

## An FPGA Design Security Solution Using a Secure Memory Device

### Introduction

FPGA designs are vulnerable to design theft because configuration bitstreams can be easily captured and copied. FPGAs are more vulnerable to cloning of the entire design rather than to intellectual property (IP) theft, since extracting IP from the bitstream is nearly impossible. In order to protect the configuration bitstream, some FPGAs are now capable of encrypting the bitstream. However, there is an additional cost for FPGAs that do not offer embedded bitstream encryption to encrypt the configuration bitstream due to the additional step of programming the encryption key in the FPGA during manufacturing. For high-volume applications, using a security companion chip is much more cost effective.

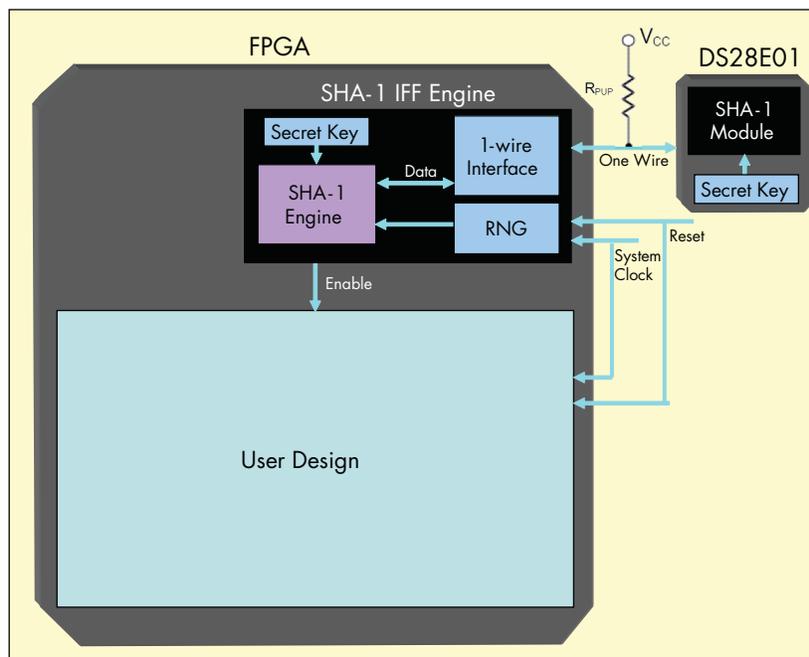
This document provides a solution to help protect FPGA designs from being cloned. Using the “identification, friend or foe” (IFF) design security approach, this solution disables the design within the FPGA until the hash algorithm computation matches in both the FPGA and a secure memory device, so the design remains secure even if the configuration data bitstream is captured. In this solution, the secure memory device is use as a security companion chip for the FPGA.

### Implementation

The IFF concept requires a security companion device to compute the hash algorithm. Dallas Semiconductor’s secure memory device DS28E01 combines a 1024-bit EEPROM with a challenge-and-response authentication security with the ISO/IEC 10118-3 Secure Hash Algorithm (SHA-1). DS28E01 is a one-wire interface device, so this solution only needs one FPGA I/O pin. The secure memory device needs a pull-up resistance connected to the one-wire I/O pin. (For the electrical specification of DS28E01, contact Dallas Semiconductor.)

Figure 1 shows the top-level block diagram of a design-security reference design using the IFF concept. The secure memory device computes a hash algorithm with the SHA-1 engine based on a secret key stored in the secure memory, a random number generated from the FPGA, and a unique ID in the secure memory.

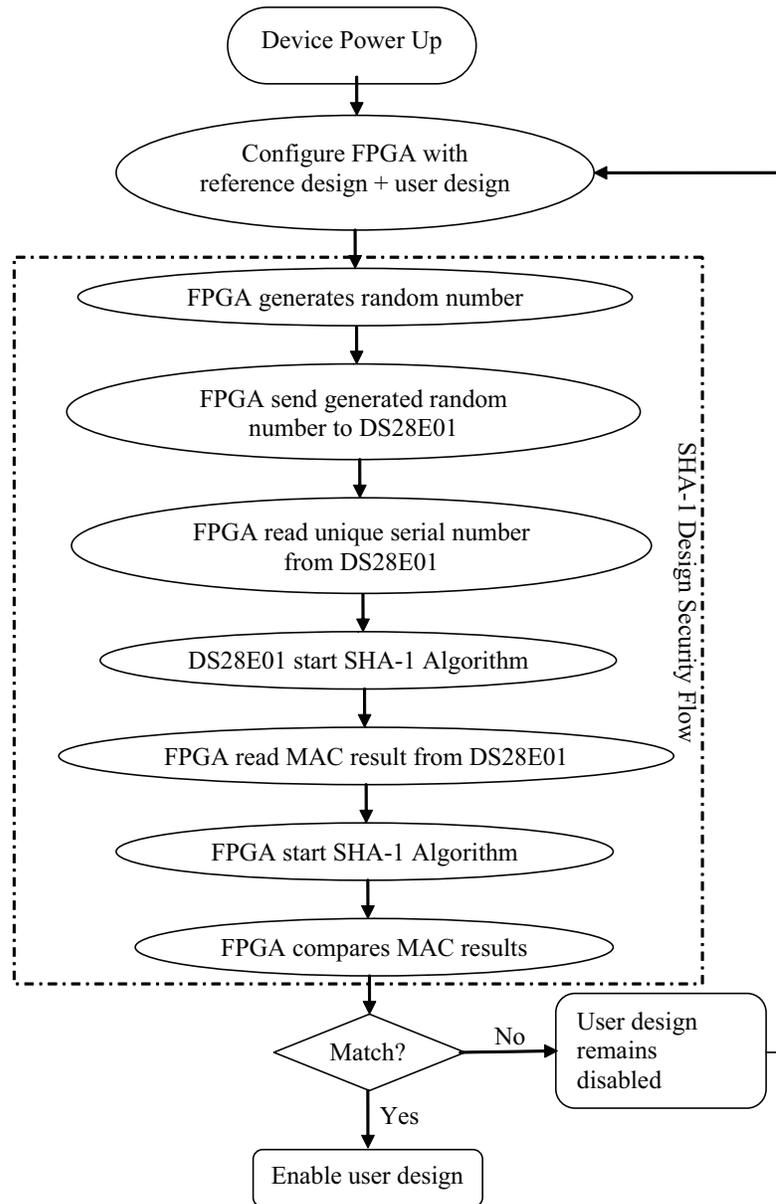
Figure 1. Design Security Solution Using the IFF Concept



To use an FPGA requires a SHA-1 IFF module that knows the matching secret key in the secure memory and is able to compute SHA-1 algorithm based on the same inputs for the SHA-1 engine in the secure memory. After the FPGA is configured, the user design will not be enabled. This SHA-1 IFF module only enables the user design when the hash computations from both the secure memory and the FPGA match.

Once the system is powered up, and after the FPGA is configured with the user design embedded with the SHA-1 IFF reference design, the FPGA generates a random number and sends it to the secure memory. The FPGA reads the computed 160-bit computed message authentication code (MAC) result from the DS28E01 and compares this with the FPGA SHA-1 IFF engine MAC result. The SHA-1 IFF block enables the user design if the MAC results match, and disables it if does not. [Figure 2](#) shows design security flow using IFF concept.

Figure 2. Design Security Flow Using IFF Concept

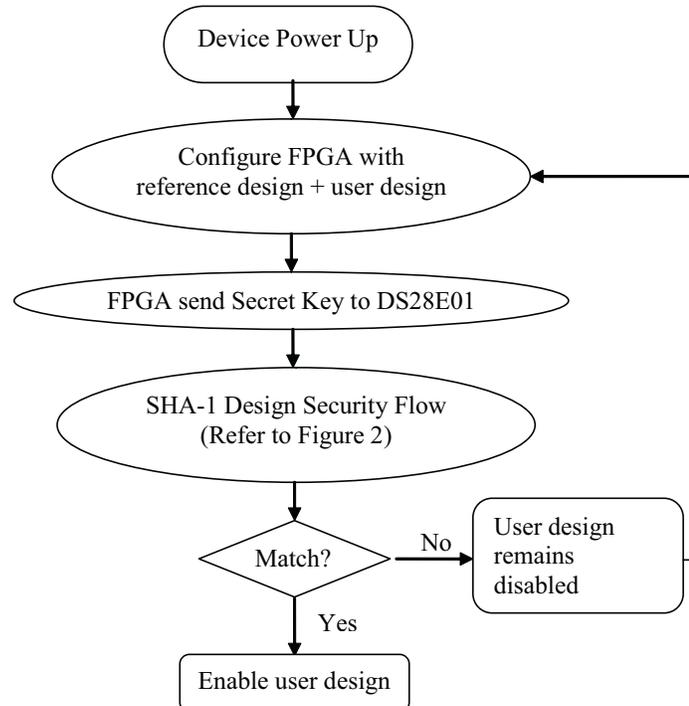


The reference design provides an additional option for users to program the DS28E01 device from the FPGA. This allows the FPGA to send the secret key to the secure memory after the FPGA is configured, and should be used during manufacturing in a secure location.

 This option should only be used for the first time to program the secure memory. For programming of the DS28E01 device during volume programming, please contact Dallas Semiconductor.

Figure 3 shows the design security flow for programming the DS28E01 via the FPGA.

Figure 3. Design Security Flow When DS28E01 Is Programmed via FPGA



Once the user design is enabled, the SHA-1 IFF engine block is turned off to reduce power consumption. Users can connect external logics or a state machine to reset the SHA-1 IFF engine to enable. The SHA-1 IFF engine block continuously computes and checks SHA-1 algorithm once the enable signal goes high.

## Design Building Blocks

The reference design for this solution consists of three main modules:

- *SHA-1 engine*: This module computes the SHA-1 algorithm and performs security authentication. This block compares the MAC result with the data received from the secure memory through the one-wire interface. It enables the user design only when its hash computation results matches with the hash computation of the SHA-1 engine in the secure memory.
- *Random number generator (RNG)*: The RNG generates a random number pattern to the SHA-1 engine module when the reset signal is asserted to this module. The SHA-1 IFF reference design uses an 8-bit RNG block. The SHA-1 engine block then processes this 8-bit random number into 40-bit random data for the hash algorithm computation.
- *One-wire interface*: This module allows data transfer to and from the reference design in the FPGA and the secure memory.

## User Design Block

The typical  $F_{MAX}$  for the SHA-1 IFF engine system clock frequency is 100 MHz or below. The user must input the SHA-1 IFF engine frequency into the reference design, thus ensuring that the data is transmitted and received between the FPGA and secure memory devices correctly. The user may have different clocks supplied to the SHA-1 IFF engine and the user design.

## Security of the Solution

The configuration data bitstream of an FPGA can be captured when it is being transmitted from an external memory device to the FPGA at power up. The FPGA design then can be copied by configuring another FPGA device using the captured configuration data bitstream. This solution helps to protect the user design as it insures that the clone device does not function. The user design in the FPGA device remains disabled without the correct secret key and hash algorithm computation.

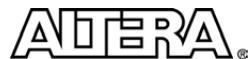
In order to clone the design to another FPGA design, the secret key and the secure memory device unique ID must be cloned. This is difficult as the DS28E01's secret key cannot be read out and reverse-engineering the SHA-1 algorithm from the MAC result to determine the key is impossible.

## Conclusion

This FPGA design security IFF solution protects Altera® FPGA designs from being cloned even if the configuration data bitstream is captured. The user design remains disabled until the hash algorithm computation in both the FPGA and the secure memory match. This design security solution protects FPGA designer's IP.

## Further Information

- Dallas Semiconductor/Maxim Integrated Products:  
[www.maxim-ic.com](http://www.maxim-ic.com)



101 Innovation Drive  
San Jose, CA 95134  
[www.altera.com](http://www.altera.com)

Copyright © 2007 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.