



[Maxim](#) > [Design Support](#) > [Technical Documents](#) > [Application Notes](#) > [1-Wire® Devices](#) > APP 3826

[Maxim](#) > [Design Support](#) > [Technical Documents](#) > [Application Notes](#) > [Security and Authentication](#) > APP 3826

Keywords: xilinx, fpga, sha, sha-1, SHA, SHA-1, board, identification, feature, management, IFF, authentication, secure, memories, copy, protection, ds2432, ibutton, 1-wire, ds28e01-100, virtex, spartan, hash, secret, key, XC3S200, coolrunner

#### APPLICATION NOTE 3826

# Xilinx® FPGA IFF Copy Protection with 1-Wire® SHA-1 Secure Memories

By: Bernhard Linke, Principal Member Technical Staff

Jun 21, 2006

*Abstract: This article explains how the Identification Friend or Foe (IFF) authentication concept in conjunction with secure memories provides secure control and copy protection of FPGA designs. This IFF approach also enables the implementation of soft-feature management and board identification. The FPGAs referenced in this application note are all from Xilinx, Inc.*

## Motivation

The development of electronic products, including the configuration code for embedded FPGAs, is costly. Design and configuration data constitute intellectual property that needs to be protected against unauthorized copies in the form of clones. There are several ways to protect such intellectual property. High-end FPGAs, such as the Xilinx Virtex-II® and Virtex-4® devices, support encryption of the configuration bitstream, a process in which a copy of the bitstream works only with FPGAs using the same secret key. This encryption approach is not, however, viable for more cost-sensitive applications. A suitable alternative to prevent cloning is authentication that uses the Identification Friend or Foe (IFF) concept. This IFF method is applicable to all FPGA families, including the low-cost Xilinx Spartan®-3 and Spartan-6 series.

## Preconditions

The IFF concept requires that the FPGA design be enhanced by a secure memory chip that the FPGA communicates with to perform an authentication. **Figure 1** shows a simplified schematic of this approach.

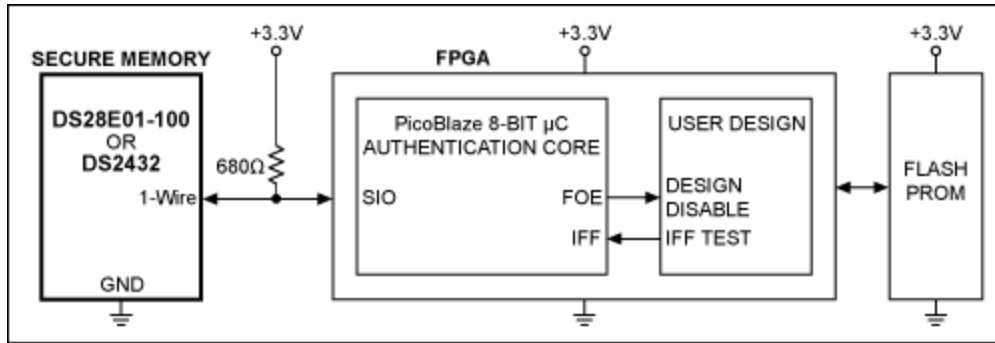


Figure 1. Simplified schematic of a Maxim 1-Wire memory device providing secure control and protection for an FPGA.

The secure memory chip needs to meet the following requirements:

- Contain a secret key that can be used for chip-internal operations, but cannot be read from outside.
- Contain a unique, not changeable, ID (identification number). This ID can be used to compute a device-specific secret key.
- Can compute a HASH that involves the secret key, a random number acting as a challenge, the unique ID number, and additional data (constants).

To provide the desired level of security, the HASH algorithm:

- Must be irreversible—making it computationally infeasible to determine the input corresponding to a HASH result.
- Must be collision-resistant—making it impractical to find another input message that produces a given HASH result.
- Must have a high avalanche effect—causing any change in input to produce a significant change in the HASH result.

A thoroughly scrutinized and internationally certified one-way hash algorithm that meets these criteria is the Secure Hash Algorithm (SHA-1), which was developed by the National Institute of Standards and Technology (NIST). SHA-1 has evolved into the international standard ISO/IEC 10118-3:2004. The math behind the algorithm is publicly available through the NIST website.

Maxim's secure memories that implement the SHA-1 algorithm include the [DS28E01-100](#) and [DS2432](#). The 1-Wire interface of these devices makes them particularly well-suited for this application because they require only a single FPGA pin for operation.

To take advantage of the memory chip's security features, the FPGA must be able to do the following:

- Generate random numbers.
- Know a secret key that can be used for internal operations, but cannot be discovered from outside.
- Compute a HASH that involves the secret key, a random number and additional data, just like the secure memory.
- Compare HASH results.

This FPGA functionality requires the capabilities of a microcontroller, such as the PicoBlaze™, which is available from Xilinx as a free macro optimized for the Spartan-3, Spartan-6, Virtex-II, Virtex-II Pro®,

and Virtex-4 FPGAs and CoolRunner®-II CPLDs. PicoBlaze occupies 192 logic cells, which represents just 5% of a Spartan-3 XC3S200 device.

## How the Technique Works

When power is applied, the FPGA configures itself from the Flash PROM. Now the FPGA's microcontroller function becomes active and performs the authentication, which consists of the following steps:

1. Generate a random number and send it as a challenge (**Q**) to the secure memory.
2. Instruct the secure memory to compute a HASH based on its secret key, the challenge Q, its unique ID, and other fixed data.
3. Compute a HASH, i.e., the expected response (**E**), based on the same input and constants used by the secure memory and the FPGA's secret key.
4. Take the HASH computed by the secure memory (Read Authentication MAC) as response (**A**) and compare it to the expected response (**E**).

If A and E match, the microcontroller identifies the circuit is a 'Friend', because it apparently knows the secret key. The FPGA transitions to normal operation, enabling/performing all the functions defined in its configuration code. If A and E are different, however, the circuit must be a 'Foe'. In this case the FPGA takes application-specific actions other than normal operation.

## Why the Process Is Safe

For every unit to be built, the owner of the design (OEM) must provide one properly preprogrammed DS28E01-100 or DS2432 to the party that manufactures the product with the embedded FPGA (contract manufacturer, or CM). This one-to-one relationship limits the number of authorized units that the CM can build. To prevent the CM from tampering with the secure memory (perhaps claiming that more memories are needed because they are not programmed properly), it is advisable to write-protect the secret key. There is no need to worry about the security of the 1-Wire EEPROM data memory, even if it is not write-protected. By design, this memory data can only be changed by individuals who know the secret key. As a welcome side-effect, this characteristic enables the application designer to implement soft-feature management: the FPGA can enable/disable functions depending on data that it reads from the SHA-1 secured memory.

It is not always convenient or practical for the OEM to preprogram memory devices before delivery to the CM. To address this situation, Maxim also provides a SHA-1 secret and EEPROM-array data preprogramming service for the OEM. With this service, 1-Wire memory devices are registered and configured at the Maxim factory according to OEM input. Then Maxim ships directly to the CM. Key benefits of this service include:

1. Eliminates the need for the OEM to divulge the SHA-1 secret to the CM.
2. Eliminates the need for the OEM to implement a preprogramming system.
3. Only OEM-authorized third parties have access to registered devices.
4. Maxim maintains records of shipped unit quantities if needed for OEM auditing purposes.

Besides the inherent safety provided by SHA-1, the principal safety element for the above type of IFF authentication is the **secret key**, which is neither readable from the secure memory nor from the FPGA.

The secret key, furthermore, cannot be discovered by eavesdropping on the configuration bitstream when the FPGA configures itself. Reverse-engineering the bitstream to determine the design with the intent of removing the authentication step is a prohibitively difficult task, analogous to trying to determine the C++ code behind a large Windows® application starting only with the executable code.

Another critical security component is the **randomness** of the challenge Q. A predictable challenge (i.e., a constant) causes a predictable response, which can be recorded once and then replayed later by a microcontroller in place of the secure memory. With such a predictable challenge the microcontroller can effectively deceive the FPGA to consider the design as a Friend. The randomness of the challenge in this IFF approach alleviates this concern.

Security can be improved further if the secret key in each secure memory is device-specific, i.e., an **individual secret** key computed from a master secret, the memory's unique ID, and application-specific constants. Should an individual key become public, only a single device is affected, not the security of the system. To support individual secret keys, the FPGA needs to know the master secret and compute the memory chip's secret key first before computing the expected response.

## More Details

The information provided here shows the general concept of IFF authentication implemented through FPGA and a secure memory on a fairly abstract level. For more details, please refer to the following documents:

- Tutorial 3675, "[Protecting the R&D Investment—Two-Way Authentication and Secure Soft-Feature Settings](#)" (security and system aspects).
- Tutorial 1201, "[White Paper 8: 1-Wire SHA-1 Overview](#)" (HASH, SHA-1, trusted token, transaction details, further links to information and product data sheets).
- XILINX Application note XAPP780, "[FPGA IFF Copy Protection Using Dallas Semiconductor/Maxim DS2432 Secure EEPROMs](#)" (successful implementation of the concept, FPGA implementation details).
- [Secure Hash Standard](#), Federal Information Processing Standards Publication 180-1.
- Application note 178, "[Printed Circuit Board Identification Using 1-Wire Products](#)" (concept, defining data, external probe point).
- Application note 3522, "[White Paper 9: Are SHA-1 Devices Still Secure Enough?](#)" (collision-resistancy, attack on the SHA-1 digest).
- Reference design 4477, "[Reference Design of a 1-Wire® Bidirectional Voltage-Level Translator for 1.8V to 5V](#)" (interfacing a 1.8V 1-Wire master to a 5V 1-Wire slave).
- Application Note 4594, "[Protect Your FPGA Against Piracy: Cost-Effective Authentication Scheme Protects IP in SRAM-Based FPGA Designs](#)" (a newer, extended version of application note 3826).

## Conclusion

Protection against unauthorized cloning of intellectual property using the well-known Identification Friend or Foe (IFF) concept requires adding just one low-cost chip and updates to the FPGA configuration code. The secure memory chip and its in-circuit programmability enable the designer to implement soft-feature management, changing configurations remotely rather than sending a technician. Secure memory space not used for feature management is thus available for board identification purposes.

1-Wire is a registered trademark of Maxim Integrated Products, Inc.  
CoolRunner is a registered trademark of Xilinx, Inc.  
PicoBlaze is a trademark of Xilinx, Inc.  
Spartan is a registered trademark of Xilinx, Inc.  
Virtex II is a registered trademark of Xilinx, Inc.  
Virtex II Pro is a registered trademark of Xilinx, Inc.  
Virtex-4 is a registered trademark of Xilinx, Inc.  
Windows is a registered trademark and registered service mark of Microsoft Corporation.  
Xilinx is a registered trademark and registered service mark of Xilinx, Inc.

Related Parts		
<a href="#">DS2432</a>	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	<a href="#">Free Samples</a>
<a href="#">DS28E01-100</a>	1Kb Protected 1-Wire EEPROM with SHA-1 Engine	<a href="#">Free Samples</a>
<a href="#">DS28E02</a>	1-Wire SHA-1 Authenticated 1Kb EEPROM with 1.8V Operation	<a href="#">Free Samples</a>
<a href="#">DS28E10</a>	1-Wire SHA-1 Authenticator	<a href="#">Free Samples</a>

---

#### More Information

For Technical Support: <http://www.maximintegrated.com/support>

For Samples: <http://www.maximintegrated.com/samples>

Other Questions and Comments: <http://www.maximintegrated.com/contact>

---

Application Note 3826: <http://www.maximintegrated.com/an3826>

APPLICATION NOTE 3826, AN3826, AN 3826, APP3826, Appnote3826, Appnote 3826

Copyright © by Maxim Integrated Products

Additional Legal Notices: <http://www.maximintegrated.com/legal>