キーワード: IP, 著作権侵害, FPGA, 海賊対策, 盗み, 保護, SHA, SHA-1, MAC, 認証, Xilinx, Altera, 知的財産

Mar 01, 2010

アプリケーションノート4594

著作権侵害からのFPGAの保護：SRAMベースのFPGA設計のIPを保護するコスト効率に優れた認証方式

筆者：Bernhard Linke

要約：このアプリケーションノートでは、*FPGA* (フィールドプログラマブルゲートアレイ)と、*FPGA*がシステムの重要な機能と知的財産*(IP)*をどのようにして保持することができるのかについて説明します。*IP*を著作権侵害から保護する方法について記載しています。*SHA-1*のチャレンジ&レスポンス認証は最も安全な方法として評価されています。このアプリケーションノートでは、*SRAM*ベースの*FPGA*設計の*IP*を保護するコスト効率に優れた認証方式を紹介します。*DS28E01*と*DS28CN01*のデバイスに注目します。

このアプリケーションノートは、2008年7月に「Electronic Design」誌および出版社のウェブサイトにアーティクルとして掲載されました。

この20年間で、FPGA (フィールドプログラマブルゲートアレイ)は、民生用と産業用の両方のアプリケーションにおいて、プロトタイピングツールから柔軟性のある製品ソリューションへと遷移してきました。FPGAロジックが数千ゲートから数百万ゲートに増大して複雑になるにしたがって、デバイスはシステムのより多くの重要な機能(知的財産すなわちIPなど)を保持することができるようになりました。

今日、設計者は、さまざまな技術を用いて構成データを保持するFPGAを選択することができます。この構成データとしては、OTP (ワンタイムプログラマブル)アンチヒューズ、再プログラム可能なフラッシュベースの記憶セル、および再プログラム可能でSRAMベースの構成可能論理セルがあります。構成データがFPGAチップに格納され、その格納データが読み取られないようにする機構があることから、アンチヒューズやフラッシュベースのソリューションは比較的安全なソリューションといえます。また、デパッキング、マイクロプロービング、電圧コントラスト法による電子ビーム顕微鏡検査、および集束イオンビーム(FIB)のプロービングなどの非常に高度な方式を用いてシリコン内部をのぞき見されたり、セキュリティの機構を無効にされたりしない限り[1]、データが危険にさらされる可能性は極めて低いと考えられます(FPGAの簡単な背景説明については、付録Aの「FPGAの技術的なオプションと問題」を参照してください)。

しかし、スタティックRAMベース(SRAM)のFPGAには、違法なコピーや盗難からIP (構成データ)を保護するための安全対策がほとんどありません。これは、いったんデータがロードされると、SRAMメモリセルに保持されるため、その内容を調べて簡単に把握することができるからです。また、チップに構成データをロードする前に構成データを保護する何らかのセキュリティ機構を備えていなければ、その構成データはのぞき見に対して無防備になります。FPGAが構成パターンをロードするとき、パワーアップ時にFPGAによって読み取られる個別のメモリチップに通常、ビットストリームが格納されるため、この構成データを検索することが可能になります。ただし、そのデータを保護する簡単な方法、つまり誰かに構成パターンをコピーされたりIPを盗まれたりしないようにする方法がいくつかあります。

SRAMベースのFPGAの弱点

FPGAと構成メモリとの2チップソリューションであることから、構成データのビットストリームはパワーアップの段階で露出されます。FPGAは、ビットストリームが「純正」であるか不法に入手したコピーであるかを見分けることができないため、構成データに格納されたIPはまったく保護されません。このよく知られた問題の一部は秘密鍵とビットストリームの暗号化で対処されています。ただし、これらの保護はハイエンドのFPGAに使用が限られています。コストが高いため民生用アプリケーションには適していないからです。

保護のない場合

ビットストリームを暗号化しないでSRAMベースのFPGAを使用するアプリケーションは、特に著作権侵害を受けやすくなります。構成ビットストリームはキャプチャされたり、構成PROM内で書き換えられたり、または単純に再現されたりして元の設計のクローンを作成される可能性があります。クローン製品は元の製品と競合するため、研究開発のための投資を奪い取ることになると同時に、元の製造業者の市場占有率と収益性を縮小させることになります。

アンチヒューズベースまたはフラッシュベースのFPGAは、構成データが露出されないため、暗号化しない場合でもSRAMベースのFPGAに比べて安全です。しかし、FPGAをプログラムするためにアセンブリハウスが設けられる場合、アセンブリハウスは認可されている数量以上の数量をプログラムして、開発コストをかけないで自らが販売することが可能です。このような認可されていないデバイスは認可されたデバイスと区別がつかないため、企業の収益性に著しい影響を与える可能性があります。

SRAMベースのFPGAを少し安全にする方法の1つは、マルチチップパッケージを活用して、パッケージにFPGAと不揮発性メモリを合わせて実装することです。しかし、誰かがそのパッケージを開けた場合は、メモリとFPGAのデータインタフェースはむき出しになり、構成パターンが危険にさらされる可能性があります。

構成ビットストリームの構造(すなわち、データ要素の並びと、これらの符号化と識別の方法)については、ほとんど文書化されていません。理論的には可能であっても[1]、ビットストリームの難解さ、複雑さ、およびサイズの大きさが、リバースエンジニアリングのプロセスを困難で時間のかかるものにしています。構成ストリームのリバースエンジニアリングが、たとえ部分的であっても成功した場合は、セットトップボックスに不正侵入してサービスを盗んだり、あるいは自動車のパワートレインの設定を改竄したりすることができるため、元の製造業者の責任問題が生じます。

チャレンジ…

システムのコストを爆発的に上昇させないようにするため、設計者はSRAMベースのFPGAを暗号化なしで使用し続けなければなりません。ただし、設計者は著作権侵害からIPを保護する方法を見つける必要があります。さらに、セキュリティ対策で追加されるコストをできるだけ低減して、生産工程への影響を最小限に抑える必要があります。

重要なことは、セキュリティに関するハードウェアが回路基板の利用可能スペース内に収まり、ただし全体の消費電力は増大しないということです。また、FPGAのリソース(ピンと論理素子の数)に与えるセキュリティの影響をできるだけ小さくする必要もあります。

アンドレスポンス：認証

認証プロセスの目的は、2つ以上の実在するものの間で同一性を立証することです。鍵をベースにした認証では、秘密鍵と認証対象のデータ(すなわち、「メッセージ」)を入力として、メッセージ認証コード(MAC)を計算します。次に、MACをメッセージに付加します。メッセージの受信者も同じ計算を実行して、受信側のMACと、メッセージとともに受信したMACとを比較します。両方のMACが一致すれば、メッセージは本物です。

この基本モデルには欠点があります。つまり、不正な送信者が、傍受したメッセージを後で再現することによって、そのメッセージが本物と間違えられる可能性があるということです。MACの受信者によって選択されたランダムチャレンジをMACの計算に採り入れた場合は、単純な「反射攻撃」が成功する可能性はなくなります。図1は一般的な概念を示しています。チャレンジを長くするほど、起こり得る反射攻撃に予想されるすべてのレスポンスを記録することは困難になります。

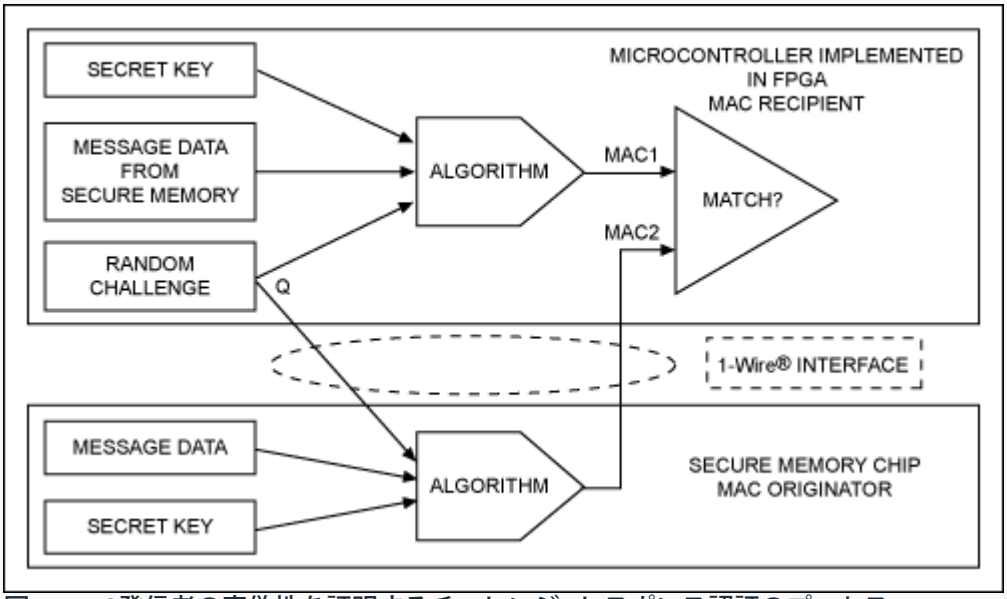

図*1. MAC*発信者の真偽性を証明するチャレンジ&レスポンス認証のプロセス

MAC発信者の真偽性を証明するため、MACの受信者は乱数を生成し、それをチャレンジとして発信者に送信します。次にMAC発信者は、秘密鍵、メッセージ、および受信者のチャレンジに基づいて新しいMACを計算する必要があります。発信者は、計算結果を受信者に返します。発信者がいずれのチャレンジについても有効なMACを生成することが可能であることが証明された場合は、発信者が秘密鍵を知っていることが明白であるため、本物と見なすことができます。このプロセスは専門用語で、チャレンジ&レスポンス認証と呼ばれています(図1を参照)。

MACを計算するために使用されるアルゴリズムは、Gost-Hash、HAS-160、HAVAL、MDC-2、MD2、MD4、MD5、RIPEMD、SHAファミリ、Tiger、およびWHIRLPOOLなど多数あります。徹底的に吟味されて国際的に認定された一方向のハッシュアルゴリズムがSHA-1です。これは、米国国立標準技術研究所(NIST)によって開発されました。SHA-1は、国際規格のISO/IEC 10118-3：2004に発展しています。

SHA-1アルゴリズムの基となる計算は、NISTのウェブサイト[2]を通じて公開されています。SHA-1アルゴリズムの顕著な特長を以下に示します。

- 不可逆性：MACに対応する入力を求めることは、計算上、実行不可能です。
- 衝突耐性：所定のMACを生成する複数の入力メッセージを見つけることは実質的に不可能です。
- 大きななだれ効果：入力がわずかに変化しても、MACの結果は大幅に変化します。

これらの理由によって、さらにはアルゴリズムが国際的に吟味されていることによって、SHA-1は、セキュアメモリのチャレンジ&レスポンス認証の優れた選択肢になります。

ハードウェアの実装

チャレンジ&レスポンス認証の方式は、SRAMベースのFPGAシステム設計の一部として、安価に実装することができます(図2)。この例では、セキュアメモリデバイスは1つのピンのみを使用して、双方向(オープンドレイン)通信用に構成されたFPGAのピンに接続されています。V_{DD}への抵抗接続によってセキュアメモリに給電し、オープンドレイン通信用のバイアスを設けています。マキシムのSHA-1エンジンを備えた1Kb保護1-Wire EEPROMのDS28E01は、この方式によく適合しています。このデバイスに含まれるのは、SHA-1エンジン、128バイトのユーザーメモリ、チップ内部の動作には使用可能だが外部からは読取り不能な秘密鍵、および固有で不変の識別番号です。

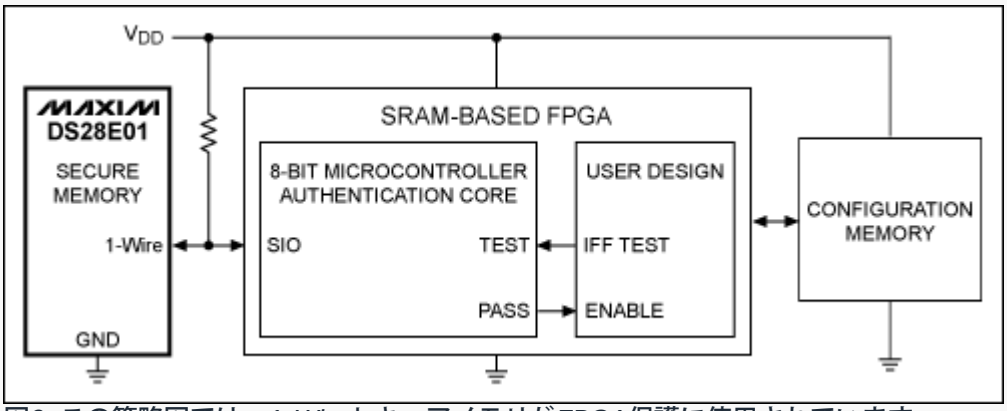

図*2.* この簡略図では、*1-Wire*セキュアメモリが*FPGA*保護に使用されています。

DS28E01の1-Wireインタフェースは、通信チャネルをチャレンジ&レスポンス認証用のFPGAのピン1つだけに減らしています。FPGAは多くの場合、I/Oピンに制限があるため、セキュリティソリューションの影響は最小限になります。別の実装は、FPGA上に実装したより一般的なI²CインタフェースとDS28CN01 (DS28E01のI²C同等品)を使用するか、あるいはSHA-1エンジンやその他の機能を小型のASICまたはCPLDに実装することで構築することができます。ただし、セキュリティがデバイスの唯一の機能であるような場合、ASICの手法ではコストが高くつく可能性があります。

DS28E01のセキュリティ機能を活用するには、FPGAで以下が実行可能である必要があります。

- チャレンジ用の乱数を生成する(オンチップの乱数生成器は通常、擬似乱数を生成しますが、これは真の乱数ほど安全ではありません)。
- 内部動作で使用可能な、ただし外部からは検出することができない秘密鍵を識別する。
- セキュアメモリと同様、秘密鍵、乱数、およびその他データを含むSHA-1のMACを計算する。
- FPGAに実装されたCPUのXOR機能を使用してバイト単位でデータバイトを比較する。

SHA-1のMACの計算の詳細については、Secure Hash Standard (セキュアハッシュ規格)[2]を確認してください。アプリケーションノート3675「R&D投資の保護—双方向の認証とセキュアなソフトによる機能の設定」では、セキュアメモリのアーキテクチャに関する技術的側面とそのセキュリティ概念について詳述しています。

マイクロコントローラのような機能は通常、主要なFPGAのベンダーから無償のマクロとして入手することができます。Xilinx®のマイクロコントローラ機能は192の論理セルを占有しています。これはSpartan®-3 XC3S50デバイスのちょうど11%になります。Altera®デバイスの同様のマイクロコントローラでは、850の論理素子を占有しています。これはEP2C5の18.5%に相当し、Cyclone® IIファミリの中でも最も小さな製品になります。

動作の仕組み

電圧が印加されると、FPGAはその構成メモリからFPGAそのものを設定します。これでFPGAのマイクロコントローラ機能が有効になり、敵味方識別(IFF)としても知られるチャレンジ&レスポンス認証を実行します。この識別では、以下の手順が必要となります。

- 乱数を生成して、チャレンジ(Q)としてセキュアメモリに送出します。
- 秘密鍵、チャレンジ、固有の識別番号、およびその他の固定データに基づいてSHA-1のMACを計算するようセキュアメモリに命令します。
- セキュアメモリが使用する同じ入力と定数、およびFPGAの秘密鍵に基づいてSHA-1のMAC、すなわち予測レスポンス(MAC1)を計算します。
- セキュアメモリで計算されたSHA-1のMAC (認証MACの読取り)をレスポンス(MAC2)と見なし、これを予測レスポンス(MAC1)と比較します。

MAC1とMAC2が一致すれば、セキュアメモリ環境で秘密鍵が認識されていると思われるため、FPGAはこの環境を「味方」と識別します。FPGAは通常動作に遷移して、構成コードで定義されたすべての機能を有効/実行します。しかし、MACが異なる場合は、環境は「敵」になります。この場合、FPGAは通常動作ではない、アプリケーション固有の動作を実行します。

このプロセスが安全である理由

SHA-1によって提供される固有のセキュリティに加えて、上記のIFF認証プロセスの主要なセキュリティ要素となるのは秘密鍵です。この秘密鍵をセキュアメモリまたはFPGAから読み取ることはできません。さらに、ビットストリームのデータはスクランブルがかけられているため、FPGA自体の設定時に構成ビットストリームを盗聴しても、秘密鍵を明らかにすることはできません。認証ステップをなくす目的で、ビットストリームをリバースエンジニアリングして設計内容を把握することは、ビットストリームのサイズから非常に時間がかかるため、極めて困難な作業となります。

もう1つの重要なセキュリティ要素は、チャレンジのランダム性です。予測可能なチャレンジ(すなわち、定数)では、レスポンスが予測可能になります。これをいったん記録して、セキュアメモリをエミュレートするマイクロコントローラによって後で再生することができます。予測可能チャレンジによって、マイクロコントローラは、環境を味方と見なすようFPGAを効果的に欺くことができます。IFF手法のチャレンジのランダム性によって、この懸念が緩和されます。

各セキュアメモリ内の秘密鍵がデバイス固有(すなわち、マスタシークレットから計算された個別秘密鍵、SHA-1メモリ独自の識別番号、およびアプリケーション固有の定数)の場合は、セキュリティをさらに向上させることができます。個別鍵が公開された場合は、システム全体のセキュリティではなく、1つのデバイスのみが影響を受けることになります。個別秘密鍵をサポートするには、FPGAはマスタシークレットを認識して、予測レスポンスを計算する前に、まず1-Wire SHA-1メモリチップの秘密鍵を計算する必要があります。

物流面について

ビルドする単位ごとに、設計側(OEM)は、1つの事前に正しく設定したセキュアメモリを、FPGAの組込み製品を製造する委託製造業者(CM)に供給する必要があります。この1対1の関係によって、CMがビルド可能な認可ユニットの数が制限されます。CMがセキュアメモリを改竄する(たとえば、メモリが正しく設定されていないため、追加メモリが必要であると要求する)のを防ぐためにOEMは秘密鍵を書込み保護することが望まれます。

1-Wire EEPROMデータメモリが書込み保護されていない場合でも、そのセキュリティを懸念する必要はありません。設計上、秘密鍵を知っている人しかこのメモリデータを変更することはできません。歓迎される副次的作用としては、この特性によってアプリケーション設計者は、ソフトによる機能管理(FPGAがSHA-1のセキュアメモリから読み取るデータに応じて機能を有効または無効にすることができる)を実装することができます。

CMに出荷する前に、OEMがメモリデバイスを事前設定するのは、必ずしも現実的ではありません。この状況に対処するために、セキュアメモリの製造業者はSHA-1シークレットとEEPROMアレイの事前設定サービスをOEMに提供します。マキシムは、OEMの入力に応じてセキュアメモリデバイスを工場で登録して設定し、さらにCMに直接出荷するというサービスを提供しています。このサービスの主な利点は、以下のとおりです。

- OEMが秘密鍵をCMに開示する必要がない。
- OEMが事前設定のシステムを実装する必要がない。
- OEMが認可したサードパーティのみが、登録されたデバイスにアクセス可能。
- ベンダーは、OEMの監査の必要に応じて、出荷された数量の記録を保管する。

コンセプトの実証

このアプリケーションノートに記載したFPGAのセキュリティ手法は、AlteraとXilinxの製品でテストされています。Alteraは、その白書「An FPGA Design Security Solution Using a Secure Memory Device」の中で次のように結論をまとめています。「このFPGA設計セキュリティのIFFソリューションは、たとえ構成データビットストリームがキャプチャされたとしても、AlteraのFPGA設計をクローンから保護することができます。ユーザー設計は、FPGAとセキュアメモリの両方のハッシュアルゴリズムの計算が一致するまで無効にされます。この設計セキュリティのソリューションは、FPGA設計者のIPを保護します」[3]

同様に、Xilinxは、そのアプリケーションノートXAPP780で以下のように述べています。「システムのセキュリティは、基本的に秘密鍵の秘密性と安全な環境で秘密鍵を導入することに基づいています。秘密鍵を除いた、この全リファレンスデザインは、広く認められているケルクホフスの原理によって永続的に公然です。このアプリケーションノートで示したプログラミングと認証の簡単なインタフェースによって、極めて容易にこのコピープロテクト方式を実装することができます」[4] (フランドルの言語学者アウグスト・ケルクホフスは、軍用の暗号化技術に関する革新的な論文の中で、セキュリティは秘匿に頼るのではなく、鍵の強度に依存すべきだと主張しました。彼は、セキュリティの侵害が発生した場合、交換する必要があるのはシステム全体ではなく鍵のみであると主張しました。)

結論

IPの著作権侵害に対する保護では、DS28E01などの低コストのチップを1つ追加し、FPGAの構成コードを更新する必要があります。1-Wireインタフェースのおかげで、セキュリティの目的で使用するFPGAピンは1つのみです。これ以上のFPGAピンが利用可能な場合は、1-WireバージョンのI²Cバージョンを使用することができます。この場合も、FPGAの構成パターンの変更と、組込みマイクロコントローラの制御ソフトウェアの変更が必要となります。

固定の秘密鍵または計算した秘密鍵のいずれかと、アプリケーション固有のデータを事前に設定したセキュアメモリを発注することができます。次に、事前に設定した部品はOEMにのみ出荷されます。場合によっては認可されたCMに出荷されることもあります。CMは、提供された事前設定済みの部品と同じユニット数だけビルドすることができます。

付録A

FPGAの技術的なオプションと問題

5つの主流なFPGAベンダーであるActel®、Altera、Lattice Semiconductor Corporation®、QuickLogic®、およびXilinxの市場占有率をすべて合わせると、約98%になります。残りは、FPGAと同様の機能を提供するいくつかの特殊分野のメーカーが占めています。

これらの企業はすべて自社工場を持たないで、ウェハの生産を台湾、日本、シンガポール、またはドイツのウェハファウンドリに依存しています。これによって企業は、SRAM能力を備えた高速ロジックや高密度フローティングゲートメモリアレイなどのファウンドリの技術を利用することができます。標準のプロセスフローを活用することで、FPGA企業は製造コストを低く保つことができます。

ただし、ActelとQuickLogicは、ファウンドリと協力して独自のアンチヒューズ技術をプロセスフローに組み込んでいます。アンチヒューズ技術によって、優れたセキュリティ、小さなセルサイズ、放射線耐性、さらに当然ですが不揮発性など、固有の利点がいくつか得られます。ただし、アンチヒューズ技術には、いまだに1つの制約があります。フラッシュベースのデバイスとは異なり、いったんセルが構成されると、(「ヒューズ」という名前が示すとおり)再構成することはできません。さらに、アンチヒューズのプロセスは、SRAMベースのFPGAよりも複雑ですが、セルが小さくなるため、結果としてチップが小型化され、同等のロジック容量で低コストになります。

長い間、SRAM技術とフラッシュ技術は相互に排他的で、1つのチップに組み込むのは容易ではありませんでした。しかし、市場の需要によって半導体技術が近年進展したおかげで、このシナリオは変わりました。小規模の競合他社(Actel、Lattice Semiconductor、およびQuickLogic)は現在、フラッシュベースのシングルチップFPGA (それぞれ、ProASICファミリ、Lattice®XP2ファミリ、およびPolarProファミリ)を提供しています。

たとえば、AlteraはフラッシュベースのCPLD (MAXII)を2004年から提供していますが、現在この技術はFPGAには使用されていません。Xilinxは、フラッシュベースのFPGAであるSpartan-3ANの生産ラインを2007年初めに導入して、単一のパッケージに2チップ(FPGAとフラッシュメモリ)を搭載しています。ただし、XilinxにはモノリシックフラッシュベースのFPGA製品[5]はありません。フラッシュの格納を共統合することができる機能によって、外付けの構成メモリを使用するSRAMベースのFPGAに比べて、IPセキュリティが向上します。シングルチップに共統合することによって、誰かがメモリと構成可能なロジックアレイとの間のデータ転送をのぞき見ることは困難になります。

これらのセキュリティ要件のすべてを満足する手法は、FPGAとセキュアメモリとの間でチャレンジ&レスポンスのデータ交換(認証)を必要とします。セキュアメモリチップは1980年後期の発明です。ヨーロッパでは最初に公衆電話のカード用に、1990年代には銀行のカード用によく使われました。現在、セキュアメモリチップは、GSMの携帯電話の重要な構成要素(すなわち、SIMカード)になっています。セキュアチップカードとホストシステムの間の通信規格として普及しているのが、I²Cシリアルバスです。

セキュアメモリは銀行または電話アプリケーション向けの特注品であり、汎用的な用途には有効でないと考えられていました。これは、Dallas Semiconductor (現在のMaxim Integrated Products)が、SHA-1のハッシュアルゴリズムを組み込んだデバイスを導入した2000年頃に変化しました。第1世代はDS2432で、その後に強化されたDS28E01が続きました。これらのデバイスは、1-Wireインタフェースを使用しており、これは通信と電源の両方で役立ちます。2007年に発表されたDS28CN01は、I²Cインタフェースを使用していますが、それ以外はDS28E01と同じです。

IPセキュリティに関して検討すべきもう1つの側面はファウンドリの誠実性です。これはファウンドリが多くの場合、FPGA設計に関する秘密の情報を所有しているためです。機密情報が間違った手に渡らないように防御するには、信頼または厳しい管理と監視が必要になります。これについては、海外のファウンドリよりも国内のファウンドリの方が、実現が容易であると思われますが、これまでのところ、主要なファウンドリは、設計の詳細を保護するための優れた誠実性を示しています。

参考文献

1. Saar Drimer著「Volatile FPGA design security — a survey」、作業進行中、http://www.cl.cam.ac.uk/~sd410/papers/fpga_security.pdf
2. Secure Hash Standard (セキュアハッシュ規格)、http://www.itl.nist.gov/fipspubs/fip180-1.htm
3. Alteraの白書01033、「An FPGA Design Security Solution Using a Secure Memory Device」、http://www.altera.com/literature/wp/wp-01033.pdf
4. XilinxのアプリケーションノートXAPP780「FPGA IFF Copy Protection Using Dallas Semiconductor/Maxim DS2432 Secure EEPROMs」、http://www.xilinx.com/support/documentation/application_notes/xapp780.pdf
5. FPGA and Structured ASIC Journal、2007年2月27日発行、「Short Stack with Syrup」、www.techfocusmedia.net/fpgajournal/feature_articles/20070227_stack

関連製品

DS28CN01	SHA-1エンジン付き、1kビットI²C/SMBus EEPROM	
DS28E01-100	SHA-1エンジン付き、1Kb保護1-Wire EEPROM	
DS28E02	1.8V動作、1-Wire SHA-1認証1Kb EEPROM	
DS28E10	1-Wire SHA-1認証用IC	-- 無料サンプル

自動アップデート
お客様が関心のある分野でアプリケーションノートが新規に掲載された際に自動通知Eメールの受信を希望する場合は、EE-Mail™にご登録ください。

アプリケーションノー4594: japan.maxim-ic.com/an4594
その他の情報
テクニカルサポート：japan.maxim-ic.com/support
サンプル請求：japan.maxim-ic.com/samples
その他の質問およびコメント：japan.maxim-ic.com/contact

AN4594, AN 4594, APP4594, Appnote4594, Appnote 4594