Application Note

Convert Sitara MPU HS-FS Silicon to HS-SE with No Boot Mode Switch

Zekun Bai, Prashant Shivhare, and Donna Xu

ABSTRACT

In automotive and industry applications, to protect system security and functional privacy, and to prevent application images from being maliciously tampered with, copied, or deleted, mass-produced products generally utilize high-security chips, unlike those used in general purpose development. The Sitara processors offer general purpose (GP) and high security (HS) chip types to address this issue. The Sitara HS chip incorporates an OTP identifier eFuse and multiple hardware security accelerators, adding encryption and decryption capabilities for system images and signature verification during the boot process, protecting the system from external malicious tampering.

The HS device also has two subtypes that represent the state of the HS device: high security-field securable (HS-FS) and high security-security enforced (HS-SE). All security features are enabled in the HS-SE device. During the development process, customers must use the TI-provided Keywriter tool to program a key into the chip, converting the chip from HS-FS to HS-SE.

This programming process typically involves different approaches. This application note summarizes common programming methods and proposes several new approaches that are more efficient and reduce manual intervention on the production line. These new methods eliminate the need to switch boot modes. Key programming can be performed within a single boot mode.

Table of Contents

Trademarks

1 Introduction

Here are the differences between HS-FS and HS-SE.

HS-FS Device

- Allows customers to run diagnostics code on HS device without creating signed images.
- No secure boot
- JTAG open

HS-SE Device

- Fully secure HS device
- All security policies applied
- Enforce secure boot
- JTAG closed
- Firewalls engaged
- All available security features active

During the development, customers must use Keywriter to convert HS-FS to HS-SE.

OTP Keywriter

The OTP writer for K3 platforms is developed as a single binary that runs on HS-FS devices and program the customer eFuse keys.

The OTP writer is a single image and contains a secure part and a non-secure part.

- Non-secure part, or the OTP app, runs on the R5
- Secure part, which is essentially an OTP driver, runs as part of SYSFW on the DMSC subsystem

Non-secure factory key provisioning support:

- Keywriter contains the encrypted TI FEK Private Key
- FEK Public Key given to customer to encrypt symmetric keys (SMEK and BMEK)

User configurable parameters are input using a X509 certificate. This OTP config cert contains:

- SMPK Hash and FEK encrypted SMEK, options and BCH
- BMPK Hash and FEK encrypted BMEK, options and BCH
- SWREV, KEYREV (to select the active key), KEYCNT (number of keys used)
- GPIO used for VPP (optional for 16FF devices)
- UART mux cfg for wkup UART
- TI FEK Private Key, encrypted using key derived from TI Symmetric key (MEK)
- Signed certificate with full Root Key (cloning protection)

2 HS Device Flashing With Boot Mode Switch

The typical boot flow of using Keywriter for mass production is Memory A boot mode and Memory B boot mode.

Here is an example:

1. Use an SD card as the boot medium (1) and store the keywriter boot image on the SD card. Use Norflash as the boot medium (2) and burn the service boot image and application offline at the factory.
2. Set the boot mode to SD card for the first power-up. Based on the boot mode setting, the Boot ROM reads the keywriter from the SD card, burns the customer key into the OTP area of the chip, and converts the chip from HS-FS to HS-SE.
3. Power off and switch the boot mode to OSPI boot mode.
4. Power on again, the Boot ROM reads the service boot image from the OSPI Norflash. Since the boot image is signed and encrypted, the Boot ROM verifies and decrypts the signature using the Boot ROM's X509 header, and then boots the cores normally.

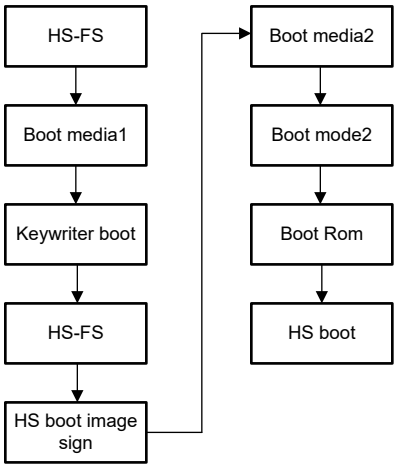

Figure 2-1. Typical Boot Flow with Boot Mode Switch

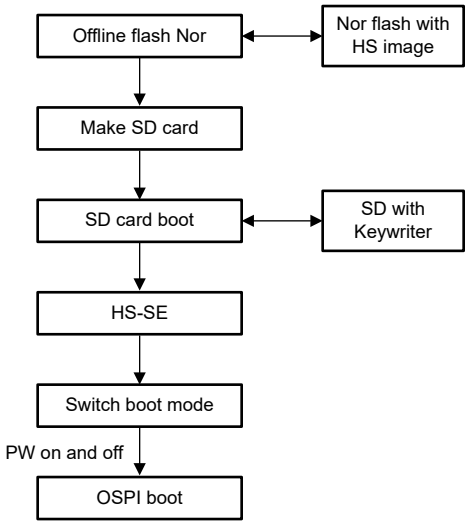

Figure 2-2. SD Card as First Boot, OSPI Boot as Second Boot

This design is relatively mature, avoiding many reliability and field operation issues by switching the boot mode in a single operation.

However, compatibility with both memory boot modes increases BOM costs and design complexity. A second issue is that switching the boot mode requires additional jigs, which complicates the production line.

Furthermore, this boot mode switching often requires manual intervention, further reducing production line efficiency and increasing the rate of issues.

3 HS Device Flashing Without Boot Mode Switch

If a jig-free and no-manual burning Keywriter method can be implemented on the production line that can support secure boot, that is significant to customers that must implement secure boot. TI's processors support redundant boot modes and redundant OSPI boot offset mechanisms. This application note leverages this mechanism to provide several designs to the problem.

Primary OSPI Offset and Backup OSPI Offset

The OSPI protocol is described according to bit-width (one or eight) and data rate (Single Data Rate (S) or Double Data rate (D)) for the Command/Address/Data segments of the protocol. The OSPI boot mode supports 1S-1S-8S mode. The Command and Address issued are eight bits and 24 bits respectively. The Read Command that is issued for OSPI mode is 0x8B followed by zero for address and eight dummy cycles. The frequency of operation supported is 50MHz. In the OSPI boot mode, the ROM code initializes the OSPI module and the image is read from the OSPI flash connected to the selected chip-select. If the image fails to be read correctly from offset 0x0 of the flash memory, the ROM attempts to obtain the image at offset 0x400000. This is the only redundant image location supported by the ROM. ROM code first copies boot image into on-chip RAM and then executes the image.

Primary Boot Mode and Backup Boot Mode

The DMSC is the boot controller for the Public ROM. DMSC performs the required configurations and releases the reset to R5.

R5 checks primary boot mode media and checks image integrity. If primary boot mode fails, then R5 changes to backup boot mode and checks image integrity.

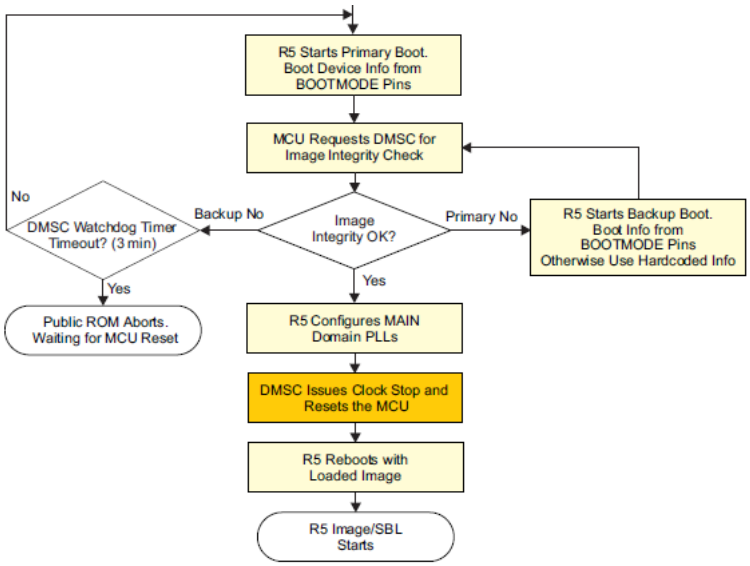

Figure 3-1. Sitara Primary Boot and Back up Boot Flow

3.1 Design 1: Booting from Backup Boot Media

The basic idea of this design is to connect to an external PC through a DFU. The PC stores three bootloaders. The first bootloader is a Keywriter, which is used to burn the key. The second bootloader is based on the SBL_Uniflash provided by the SDK. SBL_Uniflash is pre-signed and encrypted with the key from the customer. After booting, SBL_Uniflash burns the normal application files and bootloader3 (pre-signed and encrypted with the customer's key) to related offsets in the flash memory. Upon power up, the Boot ROM reads the application bootloader from the primary boot mode, which is OSPI and boots the SOC.

Requirements

Boot mode => Primary: OSPI, Backup: UART/DFU (DFU preferable).

Flash is completely empty. More importantly, the flash is erased at 0x0 and 0x400000 offsets so that ROM fallbacks to backup boot.

Procedure

1. On First POR: Device state: HS-FS. ROM boots OTP Keywriter from the selected backup boot media. Program the Keys according to the Keywriter guide. Convert HS-FS to HS-SE.
2. On Second POR: Device state: HS-SE.ROM boots flash writer from the selected backup boot media. Flash the SDK images (SBL/Applications)
3. On Third POR: Device state: HS-SE. ROM boots from the primary boot media OSPI.

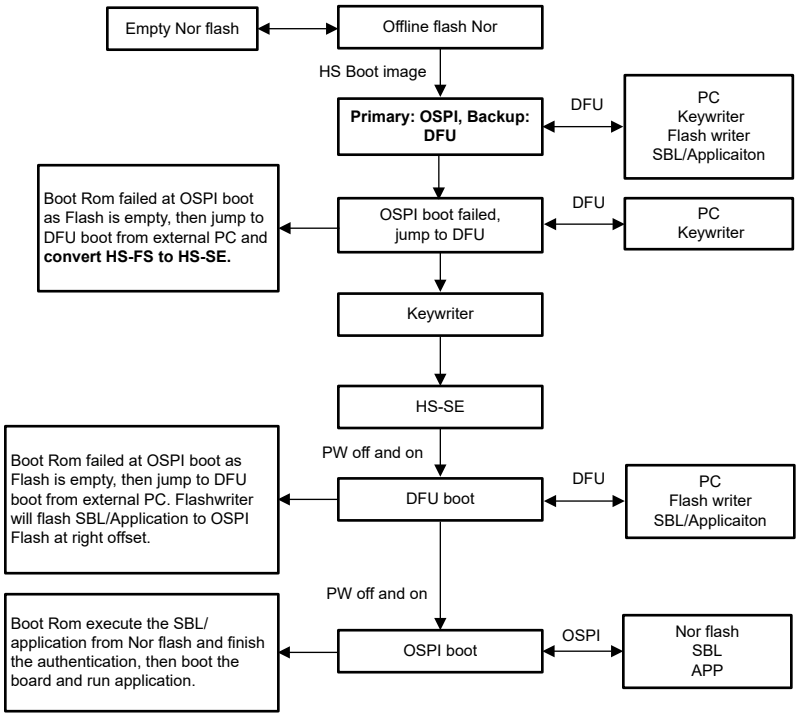

Figure 3-2. Boot from Backup Boot Media

3.2 Design 2: Booting from Primary Boot Media

The basic idea of this design is to offline burn the required files to the Nor flash at specified offsets. These files include three bootloaders.

- The first bootloader is a Keywriter located at address 0x0, which is used to burn the key.
- The second bootloader is a Flash Writer located at address 0x400000. This flash writer is single stage SBL/SPL based with the only job of moving the SBL from known location X to primary offset 0x0 and optionally redundant offset 0x400000.
- The third bootloader contains the boot loader from the customer and application (pre-signed and encrypted with the key from the customer). The address can be flexibly configured based on the selected Nor flash capacity, and the requirement is that the addresses do not overlap with the first two bootloaders.

During the first power-up, the program executes from the keywriter at address 0x0, converting the chip from HS-FS to HS-SE. During the second power-up, the BootROM fails to verify the keywriter at address 0x0 and jumps to address 0x400000 to execute the flash writer. This program burns the service startup program of the customer and application (pre-signed and encrypted with the key from the customer) to address 0x0 in the flash memory, overwriting the previous keywriter file. At this time, after powering on again, the Boot ROM normally reads the business startup program and application from address 0x0, completes the key signature verification and decryption, and then completes the boot and starts the SOC.

Requirements

Boot mode => Primary: OSPI, Backup: (Don't care)

Flash is offline flashed with three parts of files on related offset.

Procedure

1. On First POR: Device state: HS-FS. ROM boots OTP Keywriter from the 0x0 Nor flash offset. Program the keys according to the Keywriter guide. Convert HS-FS to HS-SE.
2. On Second POR: Device state: HS-SE. ROM boots Flash Writer from the 0x400000 Nor flash offset. Flash the SDK images (SBL/Applications) into 0x0, 0x8000.
3. On Third POR: Device state: HS-SE. ROM boot customer bootloader and application from 0x0 offset and bring up board.

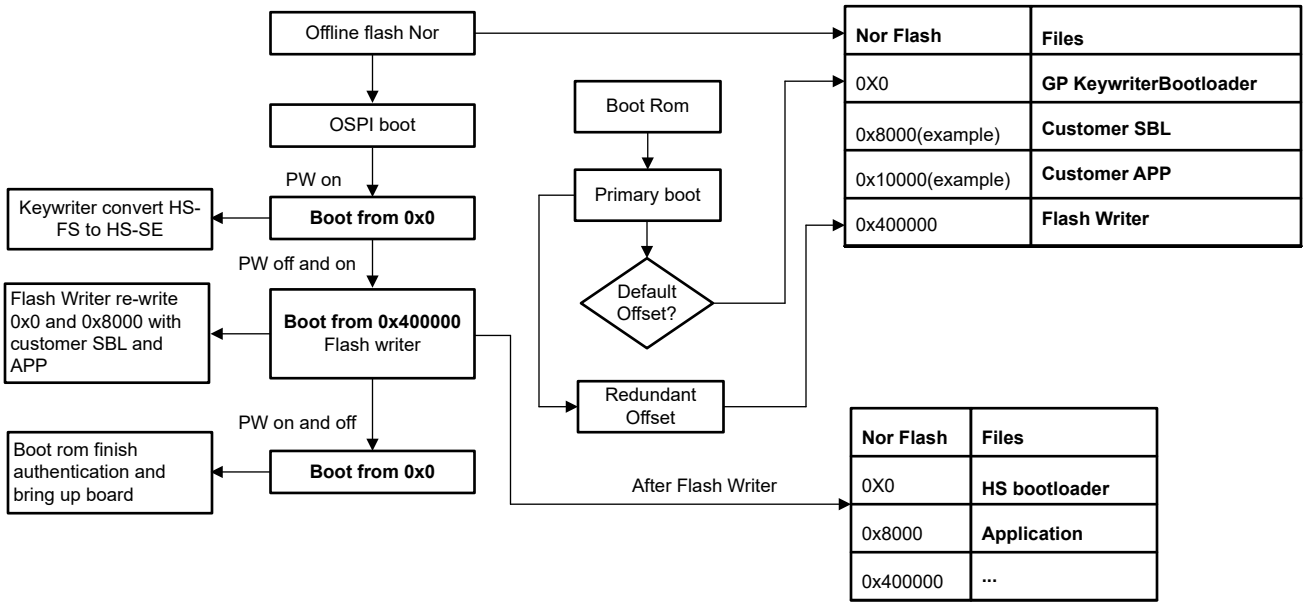

Figure 3-3. Boot from Primary Boot Media

4 Summary

This application note describes the key-burning process commonly used by customers, specifically switching boot modes to burn the key and boot the business files normally.

This document also provides two feasible new approaches. These approaches eliminate the need for switching boot modes. Instead, the approaches leverage the redundant boot modes of the chip to achieve key-burning and business file booting with multiple power cycles. The key benefit of this approach is streamlined production processes and manual intervention. Furthermore, the hardware design from the customer no longer needs to support multiple boot modes, further simplifying the design and system BOM cost. Note that these are only suggestions and the customer must test them thoroughly.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025