社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－SLDM－124 (20)
2006／3／17

論理プログラマブルデバイスの保護アーキテクチャ

横山 浩之† 堀 洋平‡ 戸田 賢二‡

†株式会社ＫＤＤＩ研究所 〒356-8502 埼玉県ふじみ野市大原 2-1-15
‡独立行政法人 産業技術総合研究所 〒305-8568 茨城県つくば市梅園 1-1-1 中央第２
E-mail: †yokoyama@kddilabs.jp, ‡hori.y@aist.go.jp and ‡k-toda@aist.go.jp

あらまし 論理プログラマブルデバイスの内部または外部に保護回路を用意することによって，たとえ誤りや悪意のある回路構成データがデバイスに入力されたとしても，デバイスおよび周辺回路が誤動作または破損することを防止できる保護アーキテクチャについて検討する．

キーワード FPGA，回路保護，回路構成データ，アーキテクチャ

Protection Systems for Programmable Logic Device against Circuit Malfunctions

Hiroyuki YOKOYAMA†, Yohei HORI‡ and Kenji TODA‡

†KDDI R&D Laboratories, Inc. 2-1-15 Ohara, Fujimino-shi, Saitama 356-8502, Japan

‡National Institute of Advanced Industrial Science and Technology
Tsukuba Central 2, 1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan

E-mail: †yokoyama@kddilabs.jp, ‡hori.y@aist.go.jp and ‡k-toda@aist.go.jp

Abstract We propose a new architecture of circuit protection systems that prevent electrical circuits to be physically broken due to malfunctions originated from the reconfiguration data installed in a programmable logic device.

Keyword FPGA, Circuit Protection, Reconfiguration Data, Architecture

1. はじめに

端末の高機能化と高速化の要求に伴って，システムに論理プログラマブルデバイス (以下，FPGA で代表する)を搭載し，ネットワーク経由で回路構成データを取得することにより，ソフトウェアだけでなくハードウェアについても，製品出荷後に新機能を追加し不具合を修復することのできる柔軟性が求められつつある．一方，端末は，重要な個人情報を管理すると共に，付加価値の高いコンテンツを取り扱うための暗号・認証・課金等の処理を行う必要があり，高いセキュリティ強度が要求されている．そうした要求に応えるための一つのアプローチとして，データやアルゴリズムを FPGA 内部に回路として秘匿し，コンテンツに合わせて動的に構成を変更するシステムアーキテクチャが提案されている[1][2].

このように，必要に応じて端末の外部から回路構成データを取得し，端末内部にある FPGA を再構成して，機密に関わる情報を処理するようになると，FPGA が何らかの理由で誤動作した場合の対策が問題となる．特に FPGA の場合，回路構成によっては，FPGA 自身や周辺回路に対して物理的損傷を与える危険性があるため，より安全性の高い方法が必要である．一方，従来は，出荷前の検査によって FPGA の動作は十分に検証され，回路構成には問題がないことを前提にシステムが設計されていた．そのため，保護メカニズムとしては，回路構成データが正確に入力され，正しく回路に反映されることを目的としたものが多かった．しかし，今後，端末内の FPGA を出荷後に動的に再構成して使用する用途が多岐にわたり，かつ，高機能化による回路の大規模化・複雑化が進行すれば，事前検証を十分に行うことがそもそも困難になり，回路が誤動作しないことを前提としたシステムは成り立たなくなる可能性がある．さらに，ネットワークを経由して不特定の製作者による回路構成データの入力が許される場合は，悪意のある回路構成データを用いた意図的な攻撃に対する安全対策も必要になる．

そこで本稿では，ネットワーク経由で端末内の FPGA を動的に再構成するシステムを対象に，FPGA の内部または外部に保護回路を用意することによって，たとえ誤りや悪意のある回路構成データが入力されたとしても，FPGA および周辺回路が誤動作または破損することを防止できる保護アーキテクチャについて提案する．提案アーキテクチャは，設計・開発・検証を行うための環境としても有用である．

2. ネットワーク対応型動的再構成システム

2.1. システム構成

ネットワーク経由で端末内のFPGAを動的に再構成するネットワーク対応型動的再構成システムの概念図を図1に示す．各端末の内部にはFPGAが搭載されており，ユーザがサービスを利用する場合は，必要に応じて，ソフトウェアと共に回路構成データをサーバからダウンロードし，端末内のFPGAを再構成する．

ここで，回路構成データに対する伝送経路上における盗聴・改ざん・成りすまし等は，Secure Socket Layer（SSL）等の通信プロトコルによって防御できると仮定する．また，端末内においても，AES（Advanced Encryption Standard）[3]方式等で暗号化された回路構成データをFPGAに入力し，これをFPGA内部で復号化して回路構成を行う方法や[4][5]，MD5等のハッシュ方式を用いて証明書との照合を行う方法が利用可能であると仮定する．即ち，回路構成データそのものについては，少なくとも機密性と安全性は保証されているものと仮定する．

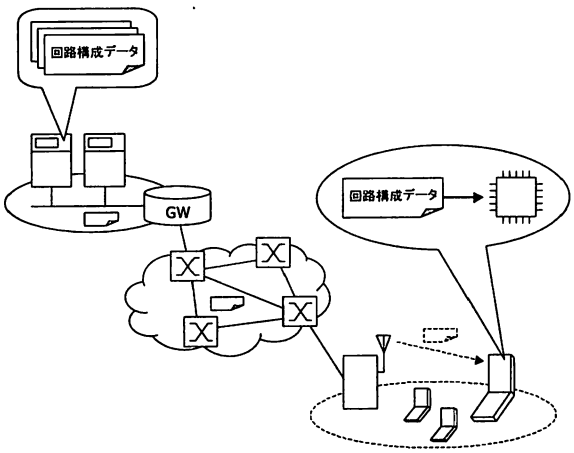

図1：ネットワーク対応型動的再構成システム

2.2. 解決すべき課題

ここでは，回路構成データの設計上の誤りに起因する誤動作からシステムを保護することを課題とする．つまり，最初から誤動作するように回路が構成されている可能性があることを前提に，システムに対する悪影響を防止する方法を検討する．

このアプローチは，従来の，宇宙放射線によってデバイス内の情報ビットが反転する現象（Single Event Upset, SEU）による誤動作を防止する考え方と共通する部分もあるが，本質的には全く異なるものである．人工衛星等へ搭載されるFPGAの中には，内蔵の専用回路が，継続的かつ自動的に回路データのCRC（Cyclic Redundancy Code）を検証し，デバイス上で生じたSEUを発見・修復する機構[6][7]を実装しているものがある．しかし，これらの機構は，あくまでも，入力された回路構成データの設計が正しいという前提で有効に機能するものであり，最初から設計に誤りがある場合には対応できない．2.1で述べた機密性・完全性を保証するメカニズムについても同様に，ひとたび回路構成データの正当性や完全性が確認されてしまえば，以後，回路の誤動作を防ぐ手段がない．端的に言えば，正規の回路構成データに最初から含まれている設計上の誤りに対して，従来技術は無力なのである．

これまで，回路設計上の誤りに起因する誤動作が問題にならなかったのは，最終的な製品の出荷時点までに回路の動作検証を完了することが十分現実的であり，誤動作した場合の対策を別途講じる必要性やメリットがなったからである．しかし，今後，FPGAを出荷後に再構成して使用する用途が多岐にわたり，かつ，高機能化によって再構成すべき回路が大規模化・複雑化すれば，あらゆる利用状況を想定した動作検証を実施するのは原理的に困難となる．

したがって，たとえ誤りや悪意のある回路構成データが入力されたとしても，FPGAおよび周辺回路が誤動作または破損することを防止できる保護アーキテクチャを利用することによってシステムの安全性を高めるアプローチには，実用上のメリットがある．

回路の誤動作による論理的・物理的な損害の具体例を以下に示す．

(1) 端末内情報の読出・複製
(2) 端末内情報の改ざん・消去
(3) 当該端末の機能上の誤動作（写真を取る，アラームが鳴る，リセットがかかる）
(4) 当該端末の物理的損害（FPGAの出力ドライバの熱破損，周辺回路の電気的破損）
(5) 他端末に対する機能上の誤動作（電話をかける，パケットを出力する）
(6) ネットワークの中継点としての誤動作（他端末に対する攻撃の踏み台）

上記内容の多くはソフトウェア・ウィルスによる損害と全く同様であり，FPGAに特化した問題は，(4)の物理的損害の項目に集約されていると言える．こうした損害を引き起こす原因がFPGAの内部回路にある場合，問題を根本的に解決するには，回路によるアプローチが有効である．

3. デバイス保護システムの構成

3.1. デバイス保護システムの基本構成

図２に，FPGA の誤動作による損害を防止するための保護回路に関する概念図を示す．保護回路は，回路構成データの内容を事前に検査する回路構成データ検査機能と，回路構成されたデバイスを動作させながら，状態を監視・制御する機能から構成されている．

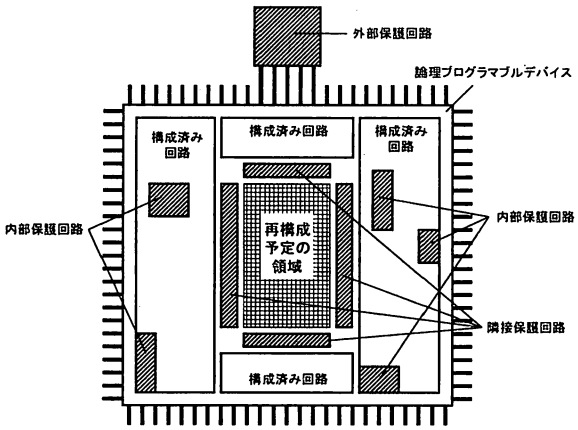

図２：保護システムの基本構成

回路構成データ検査機能は，当該デバイスに入力された回路構成データにおける回路の大きさや位置が所定の範囲であること，構成される回路が所定のインタフェース条件を満たすこと，例えば，信号を入出力する回路が配置される位置や電圧等が妥当であることを確認する．図３に回路構成データ検査機能のブロック図を示す．

図４に回路状態監視制御機能の機能ダイヤグラムを示す．回路状態監視制御機能は当該デバイスの回路を監視し，異常状態を検出して，回路を保護する動作を行う．異常状態の例を以下に示す．

(a) 論理回路が，本来とりえない論理状態あるいは望ましくない論理状態（禁止状態）に遷移する
(b) 発熱量が増加し，温度が一定値を越える
(c) 電力消費量が一定値を越える
(d) 電圧や電流の振幅が所定範囲から逸脱する
(e) 外部信号が不当にブロックされる
(f) 信号にノイズが混入する

回路保護動作の例を以下に示す．

(a) 回路構成動作の停止・禁止
(b) 所定の入出力ポートに対する警告信号の出力
(c) 所定の入出力ポートの駆動停止
(d) 再構成予定領域の強制的再構成
(e) 当該デバイスにおける所定領域の強制的再構成や電源供給の停止
(f) 当該デバイスの再起動または初期化

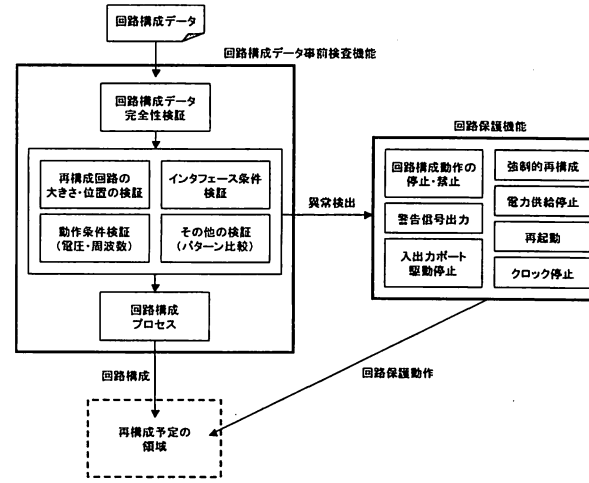

図３：回路構成データ検査機能のブロック図

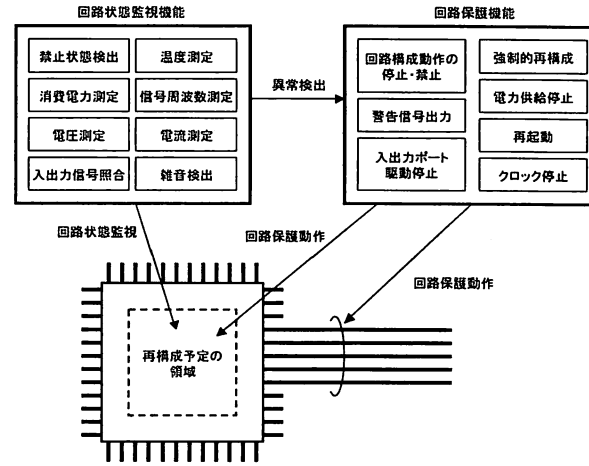

図４：回路状態監視制御の機能ダイヤグラム

保護回路のうち，当該デバイスの内部に構築されたものを内部保護回路，当該デバイスの外部に構築されたものを外部保護回路と呼ぶ．内部保護回路のうち，回路構成予定の領域に隣接する回路を，隣接保護回路と呼ぶ．

内部保護回路の実現方法としては，当該デバイス上に固定的な物理回路として実装する方法と，回路構成データを用いて動的かつ論理的に構築する方法がある．

前者の，固定的な物理回線として実装する方法は，当該デバイスのプログラマブルな領域を全面的に再構成しても，保護機能を使用することができるという長所がある．後者の，保護回路を動的かつ論理的に構築する方法は，コンパイラや CAD 等の開発用ツールと連携することで，様々な禁止条件や保護動作を，固定的なパターン，ルール，あるいはプログラムとして指定し，適切な場所に配置することができる点で自由度が高く柔軟性に優れている．

例えば，出力信号 A と B が同時に 1 となることを禁ずるというメタルールがあるとする．このとき，A および B の本来の値 A_{ORG}，B_{ORG} を次のように変換することで安全性が保証できる．

$A \leftarrow A_{ORG}$

$B \leftarrow B_{ORG}$ & $\neg A_{ORG}$

また，メタルールに対する違反を検出するには

$C \leftarrow A_{ORG}$ & B_{ORG}

で定義される C を取り出せばよい（図 5）．こうした回路をルールに応じて動的に生成することで，柔軟に回路の監視や保護を行う機能を実装できる．

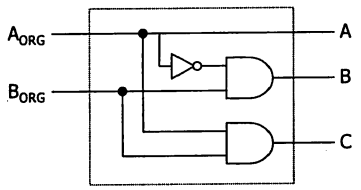

図 5：メタルールによる安全性の保証の例

FPGA の動作によって FPGA 自身が破損する例として，異常発振による熱破壊を考える．回路設計の不備により，FPGA に対する想定外の入力に起因して出力が発振してしまう場合がある．その際の出力が，一定時間に許容限度を超えて hi/low を繰り返すと，FPGA 内の出力ドライバが発熱によって破壊されることがある．そこで，FPGA の最終段における出力ドライバと，それを駆動する内部回路との間に保護回路を挿入し，出力の異常発振を防止する．例えば，現在から遡る一定時間内の hi/low の遷移回数を測定し，それが閾値を超えた場合は，それ以上の hi/low の切り替えを抑制する．さらに，別の出力ポートから，内部回路に異常発振が生じていることを通知する．こうした機能を内部保護回路として合成することにより，出力ドライバ側に温度検知機能や電流測定機能等の特別な回路を用意しなくても，FPGA を発熱による破壊から保護することができる．

外部保護回路は，FPGA の外部に保護機能を提供する回路を用意したものであり，ASIC 等の固定的な回路や，別の FPGA によって動的に構成する回路によって実現する方法がある．外部保護回路は，当該デバイス内部の回路設計に一切変更を加えることなく保護機能を追加できる点で優れている．また，当該デバイスの回路規模が小さい等の理由で，その内部に保護回路を構築できない場合にも保護手段を提供できる．

3.2. デバイス保護システムの構成例

図 6 および図 7 にデバイス保護システムの構成例を示す．図 6 は Xilinx 製の FPGA における部分書換方式を用いて再構成予定の領域を書き換える場合を想定したものである．現状の FPGA アーキテクチャにおける制約条件より，部分書換は 4 カラム単位で行う必要がある．したがって，図 6 のように，部分書換を行う領域は上下の入出力ピンに隣接しており，これらのピンに対する保護を行うには，FPGA の外部に保護回路を用意する必要がある．本図では，監視および制御を行うために，再構成予定領域に隣接するカラムへ隣接保護回路を構築し，入出力インタフェースの信号特性や，内部状態を監視している．これらの情報をもとに，異常を検出し，外部または内部保護回路において回路保護機能を実現する．

図 7 は再構成予定の領域が複数同時に存在する場合を想定した構成例である．それぞれの再構成予定領域を異なる用途に利用するならば，回路の役割や性能の組み合わせによって，異常検出の方法を修正する必要が生じる可能性がある．こうした場合に柔軟に対応するには，回路状態監視機能を FPGA 内に動的に生成する方法が優れている．

4. 考察

上記のように，FPGA の内部または外部に保護回路を用意することによって，たとえ誤りや悪意のある回路構成データが従来のセキュリティメカニズムを通過し，当該デバイスに入力されたとしても，当該デバイスおよび周辺回路の誤動作または破損を防ぐことが可能になると考える．

内部保護回路を当該デバイスの内部に構築すれば，周辺回路に対して物理的な変更を行うことなく，回路全体の安全性を高めることができる．外部保護回路を用いれば，当該デバイスを全面的に回路構成する用途に対しても，回路を保護する手段を提供することができる．回路構成予定の領域が当該デバイスの外周に面し，デバイス内に隣接保護回路を用意することが物理的に不可能である場合や，対象デバイス内部の回路容量の制約が厳しい場合にも適用可能である．

回路保護アーキテクチャは，図 8 に示すように，FPGA を用いた回路の開発環境として，動作検証やデ

バッグを行う目的に応用することもできる．その際，回路状態監視機能は，開発対象回路における動作をトレースするためのプローブとして利用可能である．また，回路保護機能をクロックの制御や内部状態のリアルタイム出力ができるように拡張することで，JTAGを用いた開発支援システムよりも実時間性に優れた分析を柔軟に行うことが可能になる．さらに，保護回路に対して，対象デバイスへの入出力信号を模擬する機能を追加すれば，実際には接続されていないペリフェラルを仮想的に取り扱うことができる．また，外部に接続されている機器の情報に基づいて，保護回路の設定を動的に変更する機能を加えることで，さらに開発や検証の利便性や柔軟性を高めることができる．

5. まとめ

論理プログラマブルデバイスの内部または外部に保護回路を用意することで，たとえ誤りや悪意のある回路構成データがデバイスに入力されたとしても，デバイスおよび周辺回路が誤動作または破損することを防止できる保護アーキテクチャについて検討した．

様々な構成例に対応したシステムの実装方法，セキュリティ強度の高い保護回路の実現に関する検討は今後の課題である．

謝辞

日頃御指導いただくKDDI研究所秋葉所長，松本取締役，野本執行役員，産業技術総合研究所大蒔コーディネータ、坂上情報技術研究部門長に感謝致します．

文　献

[1] 横山浩之，戸田賢二，“FPGA を用いたコンテンツ保護システムの開発”，信学技報 CPSY2004-114, pp.55-60, March 2005.

[2] H. Yokoyama and K. Toda, “FPGA-Based Content Protection System for Embedded Consumer Electronics,” Proc. RTCSA2005, pp.502-507, Aug. 2005.

[3] U.S. Department of Commerce/National Institute of Standards and Technology, “Announcing the Advanced Encryption Standard (AES),” FIPS PUB 197, Nov. 2001.

[4] Xilinx, “Secure Chip AES bitstream encryption/ decryption technology,” http://www.xilinx.com/

[5] Altera, “Design Security in Stratix II Devices,” http://www.altera.com/

[6] Xilinx, “Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Complete Data Sheet,” Oct. 2005.

[7] Altera, “Automatic Cyclic Redundancy Code Checking in Stratix II FPGAs,” http://www.altera.com/

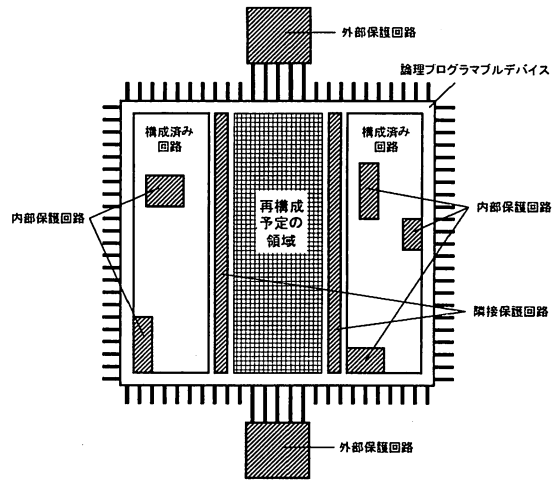

図６：デバイス保護システム構成例（１）

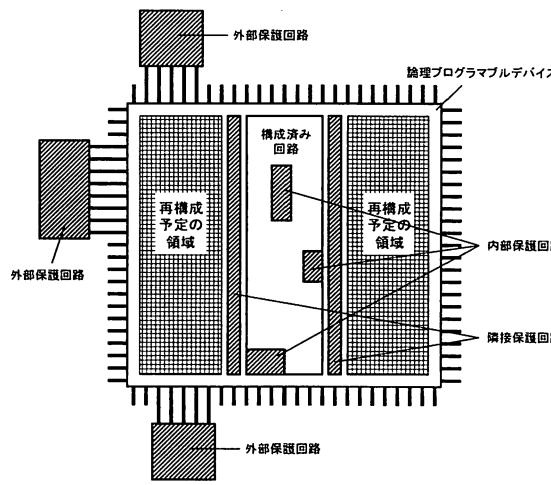

図７：デバイス保護システム構成例（２）

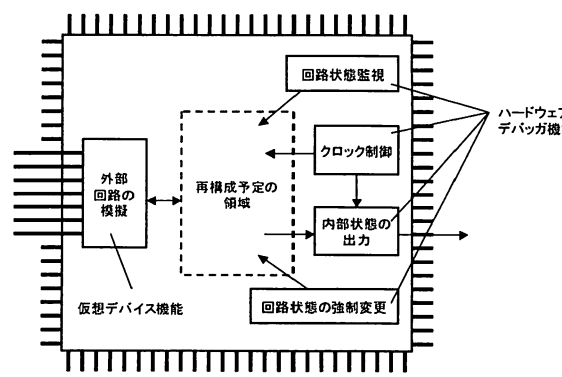

図８：ハードウェアデバッガの機能ダイヤグラム