Zeroization

Table of Contents

Introduction

Zeroization is the practice of erasing sensitive parameters to prevent their disclosure if the system is attacked, or is at an increased risk of unauthorized access. Data security is an important aspect in many applications where high-value assets are at risk. The designer must protect the data to ensure the integrity and confidentiality of the system. Historically, this issue has been a major concern in financial and military applications, but it can also play a key role in many consumer, commercial, and industrial applications where unique or highly sensitive content is embedded in the data stream that passes through the system, for example in digital rights management applications; or where it is desired that the intellectual property comprising the design itself is to be kept confidential. The Zeroization capability provides the designer with the ability to protect the data in case a tamper event is detected.

The standard erase time of Microsemi® flash based FPGAs may be longer than desired for erasing the design data in data security applications. This application note describes a fast FPGA erase algorithm which will erase Microsemi flash based FPGAs in less than 5 seconds using JTAG. It also supports erasing of the flash read only memory (FROM). The design consists of a CoreABC APB bus controller and a custom JTAG controller. If the tampering condition is triggered, the CoreABC microcontroller executes the fast erase microcode to erase an attached Microsemi FPGA through the JTAG interface.

Design Overview

This design example demonstrates the Zeroization of Microsemi FPGAs: IGLOO, ProASIC3, Fusion, and SmartFusion® customizable system-on-chip (cSoC) devices. The design is implemented to erase the flash based FPGA within 5 seconds. The FPGA fabric alone or the fabric and internal FlashROM can be erased with this algorithm. The algorithm runs on a host FPGA and uses the implemented custom JTAG interface to erase the design in the targeted device as depicted in Figure 1 on page 2.

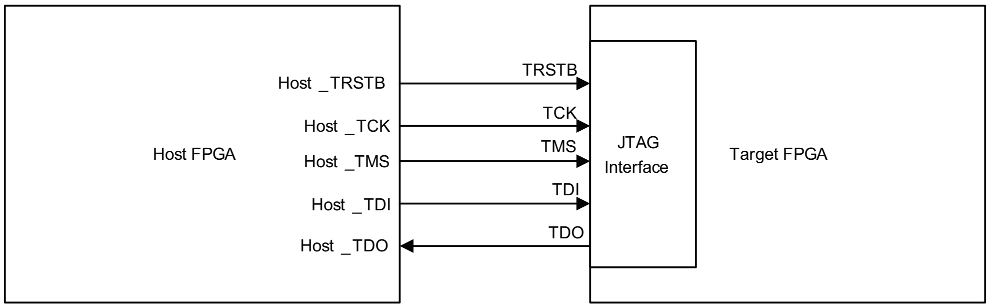

***Figure 1* • System Level Connectivity**

The standard erase process can take up to 2 minutes on a large device. This process places a high voltage on the array, forcing all bits to ‘1’. The high voltage must be sustained for relatively long periods to drive all cells to a reliable ‘1’. Lab tests have shown that an erase can also be achieved by ‘pushing’ the flash cells back and forth between 1 and 0 for a number of iterations. Setting the cells to ‘1’ is done by placing the device into erase, and setting the bits to 0 is done by programming all bits in the array to ‘0’. The difference here is that the device is erased for only 1.5 seconds. This short time does not fully erase the device. So, we follow erase with a procedure that pushes all the bits to ‘0’. We repeat this sequence 3 times to fully destroy the design. Testing and statistical analysis show that 3 iterations remove all evidence of a design from the flash cells.

The design also has status I/Os that indicate a successful completion of the Zeroization technique, or a failure in erasing the target device.

The design consumes approximately 2000-3000 tiles and 1 RAM block. The exact tile count is a function of the device targeted for erase and FlashROM erase requirement. Figure 1 shows the system level connectivity of the Zeroization security system.

Design Description

The design consists of a CoreABC micro sequencer coupled to custom JTAG control logic through a CoreAPB3 bus interface. The CoreABC controller executes the fast erase microcode algorithm that generates read/write APB bus accesses to the JTAG control logic. These accesses are processed by the custom JTAG control logic to generate JTAG bus cycles to the target FPGA. The JTAG sequences execute the fast erase algorithm on the target FPGA.

The design requires only 2 inputs: Erase and SYSCLK. SYSCLK is the system clock of this design. It also becomes the TCK to the target device. The frequency should not exceed 20 MHz, which is the maximum frequency of Microsemi FPGA JTAG interfaces. The Erase input triggers the controller to execute the fast erase algorithm. Figure 2 on page 3 shows the block diagram of the design implementation.

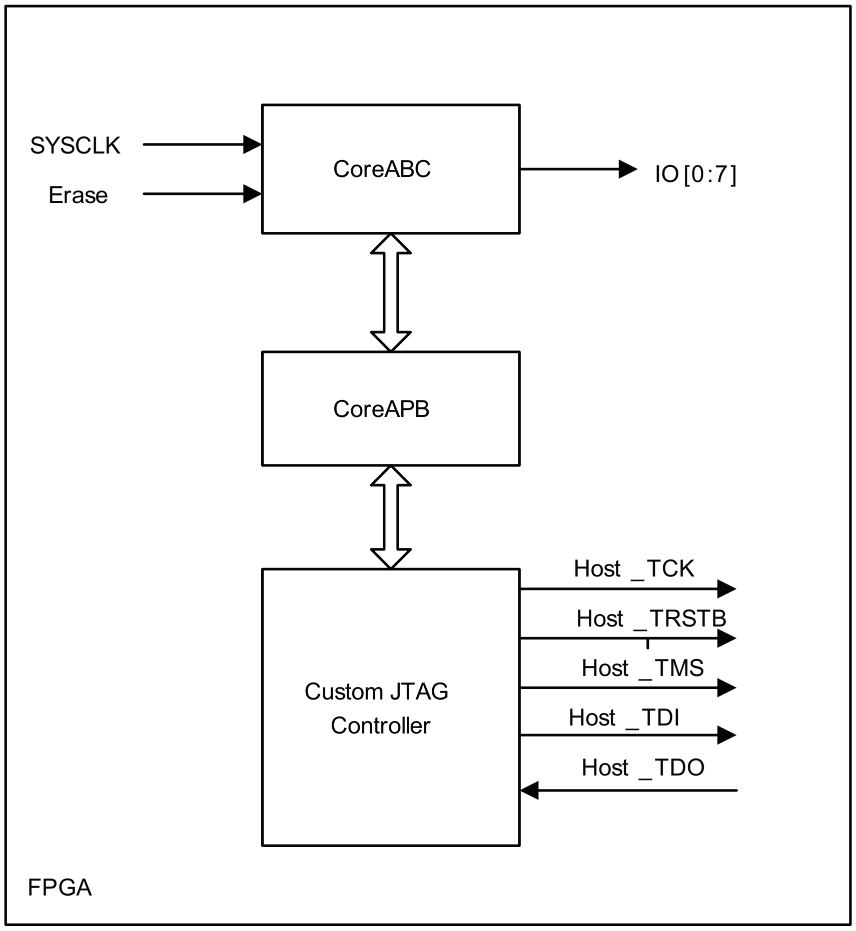

***Figure 2* • Block Diagram of Design Implementation**

The fast erase algorithm running on the CoreABC microcontroller requires target FPGA device parameters such as device family, size, and number of segments. You need to get these device specific parameters from the Microsemi factory. The CoreABC microcode needs to be edited with the target device specific information to perform Zeroization. Refer to "Appendix B - Device Parameters in Microcode" on page 6 for the modifications required in the CoreABC microcode.

An AES encrypted ‘zeros’ FlashROM data image corresponding to an all zeros plain text image is required in the microcode that is used to clear the FlashROM to all zeros. This data stream is a function of the customer AES key and should be customer generated. This data can be obtained by the following procedure:

1. In the FlashROM GUI, set all cells to ’0’. Make sure you select ‘Modifiable’ in the drop-down menu.
2. Create a programming STAPL file from this design. In Flash point, select ‘programming previously secured device’ and ‘high’ security setting. This requires the AES key but not the passkey. The resulting STAPL file will contain a 2048 bit array constant called FROMSTREAM[2048]. This data needs to be entered into the microcode.

To make this design highly reliable, the CoreABC microcode is synthesized in FPGA gates instead of RAM. With the logic design and microcode implemented in FPGA gates there is no danger of single event upset (SEU) corrupting the code. The resulting design reliability is similar to a state machine. One RAM block is required for the design and is used as a temporary stack for the CoreABC microcontroller. The data in this stack is valid only during the actual erase sequence, which is less than 5 seconds, resulting in a negligible effect on SEU reliability numbers.

The design is compact in size (low tile count). The tile count for this design ranges from 2000-3000 tiles, which can fit into a mid-sized IGLOO class device (AGL125). This design can only support a single device on the JTAG port. JTAG chains and multi-device erase support is not available.

Implementation

The design is implemented on the SmartFusion cSoC device (the Host FPGA) for erasing a Fusion AFS600 device (the Target FPGA). Figure 3 shows the SmartDesign implementation. The FAB_CLK is configured to 20 MHZ and is connected to the PCLK pin of the CoreABC block. The NSYSRESET pin of the CoreABC block is mapped to the SW1 pin on the A2F-EVAL-KIT or the A2F500-DEV-KIT and is used as an erase input. The custom JTAG controller is implemented and connected to the CoreABC controller through a CoreAPB3 bus. The custom JTAG controller pins are brought onto J22 of the SmartFusion Evaluation Kit Board or J13 of the SmartFusion Development Kit Board. The CoreABC I/Os are connected to the LEDs of the A2F-EVAL-KIT or the A2F500-DEV-KIT to indicate the Zeroization status.

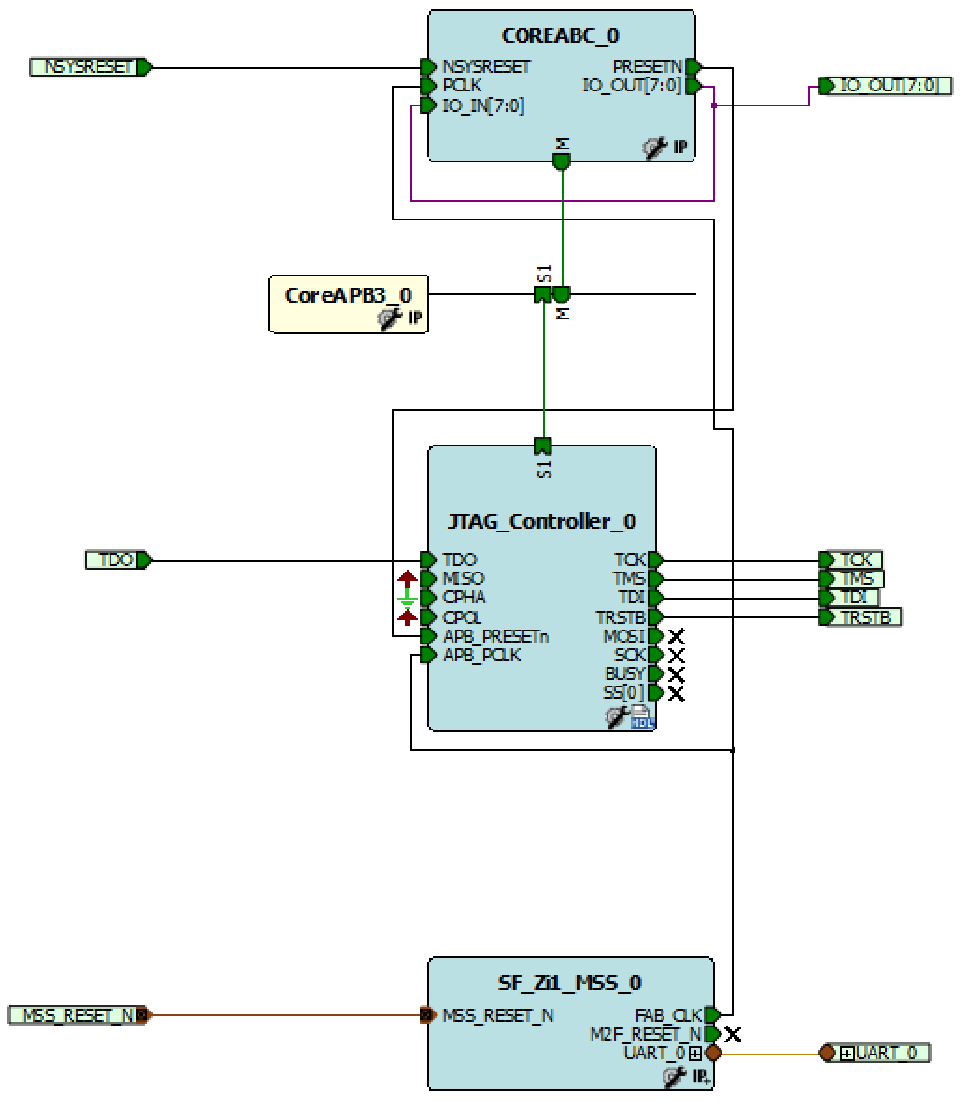

Figure 3 • **SmartDesign Implementation**

The CoreABC microcode writes data to the I/Os depending on the Zeroization status, as given below:

- 0x01 -> idle
- 0x02 -> JTAG initialized
- 0x04 -> first writing of All Zero's successful
- 0x06 -> first erase successful
- 0x09 -> second writing of All Zero's successful
- 0x0B -> second erase successful
- 0x0F -> successful completion of Zeroization
- 0x80 -> Error

The CoreABC microcode consists of the following procedures:

1. initialize: Enables the JTAG controller and keeps the target device's JTAG TAP controller in idle state. If any error occurs, the 0x80 pattern is sent to the status I/Os.
2. do_bulk_program: This procedure is to write zero's into the target FPGA. 0's are written segment-wise into the target FPGA. This is done by first latching the data into the tiles and then selecting all rows for each segment. If any error occurs, the 0x80 pattern is sent to the I/Os.
3. do_erase_disturb: For writing all 1's into the target FPGA, this procedure gives an erase command to the JTAG controller for a time duration of 1.5 seconds. After 1.5 seconds the TAP controller is forced to the reset state.
4. do_exit: This disables the programming mode of the JTAG controller outputs and forces the TAP controller into the reset state.
5. do_array_verify_0_pass: This verifies the successful completion of erasing. If an error occurs, then the 0x80 pattern is sent to the I/Os.

After successful Zeroization, 0x0F is sent to the I/Os.

The custom JTAG controller logic derives the TMS and TCK according to the instructions passed by the CoreABC microcontroller. The instruction or data sent by the CoreABC microcontroller is converted into serial bits and given to the TDI pin of the JTAG controller.

Running the Design

The design files (refer to "Appendix A - Design Files" on page 6) are created for the SmartFusion A2F200M3F device to erase a Fusion AFS600 device. Program the SmartFusion Evaluation Kit Board (A2F-EVAL-KIT) with the provided STAPL file. If you are using the A2F-EVAL-KIT, then connect the J22 pins to the AFS600-based board JTAG interface as shown in Table 1. If you are using the A2F500-DEV-KIT, then connect the J13 pins to the AFS600-based board JTAG interface as shown in Table 2 on page 6.

To erase the AFS600 device, press SW1 in the SmartFusion Evaluation Kit Board. After successful Zeroization of the AFS600 device, the LED pattern on the SmartFusion Evaluation Kit Board changes to 00001111.

Table 1 • **A2F-EVAL-KIT Connections**

J22 pins	AFS600 JTAG pins
1	TCK
2	TDI
3	TDO
4	TMS
5	TRSTB

Table 2 • **A2F500-DEV-KIT Connections**

J13 pins	AFS600 JTAG pins
1	TCK
9	TDI
3	TDO
5	TMS
8	TRSTB

Conclusion

This application note describes the enhanced security level for Microsemi flash based FPGAs using the Zeroization technique. A compact, reliable design is provided to erase the flash based FPGA, including its FlashROM, within 5 seconds of detecting a tampering event.

Appendix A - Design Files

The design files are available on the Microsemi SoC Product Groups website for download: www.microsemi.com/soc/download/rsc/?f=LPF_AC382_DF.

The design zip file consists of Libero® System-on-Chip (SoC) projects and programming files (*.stp) for A2F500 and A2F200 devices. Refer to the Readme.txt file included in the design file for directory structure and description.

The programming file (*.stp) in release mode is also provided on the Microsemi SoC Products Group website for download: www.microsemi.com/soc/download/rsc/?f=LPF_AC382_PF.

The programming zip file consists of a Readme.txt file and STAPL programming files (*.stp) for A2F500 and A2F200 devices.

Appendix B - Device Parameters in Microcode

Each device type/density has a set of associated parameters. Two of these parameters are required to be manually added to the Fast Erase microcode. The first one is “sdtiles”. This parameter is labeled as “numberofsdtiles” in the microcode and should be loaded with the value of “sdtiles” found in the device parameter database available at the Microsemi factory.

The second parameter is the “tsegmap”, which is an array ranging from 6 to 182 bits. These bits define the arrangement of tiles in each segment. The “tsegmap” is created from the SegmentMap[n] provided in the device parameter database. This array must be added to the bulk_program procedure in the Fast Erase microcode.

Below is an example parameter database for an AFS600:

```
const int sdtiles = 6;
const int tiles = 82;
const int segs = 8;
const int rows = 3444;  // number of CGs on device
const int segmentMap[82] = {
0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,
4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7}
AFS600 segments bulk_program algo.
tseg7 = 0x0 0000 0000 0000 0000 000F
tseg6 = 0x0 0000 0000 0000 0000 FFF0
tseg5 = 0x0 0000 0000 0000 0FFF 0000
tseg4 = 0x0 0000 0000 03FF F000 0000
tseg3 = 0x0 0000 003F FC00 0000 0000
tseg2 = 0x0 0003 FFC0 0000 0000 0000
tseg1 = 0x0 3FFC 0000 0000 0000 0000
tseg0 = 0x3 c000 0000 0000 0000 0000
```

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

51900250-0/3.12