

# Using SRAM PUF System Service in SmartFusion2 - Libero SoC v11.4

## Table of Contents

Purpose . . . . .	1
Introduction . . . . .	1
References . . . . .	2
System Controller Block in the SmartFusion2 Device . . . . .	3
SRAM PUF Services . . . . .	4
Design Requirements . . . . .	8
Design Description . . . . .	8
Hardware Implementation . . . . .	8
Software Implementation . . . . .	9
Setting up the Design . . . . .	10
Running the Design . . . . .	12
Conclusion . . . . .	17
Appendix A - Design and Programming Files . . . . .	18
List of Changes . . . . .	19

## Purpose

This application note explains how to access the static random-access memory (SRAM) physical unclonable functions (PUF) services in the SmartFusion<sup>®</sup>2 system-on-chip (SoC) field programmable gate array (FPGA) devices.

## Introduction

SRAM consists of transistors. The initial power-up values of these transistors behave randomly and independently due to the manufacturing differences such as: thickness of gate dielectric, number of atoms diffused in to channel region, and other random process variations.

The SRAM PUF generates a device-individual fingerprint using the startup behavior of SRAM. It can serve as a root of trust and provides a key that cannot be easily reverse engineered.

PUF is unique for each chip as the physical characteristics are unique for each chip, difficult to predict, easy to evaluate, and reliable. It has the following advantages:

- Uses only a small block of standard SRAM
- Fast Implementation in hardware

SRAM PUF feature is available in the larger SmartFusion2 devices such as:

- M2S090TS
- M2S150TS

In the M2S090TS and M2S150TS SoC devices, SRAM PUF is accessible through the system services. The system services are system controller actions initiated by asynchronous events from the ARM<sup>®</sup> Cortex<sup>®</sup>-M3 processor or a fabric master in the SmartFusion2. The SRAM PUF is used for data and design security applications and can be disabled through factory or user security settings.

This application note provides the design example to access the following SRAM-PUF services. For more information, refer to the ["SRAM PUF Services" section on page 4](#).

- Create User AC (Activation Code)
- Delete User AC
- Get Number of KC (Key Code)
- Create User KC for an Intrinsic Key
- Create User KC for an Extrinsic Key
- Export all KC
- Import all KC
- Delete User KC
- Fetch a User PUF Key
- Get a PUF Seed

## References

The following documents are referenced in this document. The references complement and help in understanding the relevant Microsemi® SmartFusion2 FPGAs device flows and features.

- [SmartFusion2 Microcontroller Subsystem User Guide](#)
- [SmartFusion2 System Controller User Guide](#)
- [SmartFusion2 Security and Reliability User Guide](#)

## System Controller Block in SmartFusion2 Device

The SRAM PUF services provide access to the System Controller's PUF core. SRAM PUF Core block is accessed through the communication block (COMM\_BLK).

There are two COMM\_BLK instances:

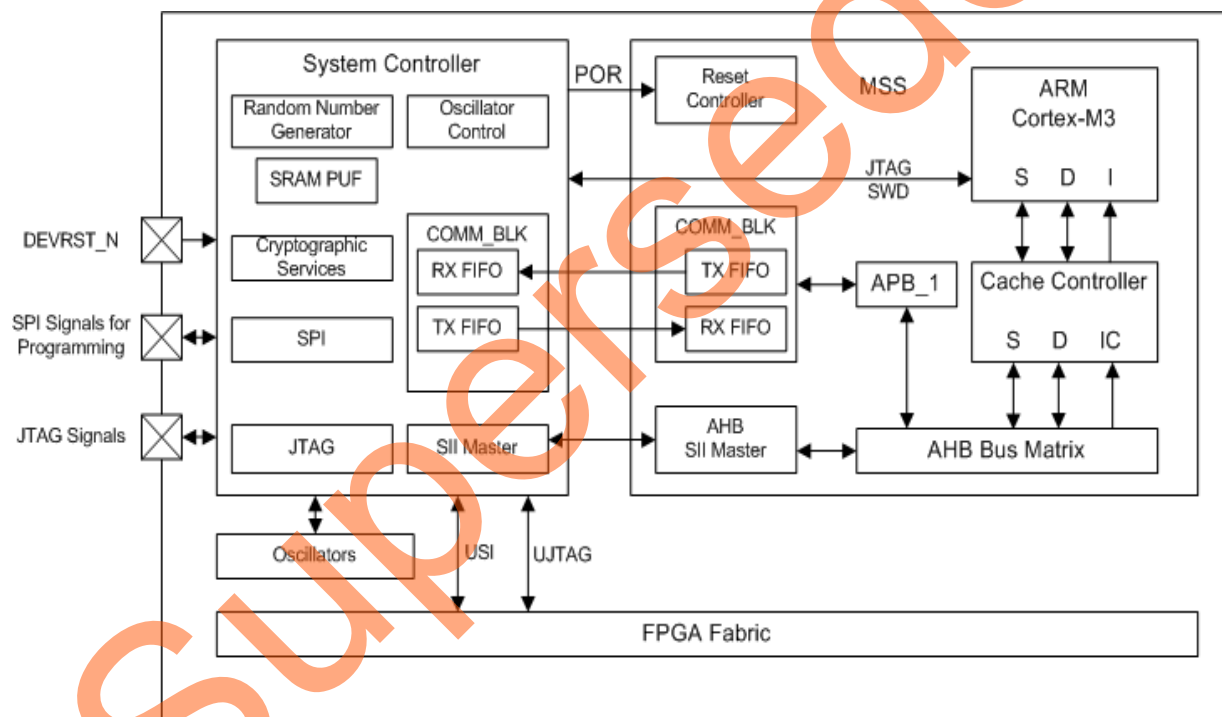
- One in the microcontroller subsystem (MSS) that the user interfaces with
- The other that communicates with the first block that is located in the system controller

The COMM\_BLK consists of an APB interface, eight byte transmit FIFO, and eight byte receive FIFO. The COMM\_BLK provides a bi-directional message passing facility between the MSS and the system controller.

The PUF system services are initiated using the COMM\_BLK in the MSS, which can be read or written by any master on the AMBA high performance bus (AHB) matrix; typically either the Cortex-M3 processor or a design in the FPGA fabric (also known as a fabric master).

The system controller receives the command through the COMM\_BLK in the system controller. On completion of the requested service, the system controller returns a status message through the COMM\_BLK. The responses generated are based on the selected command.

Figure 1 shows the System Controller block in the SmartFusion2.



**Figure 1 • System Controller Block in SmartFusion2 Device**

For more information about "System Controller", refer to the [SmartFusion2 System Controller User Guide](#).

For more information about "COMM\_BLK", refer to the "Communication Block" chapter in the [SmartFusion2 Microcontroller Subsystem User Guide](#).

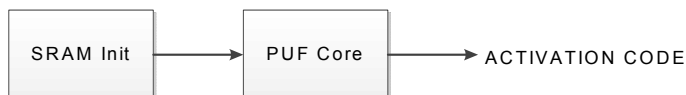
## SRAM PUF Services

The SRAM PUF core provides services for key generation.

The initial step for a key generation is the enrollment process where the start-up states of the SRAM PUF are used to derive an Activation Code for the device. The device may be enrolled multiple times, producing a new Activation Code for each enrollment.

**Note:** The Activation code differs from chip to chip and as a result cannot be generated easily, for more information, refer to ["Introduction" on page 1](#).

### Enrollment Process



**Figure 2 • Enrollment Process**

There are two modes of generating a key:

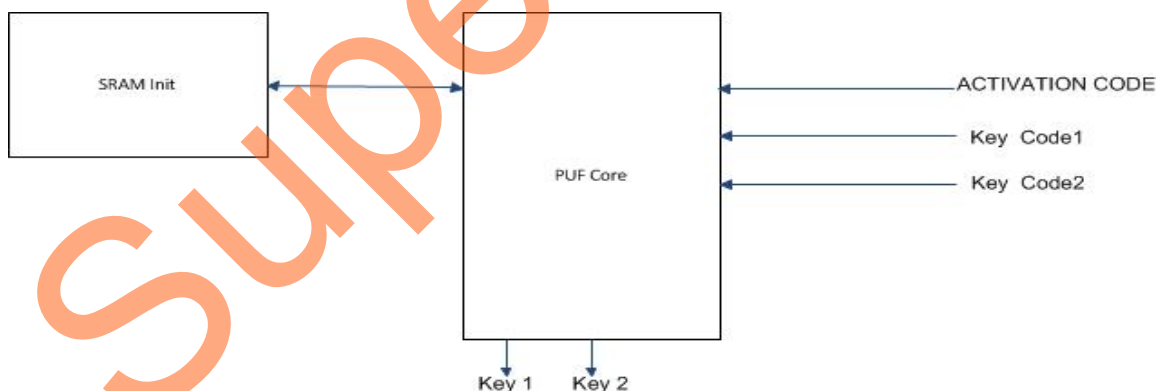
- Intrinsic
- Extrinsic

Intrinsic keys are generated by the PUF core from the SRAM and extrinsic keys are supplied by the user through the system services. For a given Activation Code, the PUF core generates one or more device-specific intrinsic keys. For each intrinsic key, the PUF creates a Key Code, which can be stored and used along with the Activation Code to reconstruct intrinsic keys.

For a given extrinsic keys provided by the client design, PUF core also generates a key code. This can be stored and later used along with the Activation Code to reconstruct the extrinsic key.

The first two keys are reserved for design security purposes. Additional data security keys of variable length may also be enrolled. These data security keys can be reconstructed.

### Generating a Key



**Figure 3 • Generating a Key**

When the correct Activation Code is assigned and then associated with an appropriate Key Code, the PUF core can reconstruct the original intrinsic or extrinsic key.

The PUF core can also be used to generate random seeds derived from the start up states of the SRAM.

### SRAM PUF Services

The PUF services provide access to the System Controller's PUF core. The PUF core provides services for the key generation.

### Creating or Deleting the User Activation Code

- CREATE\_AC sub command enrolls a new user activation code. The 1192 byte AC is stored in eNVM for future use in generating user keys.
- DELETE\_AC sub command deletes AC with all user key codes.

Table 1 shows the command value to Create or Delete the User Activation Code and response status.

**Table 1** • Create or Delete the User Activation Code Service command

System Service Name	Command Value	Sub Command	Response Status
Create or Delete the User Activation Code	25	0: CREATE_AC 1: DELETE_AC	0: Success completion 1: eNVM MSS error 2: PUF error, when creating 3: Invalid subcmd 4: eNVM program error 7: eNVM verify error 127: HRESP error occurred during MSS transfer 253: License not available in device 254: Service disabled by factory security 255: Service disabled by user security

### Create or Delete User Key Code and Export/Import All

The system service is used to generate or delete user key code with the existing activation code from the eNVM it is also used to get the number of keys. User can export and import the generated key codes using this service.

- GET\_NUMBER\_OF\_KC sub command returns the total number of keys. All keys valid and invalid are counted up to the last valid key. Invalid key is one that got deleted.
- The total number of keys are a minimum of two (KC#0 and KC#1).
- The maximum number of keys are: 58, that is, KC#0 through KC#57.
- CREATE\_INT\_KC or CREATE\_EXT\_KC sub command generates the new user key code using the existing activation code for an intrinsic and extrinsic key respectively.
- The user key code is stored in eNVM, for future use in generating the user keys.
- Key size supported range is from: 64 bits to 4096 bits.

**Note:** In this application note only 64 bit key size is used.

- EXPORT\_ALL\_KC sub command export key codes 0 to 57 in encrypted form. The stored user AC and all KC's are first XORed with the one-time pad and copied to a contiguous memory space specified. User needs to take care of the exported memory space (that is, size of the memory space can be calculated using the following formula) and the original key memory space. User can only export once.
- The Key Code size (KC\_size) depends on the key size and can be calculated with the following formula:  $KC\_size = 96 + \text{ceiling}_{256}(\text{Key Size})$ ,  $\text{ceiling}_{256}()$  rounds up to the next multiple of 256.
- IMPORT\_ALL\_KC sub command reads user AC and all KC's from a contiguous memory space used at the time of export.
- The individual private keys are regenerated from the PUF and are copied into the individual memory address spaces defined by the CREATE\_EXT\_KC or CREATE\_INT\_KC sub command and stored in private eNVM.
- This operation can only be successful if an EXPORT operation has been successful previously.
- DELETE\_KC sub command deletes the KC corresponding to key number provided. User cannot delete keys 0 and 1.

Table 2 shows the command value and response status to Create, Delete User Key Code, and Export/Import All.

**Table 2 • Create or Delete User Key Code and Export/Import All Service Command**

System Service Name	Command Value	Sub Command	Response Status
Create or Delete User Key Code (User KC) and Export or Import all	26	0: GET_NUMBER_OFKC 1: CREATE_EXT_KC 2: CREATE_INT_KC 3: EXPORT_ALL_KC 4: IMPORT_ALL_KC 5: DELETE_KC	0: Success completion 1: eNVM MSS error 2: PUF error, when creating 3: Invalid request or KC, when exporting or importing 4: eNVM program error 5: Invalid hash 6: Invalid user AC 7: eNVM verify error 8: Incorrect key size for renewing a KC 10: Private eNVM user digest mismatch 11: Invalid subcmd 12: DRBG error 127: HRESP error occurred during MSS transfer 253: License not available in device 254: Service disabled by factory security 255: Service disabled by user security

### Fetch a User PUF Key

Fetch a User PUF Key regenerates the key using the existing activation code and key code located in the eNVM memory. User needs to take care of the addresses of the PUF Key at the time of enrollment for regenerating the PUF key. User cannot use this function after EXPORT.

Table 3 shows the command value and response status to fetch a user PUF key.

**Table 3 • Fetch a User PUF Key service command**

System Service Name	Command Value	Response Status
Fetch a User PUF Key	27	0: Success completion 2: PUF error, when creating 3: Invalid keynum or argument or exported or invalid key 5: Invalid hash 10: Private eNVM user digest mismatch 127: HRESP error occurred during MSS transfer 253: License not available in device 254: Service disabled by factory security 255: Service disabled by user security

### Get a PUF Key

Get a PUF Seed generates a 256 bit seed.

Table 4 shows the command value and response status to get a PUF seed.

**Table 4 •** Get a PUF Seed service command

System Service Name	Command Value	Response Status
Get a PUF Key	29	0: Success completion 2: PUF error, when creating 127: HRESP error occurred during MSS transfer 253: License not available in device 254: Service disabled by factory security 255: Service disabled by user security

For more information about SRAM PUF service requests, refer to [SmartFusion2 System Controller User Guide](#).

Superseded

## Design Requirements

Table 5 shows the design requirements.

**Table 5 • Design Requirements**

Design Requirements	Description
<b>Hardware Requirements</b>	
SmartFusion2 Evaluation Kit: <ul style="list-style-type: none"><li>• 12 V adapter</li><li>• FlashPro4 programmer</li><li>• USB A to Mini-B cable</li></ul>	Rev D or later
Host PC or Laptop	Any 64-bit Windows Operating System
<b>Software Requirements</b>	
Libero® SoC	v11.4 SP 1
FlashPro programming software	v11.4 SP 1
USB to UART drivers	-
One of the following serial terminal emulation programs: <ul style="list-style-type: none"><li>• HyperTerminal</li><li>• TeraTerm</li><li>• PuTTY</li></ul>	-

## Design Description

The design is implemented on the SmartFusion2 Evaluation Kit Board using M2S090TS-1FGG484 device.

The design example consists of:

- RC oscillator
- Fabric CCC
- CORERESET
- MSS

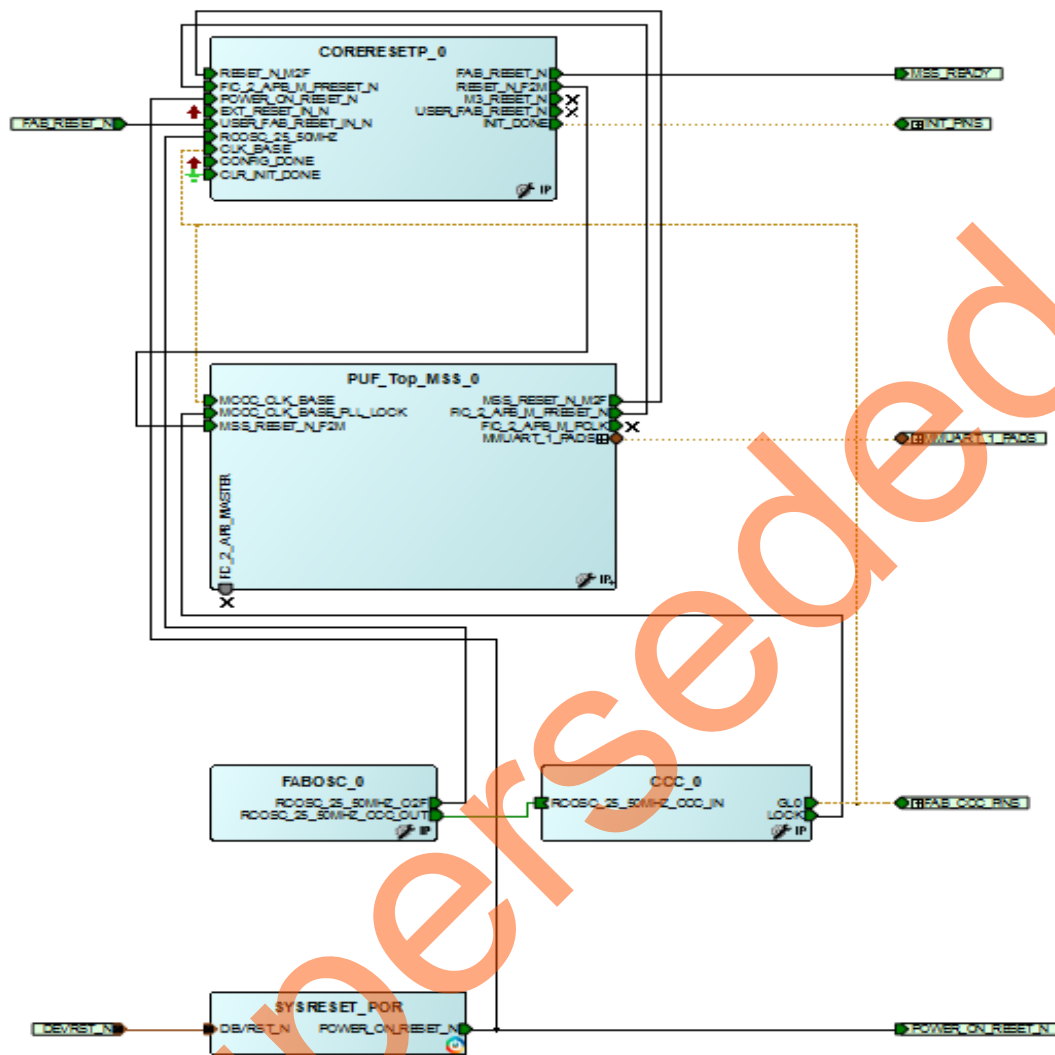
The fabric PLL is used to provide the base clock for the MSS. The system services are run using various C routine in the MSS, as shown in the following sections. In addition, a universal asynchronous receiver/transmitter (UART1) in the MSS is used to display the operation of the PUF system service.

## Hardware Implementation

Figure 4 shows a block diagram of the design example. The RC oscillator generates a 50 MHz input clock and the fabric PLL generates a 100 MHz clock from the RC oscillator. This 100 MHz clock is used as the base clock for the MSS.



The MMUART\_1 signals are for communicating with the serial terminal program.



**Figure 4 •** Block Diagram of SmartFusion2 SRAM PUF Design Example

## Software Implementation

The software design example performs the following operations:

- Create or Delete the User Activation Code
- Get Number of the Key Code
- Create or Delete the User Key Code
- Export or Import All Key Codes
- Fetch a User PUF Key
- Fetch a PUF ECC Public Key
- Get a PUF Seed

## Firmware Drivers

The following firmware drivers are used in this application:

- MSS MMUART driver: To communicate with serial terminal program on the Host PC.
- MSS System Services driver: Provides access to SmartFusion2 System Services.
- MSS eNVM driver: Provides access to SmartFusion2 eNVM.

## List of APIs

The following APIs in Table 6 are used in software design to access the SRAM PUF Services.

**Table 6 •** APIs to access the PUF System Services

API	Description
MSS_SYS_puf_create_activation_code()	Create activation code
MSS_SYS_puf_delete_activation_code()	Delete activation code
MSS_SYS_puf_get_number_of_keys()	Returns total number of user keys
MSS_SYS_puf_enroll_key()	Enrolls a new user key code, for an intrinsic and extrinsic key
MSS_SYS_puf_fetch_key()	Retrieve a user PUF key
MSS_SYS_puf_delete_key()	Delete a previously enrolled key
MSS_SYS_puf_export_keycodes()	Export an encrypted copy of all the key codes
MSS_SYS_puf_import_keycodes()	Import a set of PUF key codes that was previously exported
MSS_SYS_puf_get_random_seed()	Generate a 256-bit random seed

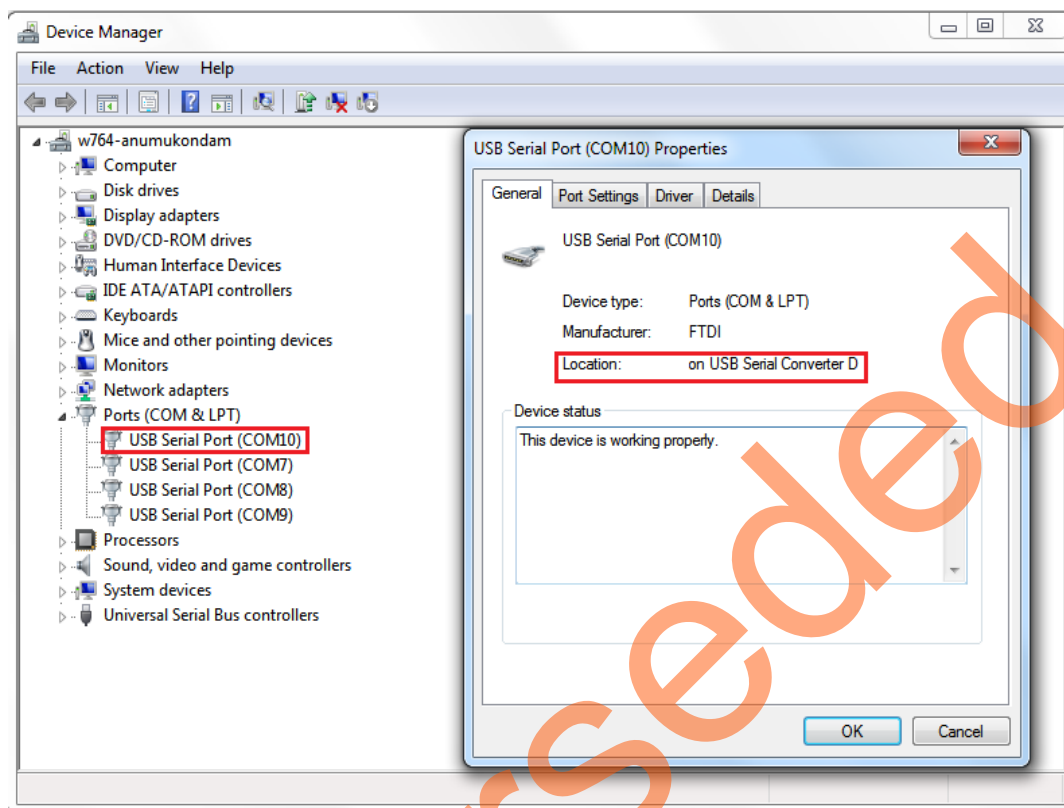
## Setting up the Design

Plug the FlashPro4 ribbon cable into the connector J5 (JTAG Programming Header) on the SmartFusion2 Evaluation Kit Board.

1. Connect the mini USB cable between the FlashPro4 and USB port of the PC.
2. Connect the power supply to the J6 connector.
3. Connect one end of the USB mini cable to the J18 connector provided on the SmartFusion2 Evaluation Kit. Connect the other end of the USB cable to the host PC.
4. Ensure that the USB to UART bridge drivers are automatically detected. This can be verified in the Device Manager.

Figure 5 shows example device manager window. If USB to UART bridge drivers are not installed, download and install the drivers from:

[www.microsemi.com/soc/documents/CDM\\_2.08.24\\_WHQL\\_Certified.zip](http://www.microsemi.com/soc/documents/CDM_2.08.24_WHQL_Certified.zip)



**Figure 5 •** Device Manager Window

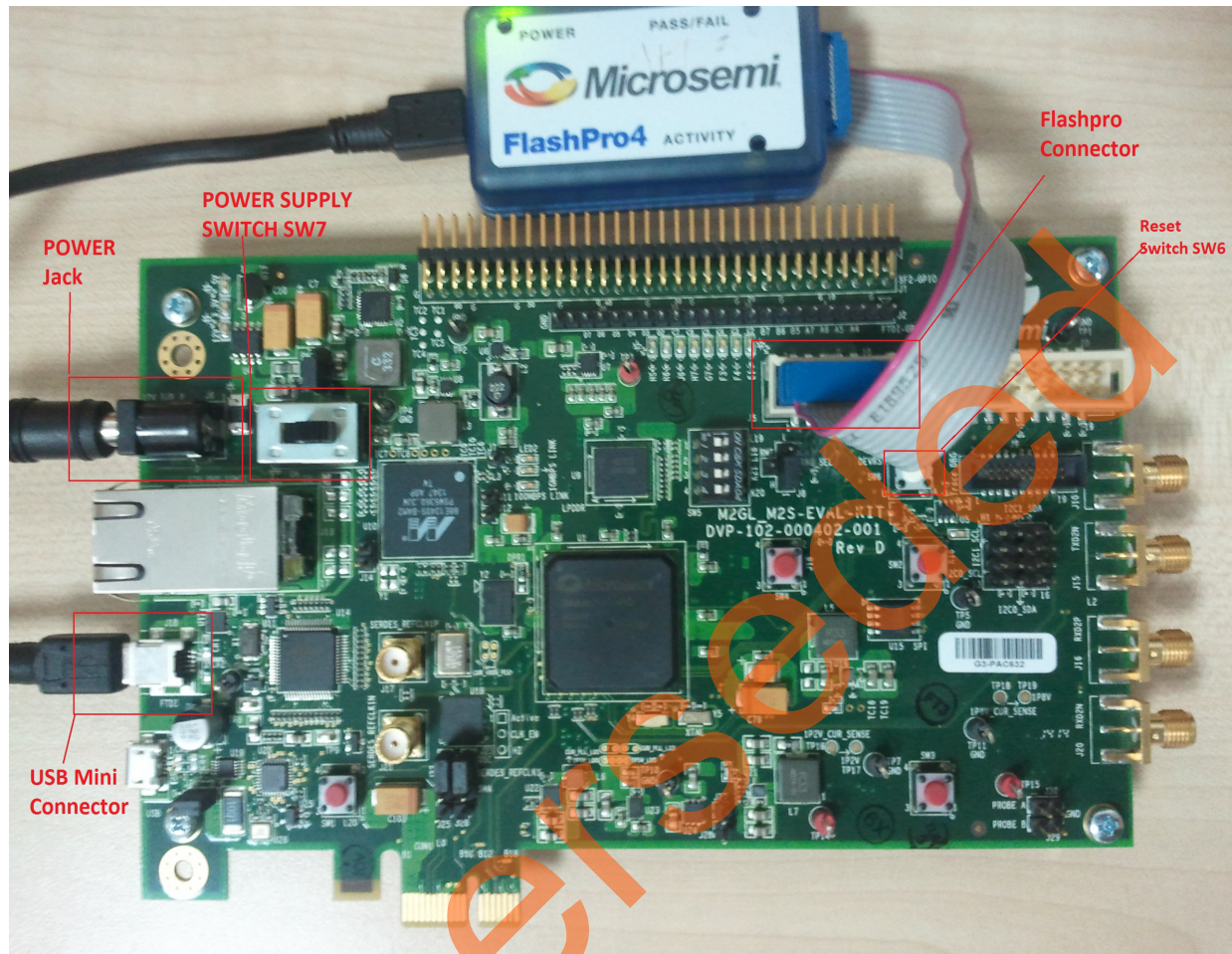
5. Connect the jumpers on the SmartFusion2 Evaluation Kit as shown in [Table 7](#).

**Note:** Ensure that power supply switch **SW7** is switched off while connecting the jumpers on the SF2 Kit.

**Table 7 •** SmartFusion2 SoC FPGA Evaluation Kit Jumper Settings

Jumper	Pin (from)	Pin (to)	Comments
J22	1	2	default
J23	1	2	default
J24	1	2	default
J8	1	2	default
J3	1	2	default

Figure 6 shows the board setup for running the PUF services design on the SmartFusion2 Evaluation Kit.



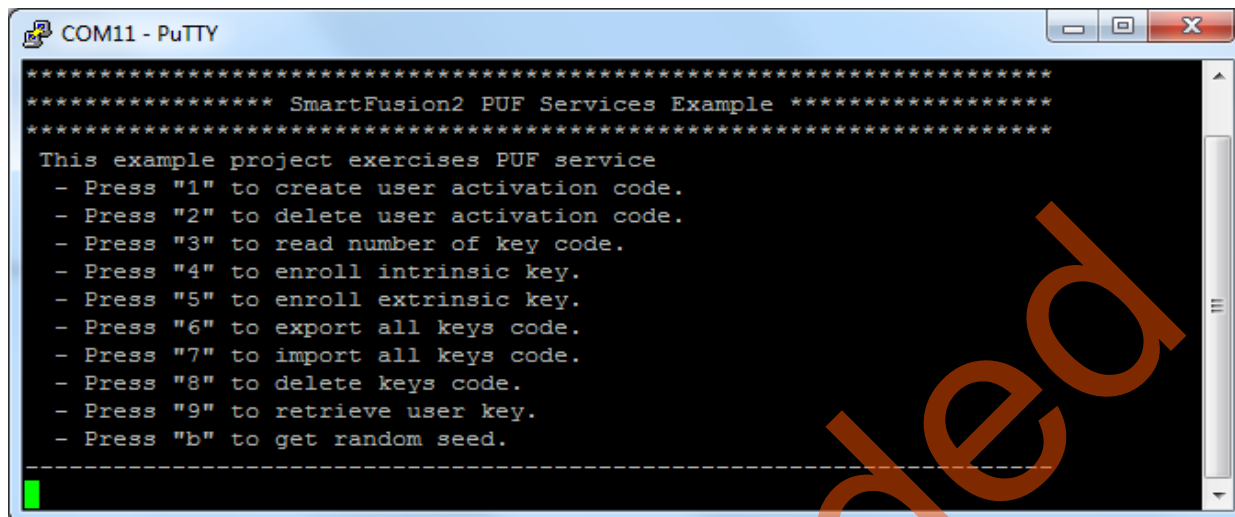
**Figure 6 • SmartFusion2 Evaluation Kit**

## Running the Design

The following steps describes how to run the design on the SmartFusion2 Evaluation Kit Board using the M2S090TS-1FGG484 device.

1. Switch ON the power supply switch, **SW7**.
2. Start a PUTTY session with 115200 baud rate, 8 data bits, 1 stop bit, no parity, and no flow control. Use any free serial terminal emulation program such as: HyperTerminal or Tera Term, if the computer does not have the PUTTY program. For more information about configuring HyperTerminal, Tera Term, or PuTTY, refer to [Configuring Serial Terminal Emulation Programs Tutorial](#).
3. Program the SmartFusion2 Evaluation Kit Board with the provided STAPL file using FlashPro4. Refer to "Appendix A - Design and Programming Files" on page 18 for more information.

4. After programming, press switch SW6 (DEVRST) PUTTY displays a message to run the PUF Services as shown in [Figure 7](#).



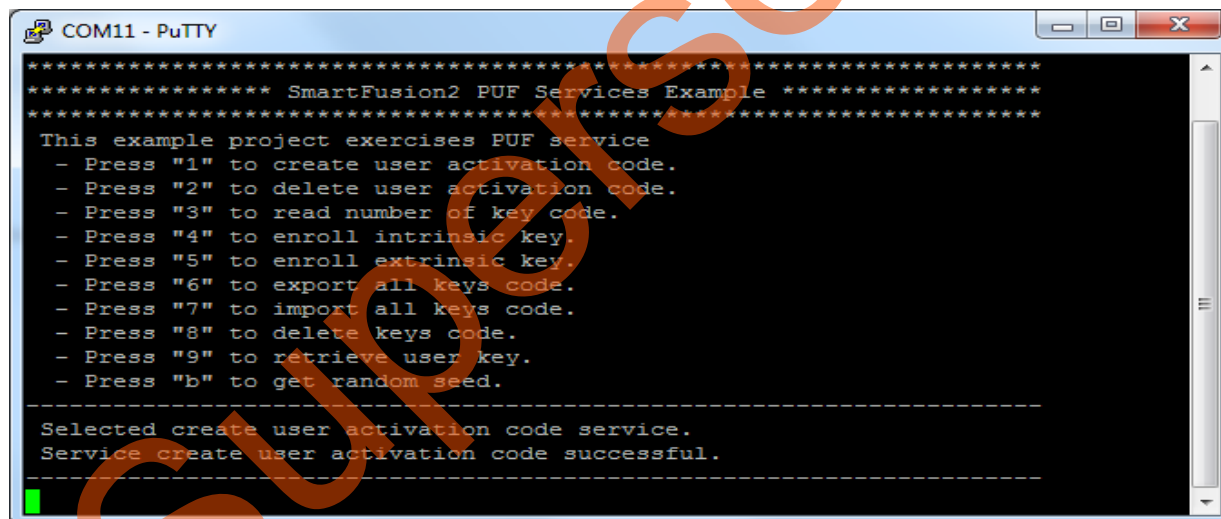
```

COM11 - PuTTY
***** SmartFusion2 PUF Services Example *****
*****
This example project exercises PUF service
- Press "1" to create user activation code.
- Press "2" to delete user activation code.
- Press "3" to read number of key code.
- Press "4" to enroll intrinsic key.
- Press "5" to enroll extrinsic key.
- Press "6" to export all keys code.
- Press "7" to import all keys code.
- Press "8" to delete keys code.
- Press "9" to retrieve user key.
- Press "b" to get random seed.
-----

```

Figure 7 • Welcome Message

5. Enter 1 to enroll new activation code as shown in [Figure 8](#).



```

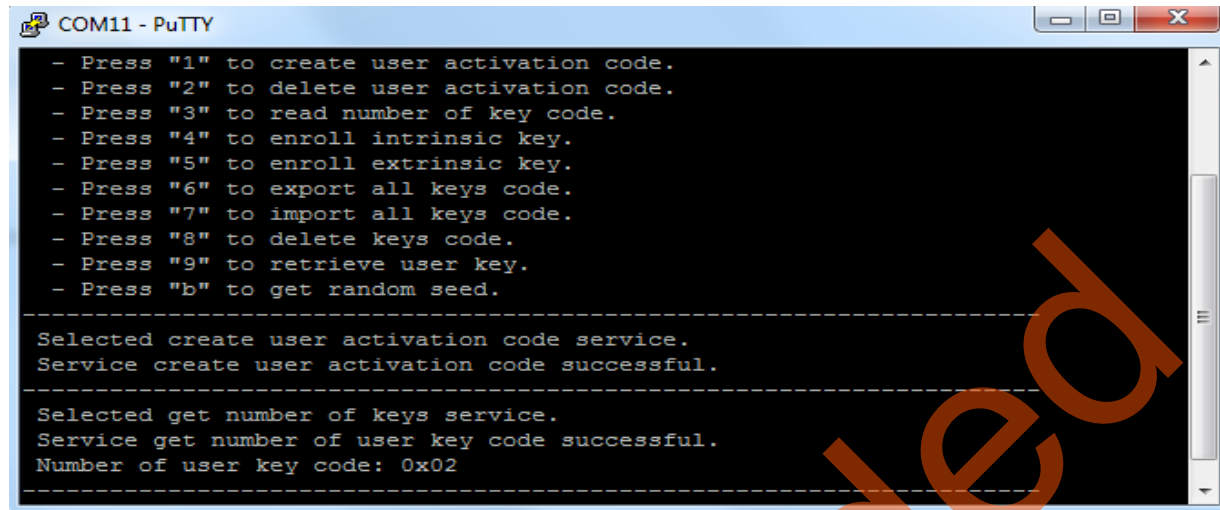
COM11 - PuTTY
***** SmartFusion2 PUF Services Example *****
*****
This example project exercises PUF service
- Press "1" to create user activation code.
- Press "2" to delete user activation code.
- Press "3" to read number of key code.
- Press "4" to enroll intrinsic key.
- Press "5" to enroll extrinsic key.
- Press "6" to export all keys code.
- Press "7" to import all keys code.
- Press "8" to delete keys code.
- Press "9" to retrieve user key.
- Press "b" to get random seed.
-----
Selected create user activation code service.
Service create user activation code successful.
-----

```

Figure 8 • Creating New Activation Code

6. Enter 3 to read number of keys. No of keys is displayed as 2 as shown in [Figure 9](#). These are design security keys: **KC0** and **KC1**.





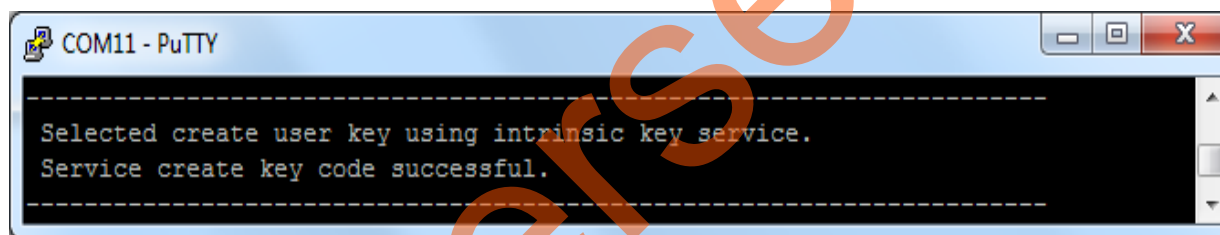
```
COM11 - PuTTY
- Press "1" to create user activation code.
- Press "2" to delete user activation code.
- Press "3" to read number of key code.
- Press "4" to enroll intrinsic key.
- Press "5" to enroll extrinsic key.
- Press "6" to export all keys code.
- Press "7" to import all keys code.
- Press "8" to delete keys code.
- Press "9" to retrieve user key.
- Press "b" to get random seed.

-----
Selected create user activation code service.
Service create user activation code successful.

-----
Selected get number of keys service.
Service get number of user key code successful.
Number of user key code: 0x02
-----
```

Figure 9 • Read Number of the Key Code

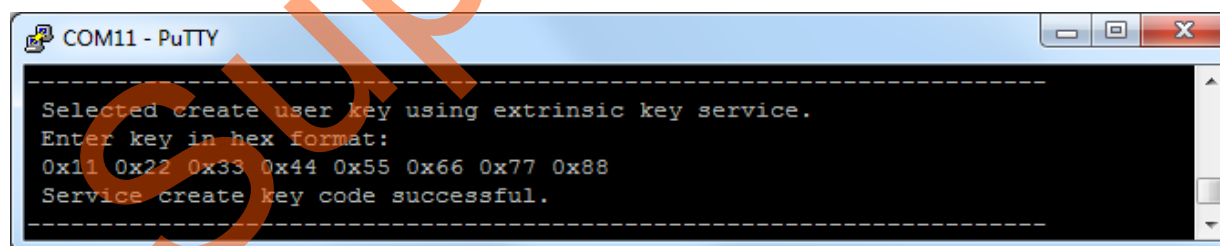
7. Enter 4 to enroll intrinsic key as shown in Figure 10.



```
COM11 - PuTTY
-----
Selected create user key using intrinsic key service.
Service create key code successful.
-----
```

Figure 10 • Enrolling an Intrinsic Key

8. Enter 5 to enroll extrinsic key. Enter 64-bit key as shown Figure 11.



```
COM11 - PuTTY
-----
Selected create user key using extrinsic key service.
Enter key in hex format:
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
Service create key code successful.
-----
```

Figure 11 • Enrolling an Extrinsic Key

9. Enter 9 to retrieve the key. Enter Key Numbers as shown in Figure 12.

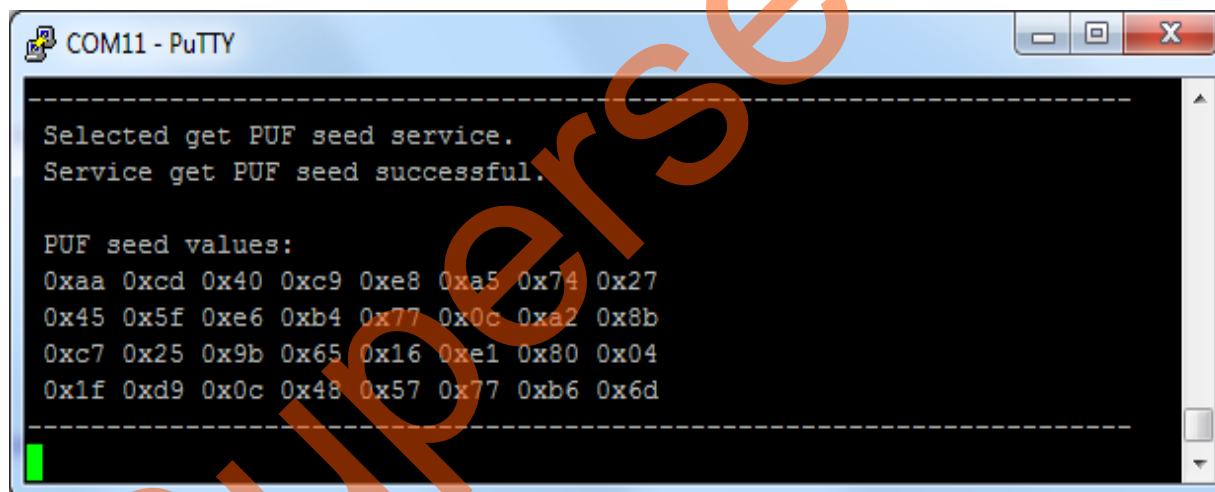


```
-----
Selected fetch user PUF key service.
Enter key number in hex format: 0x02
Service fetch user PUF key successful.
Key Number: 0x02
Key Code:
0x66 0x3b 0x08 0x9a 0xaf 0xd2 0x10 0xf3
-----

Selected fetch user PUF key service.
Enter key number in hex format: 0x03
Service fetch user PUF key successful.
Key Number: 0x03
Key Code:
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88
-----
```

Figure 12 • Retrieving the Key

10. Enter **b** to get a PUF seed as shown in Figure 13.



```
-----
Selected get PUF seed service.
Service get PUF seed successful.

PUF seed values:
0xaa 0xcd 0x40 0xc9 0xe8 0xa5 0x74 0x27
0x45 0x5f 0xe6 0xb4 0x77 0x0c 0xa2 0x8b
0xc7 0x25 0x9b 0x65 0x16 0xe1 0x80 0x04
0x1f 0xd9 0x0c 0x48 0x57 0x77 0xb6 0x6d
-----
```

Figure 13 • PUF Seed Service

11. Enter **8** to delete the key. Enter the key number to delete the key as shown in Figure 14

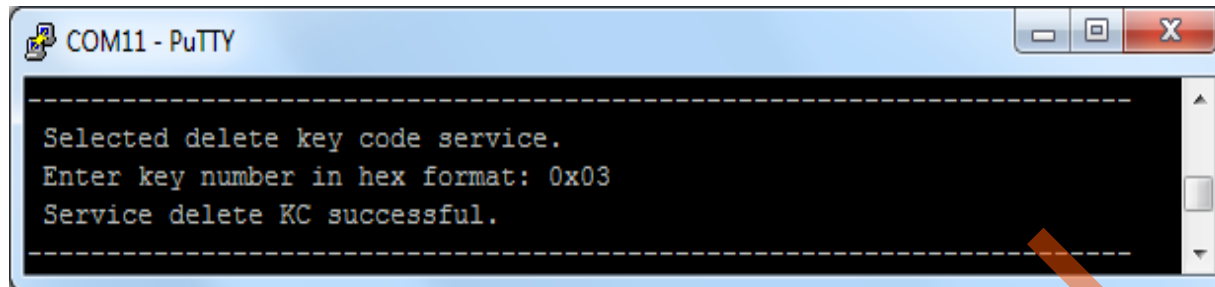


Figure 14 • Deleting Key Code

12. Enter 6 to export all key codes as shown in Figure 15.

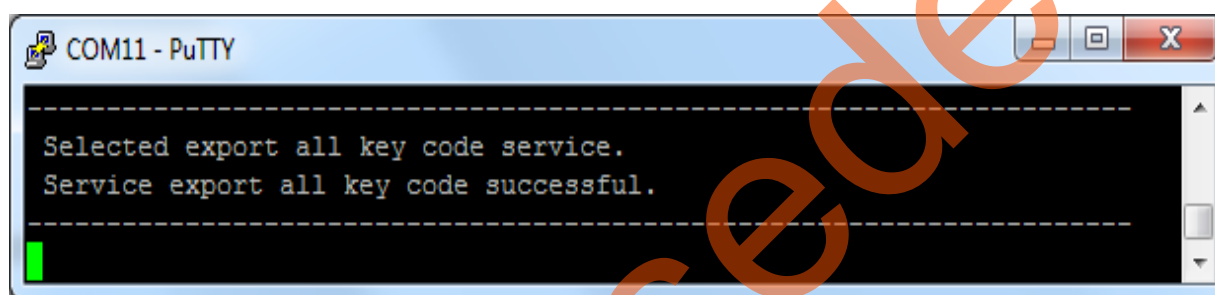


Figure 15 • Export all Key Codes

13. Enter 7 to import all key codes as shown in Figure 16.

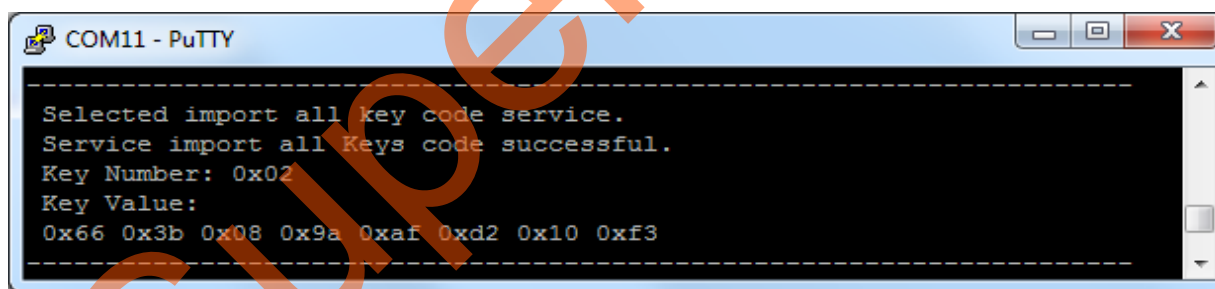


Figure 16 • Import all Key Codes



14. Enter **2** to delete activation code as shown in [Figure 17](#).

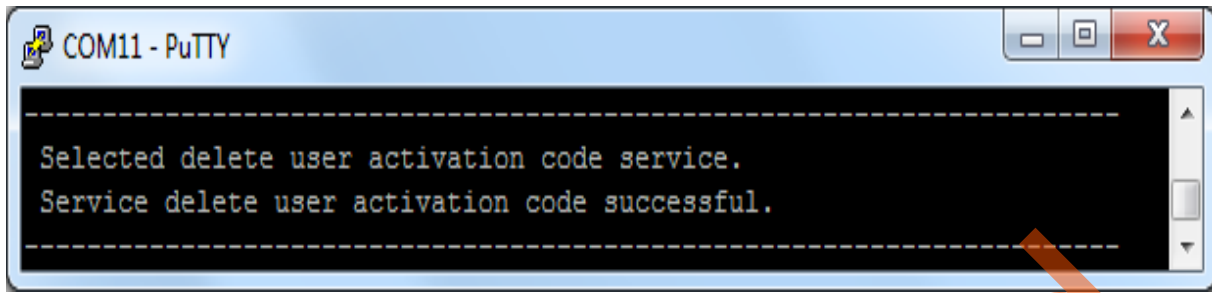


Figure 17 • Delete Activation Code

## Conclusion

This application note explains how to access the SRAM PUF services in the SmartFusion2 SoC FPGAs.

Superseded

## Appendix A - Design and Programming Files

You can download the design files from the Microsemi SoC Products Group Website:

[http://soc.microsemi.com/download/rsc/?f=SF2\\_AC434\\_11p4\\_DF](http://soc.microsemi.com/download/rsc/?f=SF2_AC434_11p4_DF)

The design files consists of a Libero Verilog project and programming files (\*.stp) for the SmartFusion2 Evaluation Kit.

Refer to the [readme.txt](#) file included in the design files for the directory structure and description.

Download the programming files (\*.stp) in release mode from the Microsemi SoC Products Group website:

[www.microsemi.com/soc/download/rsc/?f=SF2\\_AC434\\_11p4\\_PF](http://www.microsemi.com/soc/download/rsc/?f=SF2_AC434_11p4_PF)

The programming zip file consists of the STAPL programming file (\*.stp) for SmartFusion2 Evaluation Kit.

Superseded

## List of Changes

The following table lists critical changes that were made in each revision of the chapter in the application note.

Date	Changes	Page
Revision 1 (October 2014)	Initial release.	N/A

**Note:** \*The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.

Superseded

Superseded



**Microsemi®**

**Microsemi Corporate Headquarters**  
One Enterprise, Aliso Viejo CA 92656 USA  
Within the USA: +1 (800) 713-4113  
Outside the USA: +1 (949) 380-6100  
Sales: +1 (949) 380-6136  
Fax: +1 (949) 215-4996  
E-mail: [sales.support@microsemi.com](mailto:sales.support@microsemi.com)

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense and security, aerospace, and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs, and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 3,400 employees globally. Learn more at [www.microsemi.com](http://www.microsemi.com).

© 2014 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.