001).
5) Lach, J., Mangione-Smith, W.H and Potkonojak, M.: Fingerprinting Digital Circuits on Programmable Hardware, *Proc. Information Hiding Workshop '98*, pp.16–31 (1998).
6) 田丸啓吉：やわらかい LSI デバイス：FPGA，情報処理学会誌，Vol.40, No.8, pp.783–788 (1999).
7) Xilinx Inc.: VirtexTM-E 1.8 V Extended Memory Field Programmable Gate Arrays, Preliminary Product Specification, v1.4 (2001).
8) Xilinx Inc.: VirtexTM 2.5V Field Programmable Gate Arrays, Product Specification, v2.5 (2001).
9) Xilinx Inc.: XC4000E and XC4000X Series Field Programmable Gate Arrays, Product Specification, version 1.6 (1999).
10) Handschuh, H., Paillier, P. and Sterm, J.: Probing Attacks on Tamper-Resistant Devices, *Proc. CHES '99*, pp.303–315, Springer-Verlag (1999).

(平成 14 年 12 月 2 日受付)
(平成 15 年 6 月 3 日採録)

赤井健一郎

2001 年横浜国立大学大学院工学研究科人工環境システム学専攻博士課程前期修了．同年同大学院環境情報学府情報メディア環境学専攻博士課程後期に進学，現在に至る．情報セキュリティの研究に従事．

松本　勉（正会員）

1986 年東京大学大学院博士課程修了，工学博士．同年横浜国立大学工学部専任講師．同助教授，教授を経て，2001 年より同大学大学院環境情報研究院教授．1981 年より暗号や情報セキュリティの研究に従事.「明るい暗号研究会」創設メンバー．現在，暗号アルゴリズム，情報利用管理，ディジタル証拠性，情報ハイディング，バイオメトリクス，人工物メトリクス，耐タンパーソフトウェア等に広く関心を持つ．国際暗号学会 IACR 理事．暗号技術検討会構成員．ASIACRYPT '96 プログラム委員長．ASIACRYPT 2000 実行委員長．電子情報通信学会より「情報セキュリティの基礎理論」への貢献に関して業績賞を受賞．

河野　隆二（正会員）

1979 年横浜国立大学工学部情報工学科卒業．1984 年東京大学大学院博士課程修了．工学博士．同年東洋大学工学部講師．1986 年同大学工学部電気工学科助教授，1988 年横浜国立大学工学部電子情報工学科助教授を経て，1998 年より同教授．1984 年〜1985 年カナダ，トロント大学客員研究員．情報通信システム，情報理論，符号理論，ディジタル信号処理，スペクトル拡散通信（CDMA），移動通信，高度交通システム（ITS）の研究に従事．1998 年〜2002 年ソニーコンピュータサイエンス研究所先端情報通信研究室室長兼業，2002 年より（独）通信総合研究所 UWB 結集型特別グループリーダ併任，2002 年文部科学省 21 世紀 COE プログラム「横浜国立大学：情報通信技術に基づく未来社会基盤創生」拠点リーダ．電子情報通信学会スペクトル拡散研究専門委員会委員長，同 ITS 研究専門委員会委員長，同ソフトウェア無線時限研究専門委員会委員長，IEEE Transactions on Communications および Transactions on Information Theory の Editor，IEEE Information Theory Society 理事等を歴任,「スペクトル拡散通信に関する先駆的研究」で 1999 年度電子情報通信学会業績賞受賞．