Booting Overview

The HPS boot starts when CPU0 is released from reset (for example, on power up) and executes code in the internal boot ROM at the reset exception address. The bootROM code brings the SoC out of reset and into a known state. The boot process ends when the code in the boot ROM jumps to the next stage of the boot software. This next stage of boot software is referred to as the second stage boot loader. The second stage boot loader can be customized and is typically stored external to the HPS in a nonvolatile flash-based memory.

The processor can boot from the following sources:

- FPGA (HPS-to-FPGA bridge)
  - 1.8 V NAND flash memory
  - 3.0 V NAND flash memory
  - 1.8 V SD/MMC flash memory with external transceiver
  - 3.0 V SD/MMC flash memory with internal transceiver
  - 1.8 V quad SPI flash memory
  - 3.0 V quad SPI flash memory

The HPS boot supports indirect or direct execution of the second stage boot loader depending on the boot device. With indirect execution, the boot ROM code copies the second stage boot loader from the boot device into the on-chip RAM and jumps to it. Indirect execution is used for flash memory boot sources. With direct execution, the boot ROM code jumps to the second stage boot loader located in the FPGA fabric.

The SoC also supports secure boot, where the device is brought up in a secure mode and the boot loader and user software code can be validated through authentication and decryption.

FPGA Configuration Overview

Configuration of the FPGA portion of the device starts when the FPGA portion is released from the reset state (for example, after power is applied power-on). The control block (CB) in the FPGA portion of the device is responsible for obtaining an FPGA configuration image and configuring the FPGA. The FPGA
configuration ends when the configuration image has been fully loaded and the FPGA enters user mode. The FPGA configuration image is provided by users and is typically stored in non-volatile flash-based memory. The FPGA CB can obtain a configuration image from the HPS through the FPGA manager or from any of the sources supported by the Arria 10 FPGAs family.

The following three figures illustrate the possible HPS boot and FPGA configuration schemes. The arrows in the figures denote the data flow direction.

**Boot and Configuration Options**

The SoC may boot in one of three ways:
- HPS boot and FPGA configuration occurs independently.
- HPS boots first and configures the FPGA.
- The FPGA is configured first and the HPS boots from the FPGA.

In the figure below, the FPGA configuration and HPS boot can occur independently. The FPGA configuration obtains its configuration image from a non-HPS source, while the HPS bootROM obtains its second stage boot loader from a non-FPGA fabric source.

*Figure A-1: Independent FPGA Configuration and HPS Booting*

In the figure below, the FPGA is configured first through one of its non-HPS configuration sources and then the HPS executes the second stage boot loader from the on-chip RAM in the FPGA from the FPGA fabric. The HPS boot waits for the FPGA fabric to be powered on and in user mode before executing. The HPS bootROM code executes the second stage boot loader from the FPGA fabric over the HPS-to-FPGA bridge. The second stage boot loader can be obtained from the FPGA RAM or by accessing an external interface, depending on your design and implementation.
In the figure below, the HPS boots first through one of its non-FPGA fabric boot sources. The FPGA does not need to be configured in order for the HPS to boot. However, the FPGA must be in a power on state for the HPS to reset properly and for the second stage boot loader to configure the FPGA flip-flops (IOCSR) of the HMC (Hard Memory Controller) in the FPGA. The software on the HPS obtains the FPGA configuration image from any of its flash memory devices or communication interfaces, for example, the Ethernet media access controller (EMAC). The software is provided by users and the boot ROM is not involved in configuring the FPGA fabric.

Boot Stages

Booting of the HPS is a multi-stage process. Each stage is responsible for loading the next stage. The first software stage is the bootROM. The bootROM locates and executes the second stage boot loader. The second stage boot loader locates and executes the next stage software and so on. The second stage bootloader and subsequent software stages (if present) are collectively referred to as user software.

Only the bootROM code is located in the HPS. Subsequent software is located external to the HPS and is provided by users. The boot ROM code is only aware of the second stage boot loader and not aware of any potential subsequent software stages. Before the control is passed to the second stage boot loader, it can be
decrypted and/or authenticated if a secure boot is enabled. Subsequent software stages are determined by
the user code.

The figure below illustrates the typical boot flow. However, there may be more or less software stages in the
user software than shown and the roles of the software stages may vary.

Figure A-4: Typical Boot Flow

Boot Definitions

Reset

The boot process begins when a CPU in the MPU exits from the reset state. When a CPU exits from reset,
it starts running code at the reset exception address. The bootROM is mapped at the reset exception address
so the CPU starts executing codes in the boot ROM.

Cold reset is triggered by the cold reset pin and warm reset is triggered by either the warm reset pin or by
software. With warm reset, some software registers are preserved and the boot process may skip some steps
depending on software settings.

BootROM

The boot ROM is located on the on chip ROM (0xFFF0000 - 0xFFFFDFFFF, 128kB). The function of the
boot ROM code is to determine the boot source, initialize the HPS after a reset, and jump to the second stage
boot loader. In the case of indirect execution, the boot ROM code loads the second stage boot loader image
from the flash memory to on-chip RAM. The boot ROM performs the following actions to initialize the
HPS:

- Enable instruction cache, branch predictor, floating point unit, NEON vector unit
- Sets up the level 4 (L4) watchdog 0 timer
- Configures the main PLL and peripheral PLL based on the CLKSEL value
- Initializes the flash controller to default settings

The BootROM also checks the security level of the SoC device and determines if a secure boot must be
executed. If secure boot is required, then the second stage boot loader image must be authenticated and
encrypted. When booting from flash memory, the boot ROM code uses the top 4 KB of the on-chip RAM
as data workspace. This area is reserved for the bootROMcode after a reset until the bootROMcode passes
software control to second stage boot loader. This limits the maximum size of the second stage boot loader
for indirect execution to 60 KB.

For a warm boot or cold boot from FPGA, the boot ROM code does not reserve the top 4KB of the on-chip
RAM, and the user may place user data in this area without being overwritten by boot ROM. The boot ROM
code only runs on CPU0. CPU1 is held in reset while boot ROM executes on CPU0. When CPU0 exits the
boot ROM code and starts executing user software, the boot ROM access is disabled. The user software in
CPU0 needs to map the user software exception vectors at 0x0 (which is previously mapped to boot ROM
exception vectors) and release CPU1 from reset. When CPU1 is released from reset, CPU1 executes the user
software exception instead of boot ROM.
Boot Select

The boot source is determined by the fpga boot fuse bit and the BSEL pins. If the fpga boot fuse is blown, the HPS can only boot from the FPGA. All CSEL and BSEL pins are ignored. This allows the HPS to boot from encrypted user-code in the FPGA. If the fpga boot fuse is not blown, then the boot source is determined according to the BSEL pins. If the boot source is the FPGA, the boot ROM does not configure the HPS I/Os. If the BSEL pins are used for determining the boot source, then the following table shows the flash devices assigned to each encoding. Note that if the value is 0x1, then the boot source is from the FPGA fabric.

<table>
<thead>
<tr>
<th>BOOTSEL Field Value</th>
<th>Flash Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>Reserved</td>
</tr>
<tr>
<td>0x1</td>
<td>FPGA (HPS-to-FPGA bridge)</td>
</tr>
<tr>
<td>0x2</td>
<td>1.8 V NAND flash memory</td>
</tr>
<tr>
<td>0x3</td>
<td>3.0 V NAND flash memory</td>
</tr>
<tr>
<td>0x4</td>
<td>1.8 V SD/MMC flash memory with external transceiver</td>
</tr>
<tr>
<td>0x5</td>
<td>3.0 V SD/MMC flash membrane with internal transceiver</td>
</tr>
<tr>
<td>0x6</td>
<td>1.8 V SPI or quad SPI flash memory</td>
</tr>
<tr>
<td>0x7</td>
<td>3.0 V SPI or quad SPI flash memory</td>
</tr>
</tbody>
</table>

Clock Select

If the FPGA boot fuse is not blown, the clock setting (CSEL) pins are used to configure the main PLL and peripheral PLL. Boot ROM reads the clock select encoding from the Security Manager. The main PLL can be sourced from EOSC1 or internal oscillator, depending on the fuse data from security manager. The peripheral PLL uses the same clock source as main PLL. Based on the clock source and clock select settings, boot ROM configures the main PLL and peripheral PLL parameters and the clock dividers for clocks derived from the PLLs.

BootROM will not configure the PLLs for the following conditions:
- Boot from RAM (warm reset)
- Boot from FPGA
- Clock select is set to bypass mode

I/O Configuration

The flash devices needed for booting are connected to the HPS dedicated I/Os. Boot ROM configures these I/Os depending on the flash device selected by boot select setting. To configure the I/Os, boot ROM performs pin muxing and pin configuration on these I/Os.

The boot ROM is allocated 15 dedicated I/Os. There are three separate I/O tables in the boot ROM, one for each flash device.

On cold reset, the Boot ROM will always set the pin muxes for these dedicated I/Os to connect to the selected flash device. On warm reset, boot ROM will not change the pin muxes if user enables the warmrstcfgpinmux field of the ROM code control register (romcode_ctrl) in the system manager.
The boot ROM also configures the electrical behavior and direction of each of the dedicated I/Os. On cold reset, boot ROM always perform pin configuration. On warm reset, boot ROM will not change the pin muxes if user enables the \texttt{warmrstcfgio} field of the ROM code control register (\texttt{romcode_ctrl}) in the system manager.

**L4 Watchdog**

The boot ROM enables the L4 watchdog early in the boot process. This watchdog is reserved for boot ROM until boot loader indicates that it has started correctly and taken control of the exception vectors. The timeout is at least one second, depending on the clock select setting. As the watchdog is reset just before the control passes to boot loader, the boot loader needs to reset the watchdog when the boot loader starts.

**Second Stage Boot Loader**

The function of the second stage boot loader is user-defined. However, typical functions include initializing the SDRAM interface and configuring the HPS I/O pins. Initializing the SDRAM allows the second stage boot loader to load the next stage of the boot software (that might not fit in the 60 kilobytes (KB) available in the on-chip RAM). A typical next software stage is the open source boot loader, U-boot. The second stage boot loader is allowed to load the next boot software stage from any device available to the HPS. Typical sources include the same flash device that contains the second stage boot loader, a different flash device, or a communication interface such as an EMAC.

**Boot Loader**

The boot loader loads the operating system and passes software control to the operating system.

**Secure Boot**

The Boot ROM supports both nonsecure and secure boot. Secure boot allows the boot ROM to reset with the CPU in a secure mode, and validate the authenticity and integrity of the boot loader prior to execution. Secure boot ensures that no unauthorized boot loader can execute. The boot loader can be stored in an encrypted state in an external device and be decrypted directly into the HPS on-chip RAM. This prevents the code from running on devices that do not contain the encryption key, and prevents unauthorized examination of the boot loader code.

The boot ROM determines the security level based on user settings from fuse data in security manager. There is also an optional security header in the flash device. Based on the merged data from security manager and security header, boot ROM loads the boot loader image based on the following settings:

- **Clear** - No authentication or decryption is required
- **Secure Authenticated** - Authentication but no decryption required
- **Secure Encrypted** - Decryption but no authentication required
- **Secure** - Both authentication and decryption required

The authentication process is independent of the decryption process. However, if authentication and decryption are both required, authentication is done before the decryption process.
# Document Revision History

## Table A-1: Document Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2014</td>
<td>2014.08.18</td>
<td>Initial release</td>
</tr>
</tbody>
</table>

---