



Boot and Configuration

Introduction

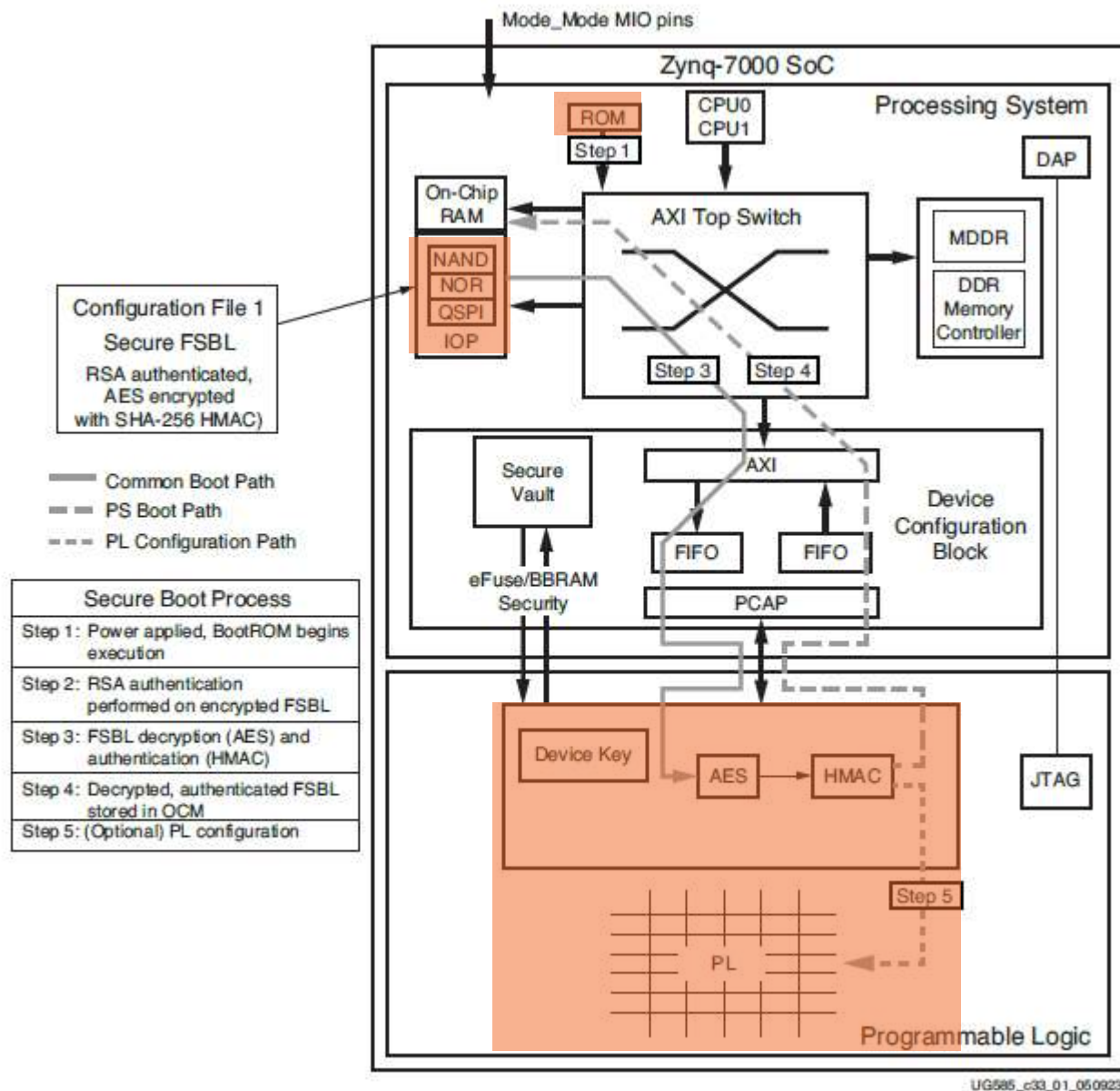
Immediately after the PS_POR_B reset pin deasserts, the hardware samples the boot strap pins and optionally enables the PS clock PLLs. Then, the PS begins executing the BootROM code in the on-chip ROM to boot the system. The POR resets the entire device with no previous state saved. The non-POR type resets also cause the BootROM to execute, but without the hardware sampling the strap pins. After a non-POR reset, some registers values are preserved and the device is aware of its previous security mode. Non-POR resets include the PS_SRST_B pin and several internal reset sources.

The BootROM is the first software to run in the APU. The BootROM executes on CPU 0 and CPU 1 executes the wait-for-event (WFE) instruction. The main tasks of the BootROM are to configure the system, copy the Boot Image FSBL/User code from the boot device to the OCM, and then branch the code execution to the OCM. Optionally, the FSBL/User code can be executed directly from a Quad-SPI or NOR device in a non-secure environment.

The PS Master boot device holds one or more boot images. A boot image is made up of the BootROM Header (also referred to as the Boot Image Header) and the first stage boot loader (FSBL). The boot device can also hold a bitstream to configure the PL and an embedded operating system, but these are not accessed by the BootROM code. The flash memory device for boot can be Quad-SPI, NAND, NOR, or SD card.

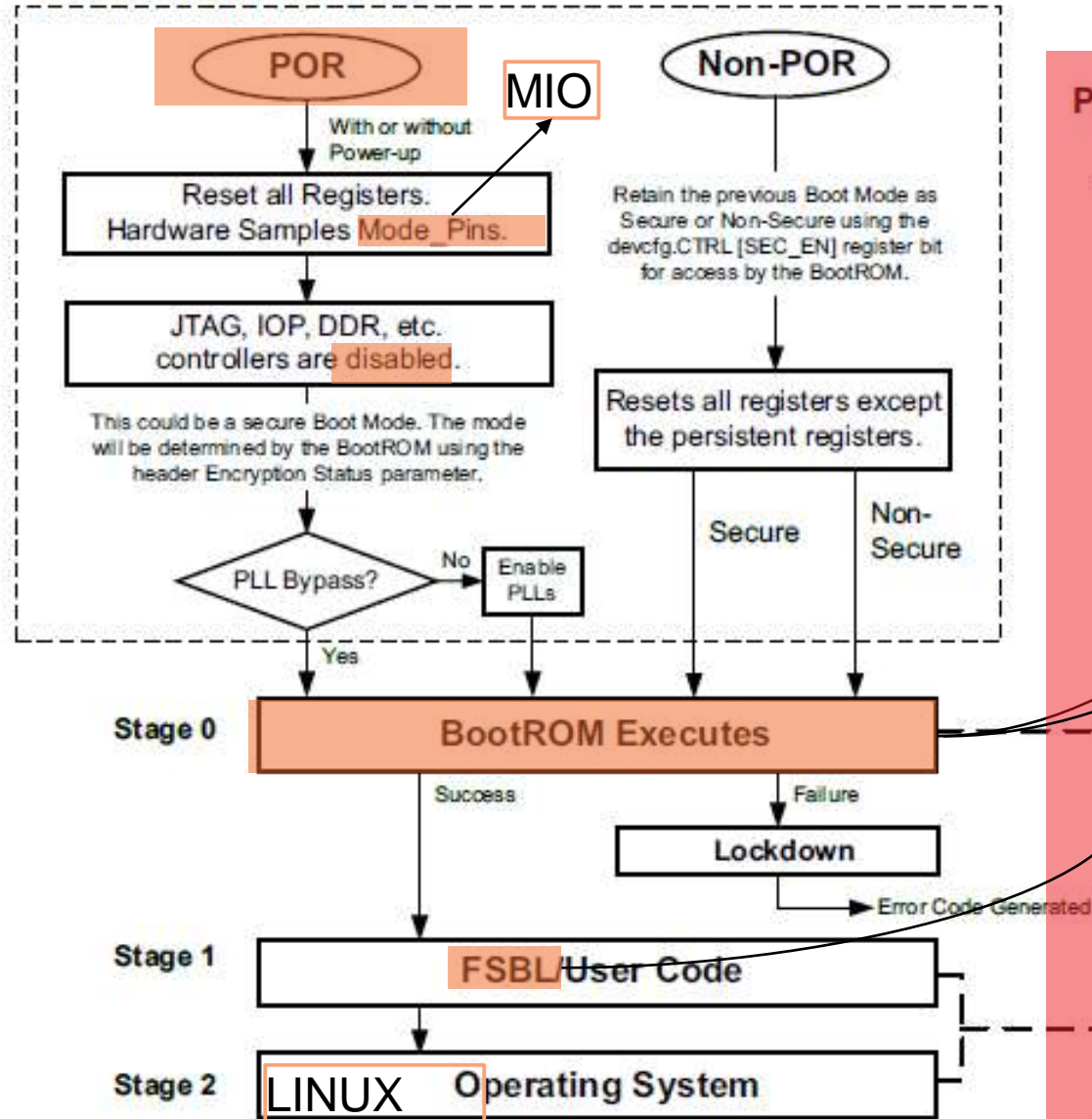
The BootROM execution flow is affected by the pin strap settings, the BootROM Header, and what the BootROM code discovers about the system. The BootROM can execute in a secure environment with encrypted FSBL/User code, or a non-secure environment. After the BootROM executes, the FSBL/User code takes responsibility of the system as described in the *Zynq-7000 SoC Software Developers Guide* (UG821).

For development, the system can be booted in JTAG mode. Or, JTAG can be enabled after a non-secure flash device boot. JTAG always implies a non-secure environment, but it allows for access to the Arm debug access port (DAP) controller in the CPU complex (APU) and the AMD test access port (TAP) controller in the PL.



UG585_c33_01_060923

PS Hardware Functions



PL Timeline

Start-up (Power-up)

The PL hardware includes a self-startup sequence to prepare it for initialization by the BootROM or User code.

Initialize

The PL must be powered up for Secure mode or if the JTAG interface is required.

Initialize, Configure, Enable

The FSBL/User/Application code can clear, program and enable the PL.

做最基礎的初始化

從非一次性暫存器讀取代碼，讀去到OCM進行運行。

對基本外部設備的初始化，也會從FSBL讀取資料，放置到OCM進行運行

FSBL (first stage bootloader)

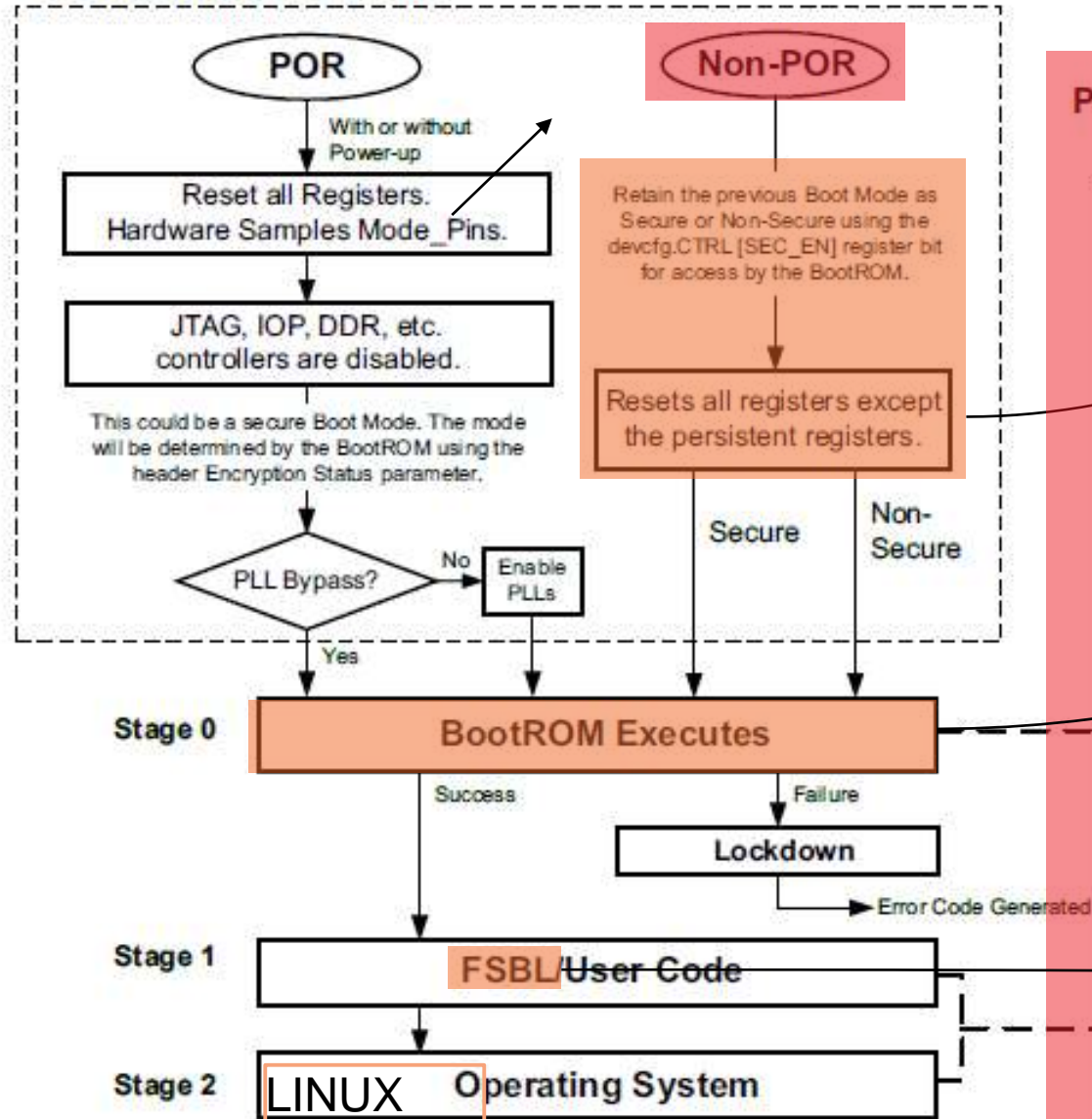
對更多的外部設備初始化，例如DDR等等

初始化之後，會讀取BINSTREAM，配置PL的LOGIC，USER CODE是我們的應用代碼。

完成後進入下個階段

UG586_c6_01_081514

PS Hardware Functions



PL Timeline

Start-up (Power-up)

The PL hardware includes a self-startup sequence to prepare it for initialization by the BootROM or User code.

Initialize

Ex, Flash

The PL must be powered up for Secure mode or if the JTAG interface is required.

Initialize, Configure, Enable

The FSBL/User/Application code can clear, program and enable the PL.

UG585_c6_01_081514

與POR不同，不會去讀取復位PIN，會復位大部分的暫存器(也是會有暫存器不會復位)

從非一次性暫存器讀取代碼，讀去到OCM進行運行。

對基本外部設備的初始化，也會從FSBL讀取資料，放置到OCM進行運行

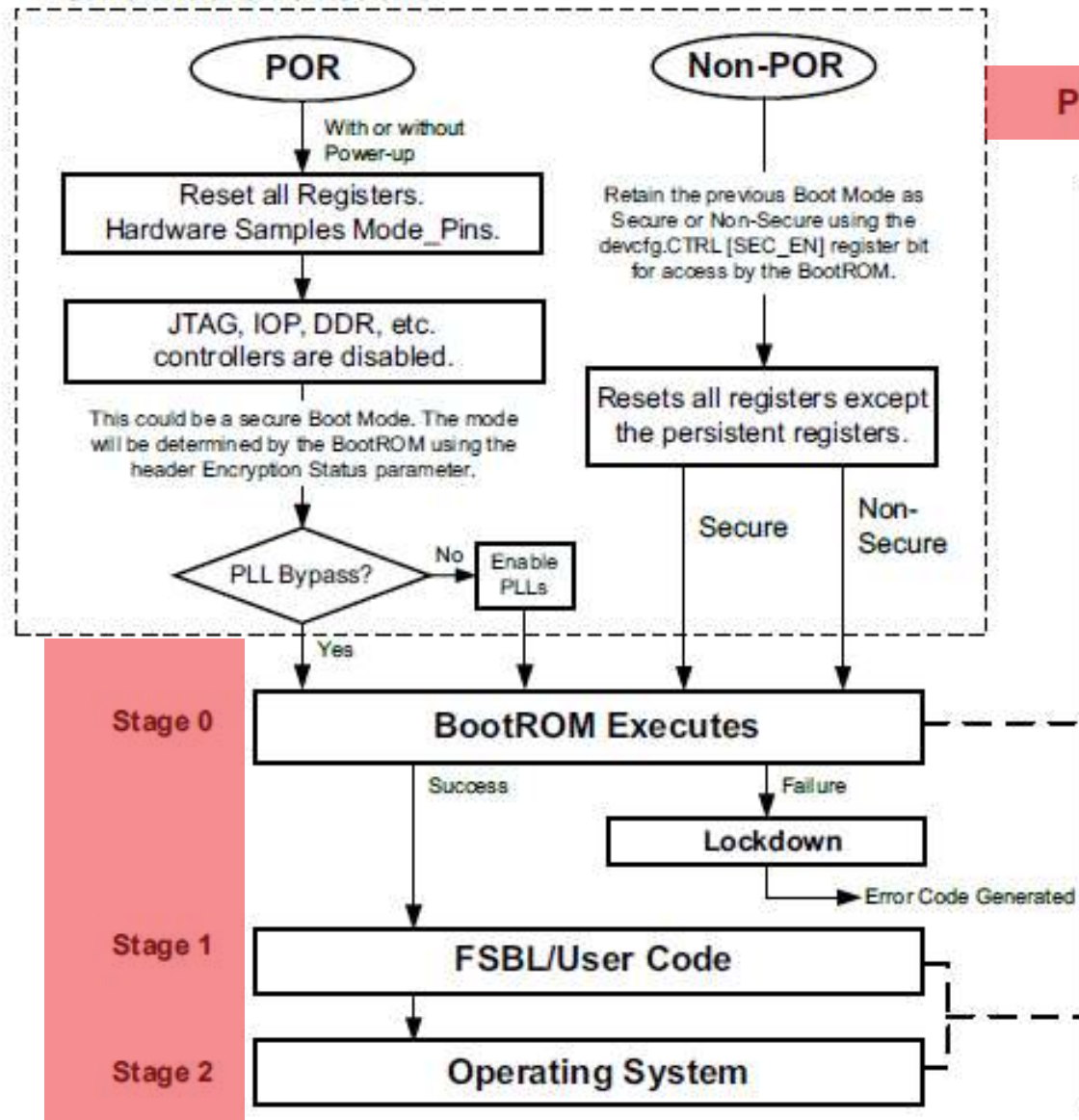
FSBL (first stage bootloader)

對更多的外部設備初始化，例如DDR等等

初始化之後，會讀取BINSTREAM，配置PL的LOGIC，USER CODE是我們的應用代碼。

完成後進入下個階段

PS Hardware Functions



PL Timeline

Start-up (Power-up)

The PL hardware includes a self-startup sequence to prepare it for initialization by the BootROM or User code.

準備的狀態

Initialize

The PL must be powered up for Secure mode or if the JTAG interface is required.

在安全模式下，我們PL必須上電完成，AES 解密認證是在PL上實現的，須等待PL設定完成。

Initialize, Configure, Enable

The FSBL/User/Application code can clear, program and enable the PL.

UG585_c6_01_081514

BootROM and Header Parameters

The BootROM header includes a dozen parameters that guide the BootROM execution flow. For example, the header includes a parameter to select the security mode: the Encryption Status parameter. In secure mode, the FSBL/User code, bitstream, and other software are encrypted. The BootROM has the ability to authenticate and decrypt the encrypted FSBL/User code. The header itself is never encrypted.

As another example, the header includes the Length of Image parameter that defines the length of the FSBL/User code that the BootROM loads into the OCM for execution. This code is limited to 192 KB in length. This parameter can be set to zero to indicate the desire to execute code directly

from the boot device (execute-in-place). All of the header parameters are described in section [6.3.2 BootROM Header](#).

The last two functions of the BootROM are to disable access to its ROM code and transfer CPU code execution to the FSBL/User code. The execution of the BootROM is detailed in section [6.3.1 BootROM Flowchart](#).

Secure PS Images and PL Bitstreams

基于哈希的消息认证码

The secure environment starts with an encrypted boot process where the PS software acts as the system master and the BootROM reads an encrypted FSBL/user code image from the selected flash memory device and processes it using the hardened, PL based Hash-based Message Authentication Code (HMAC) and an Advanced Encryption Standard (AES) module with a Cipher Block Chaining Mode (CBC). These modules are accessed from the PS through the DevC interface and the downstream Processor Configuration Access Port (PCAP) located in the PL.

6.1.2 PS Software Boot Stages

The PS software boot process is controlled by the BootROM and then the FSBL/User code. The BootROM code operation is influenced by the boot strap pins, the BootROM Header, and what the BootROM code detects in the system.

Stage 0 (BootROM: BootROM Header)

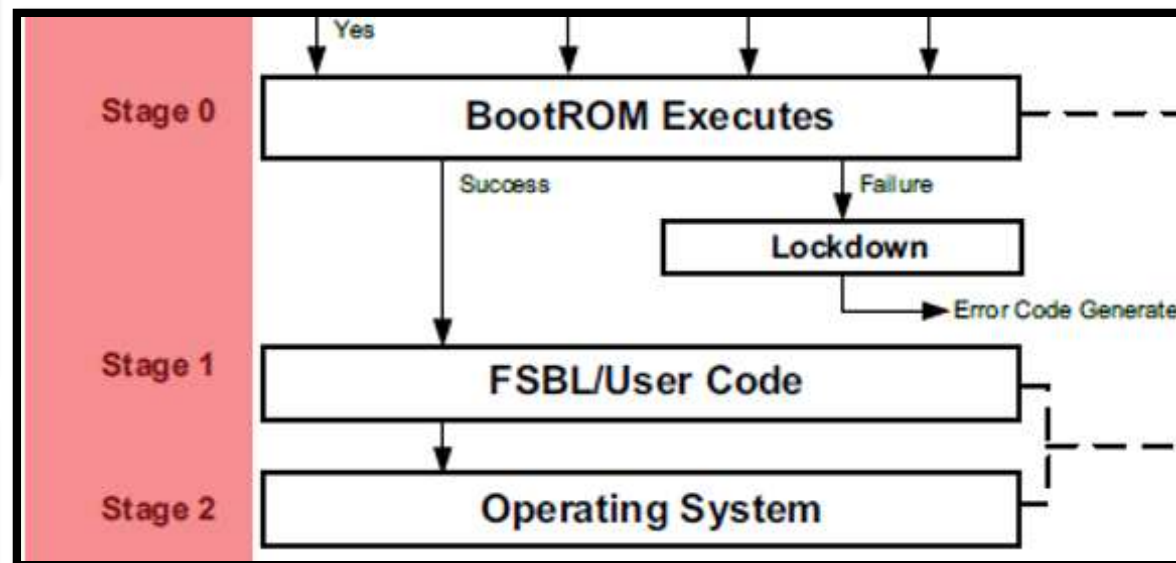
Hard-coded BootROM executes on the primary CPU (CPU 0) after a **power-on reset (POR)** or non-POR system reset (PS_SRST_B, debug, watchdog, software). The BootROM reads the BootROM Header programmed into the boot flash device to determine the boot flow and transitions to stage 1. After the hardware boot sequence, both CPUs start executing the same BootROM code

Stage 1 (FSBL/ User code)

This is generally the First Stage Boot Loader, but it can be any user-controlled code. Refer to [UG821, Zynq-7000 SoC Software Developers Guide](#) for details about the FSBL.

Stage 2 (U-Boot / System / Application)

This is generally the system software, but it could also be a second stage boot loader (SSBL). This stage is also completely within user control and is not described in this chapter. Refer to [UG821, Zynq-7000 SoC Software Developers Guide](#) for details about FSBL and stage 2 images.



6.1.3 Boot Device Content

The boot device can store multiple components and multiple versions of the components:

- BootROM Header (required by BootROM)
- FSBL/User code ELF file (required by BootROM)
- PL Bitstream (not accessed by BootROM)
- System/Application ELF file (not accessed by BootROM)

The BootROM Header is detailed in section [6.3.2 BootROM Header](#). The FSBL/User code requirements are described in [UG821](#), *Zynq-7000 SoC Software Developers Guide*.

6.2.5 Boot Mode Pin Settings

There are 7 boot mode strapping pins that are hardware programmed on the board using MIO pins [8:2]. They are sampled by the hardware soon after PS_POR_B deasserts and their values are written to software readable registers for use by the BootROM and user software. The board hardware must connect each strapping pin, MIO [8:2], to a 20 k Ω pull-up or pull-down resistor. The encoding of the mode pins are shown in Table 6-4. A pull-up resistor specifies a logic 1 and a pull-down resistor specifies a logic 0.

Five pins, BOOT_MODE[4:0], are used to select the boot mode, JTAG chain config, and if the PLLs are bypassed. The sampled values of these pins are written into the slcr.BOOT_MODE [BOOT_MODE] and [PLL_BYPASS] bit fields.

- Boot modes are explained in section 6.3 BootROM Code.
- Boot strap pins are listed in Table 6-4.
- JTAG chains are described in section 6.4.5 PL Control via User-JTAG.
- PLLs are described in section 6.2.3 Clocks and PLLs.

Boot Mode MIO Strapping Pins

Table 6-4: Boot Mode MIO Strapping Pins

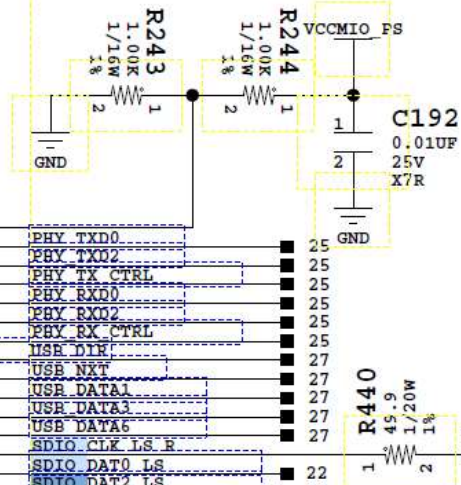
Pin-signal / Mode	MIO[8]	MIO[7]	MIO[6]	MIO[5]	MIO[4]	MIO[3]	MIO[2]
	VMODE[1]	VMODE[0]	BOOT_MODE[4]	BOOT_MODE[0]	BOOT_MODE[2]	BOOT_MODE[1]	BOOT_MODE[3]
Boot Devices							
JTAG Boot Mode; cascaded is most common ⁽¹⁾				0	0	0	JTAG Chain Routing ⁽²⁾ 0: Cascade mode 1: Independent mode
NOR Boot ⁽³⁾				0	0	1	
NAND				0	1	0	
Quad-SPI ⁽³⁾				1	0	0	
SD Card				1	1	0	
Mode for all 3 PLLs							
PLL Enabled			0	Hardware waits for PLL to lock, then executes BootROM.			
PLL Bypassed			1	Allows for a wide PS_CLK frequency range.			
MIO Bank Voltage ⁽⁴⁾							
	Bank 1	Bank 0	Voltage Bank 0 includes MIO pins 0 thru 15. Voltage Bank 1 includes MIO pins 16 thru 53.				
2.5 V, 3.3 V	0	0					
1.8 V	1	1					

BANK 501 XC7Z020CLG484

VCCMIO_PS

A10 VCCO_MIO1_501_A10
B13 VCCO_MIO1_501_B13
D9 VCCO_MIO1_501_D9
E12 VCCO_MIO1_501_E12

PS MIO VREF 501 F8	F8	PHY_TXD0	25
PS MIO17 501 E9	E9	PHY_TXD2	25
PS MIO19 501 E10	E10	PHY_TX_CTRL	25
PS MIO21 501 F11	F11	PHY_RXD0	25
PS MIO23 501 E11	E11	PHY_RXD2	25
PS MIO25 501 F12	F12	PHY_RX_CTRL	25
PS MIO27 501 D7	D7	USER_D1R	25
PS MIO29 501 E8	E8	USER_D1R	27
PS MIO31 501 F9	F9	USER_NXT	27
PS MIO33 501 G13	G13	USER_DATA1	27
PS MIO35 501 F14	F14	USER_DATA3	27
PS MIO38 501 F13	F13	USER_DATA6	27
PS MIO40 501 E14	E14	SDIO_CLK_LS_R	27
PS MIO42 501 D8	D8	SDIO_DATA0_LS	22
PS MIO44 501 E13	E13	SDIO_DATA2_LS	22
PS MIO46 501 D12	D12	CAN_RXD_LS	21
PS MIO48 501 D11	D11	USER_UART_RX	36
PS MIO50 501 D13	D13	PS_SCL_MAIN	32
PS MIO52 501 D10	D10	PHY_MDC	25
PS SRST_B 501 C9	C9	PS_SRST_B	14
PS MIO16 501 D6	D6	PHY_TX_CLK	25
PS MIO18 501 A7	A7	PHY_TXD1	25
PS MIO20 501 A8	A8	PHY_TXD3	25
PS MIO22 501 A14	A14	PHY_RX_CLK	25
PS MIO24 501 B7	B7	PHY_RXD1	25
PS MIO26 501 A13	A13	PHY_RXD3	25
PS MIO28 501 A12	A12	USER_DATA4	27
PS MIO30 501 A11	A11	USER_STP	27
PS MIO32 501 C7	C7	USER_DATA0	27
PS MIO34 501 B12	B12	USER_DATA2	27
PS MIO36 501 A9	A9	USER_CLKOUT	27
PS MIO37 501 B14	B14	USER_DATA5	27
PS MIO39 501 C13	C13	USER_DATA7	27
PS MIO41 501 C8	C8	SDIO_CMD_LS	22
PS MIO43 501 B11	B11	SDIO_DATA1_LS	22
PS MIO45 501 B9	B9	SDIO_CD_DATA1_LS	22
PS MIO47 501 B10	B10	CAN_TXD_LS	21
PS MIO49 501 C14	C14	USER_UART_TX	36
PS MIO51 501 C10	C10	PS_SDA_MAIN	32
PS MIO53 501 C12	C12	PHY_MDIO	25



SDIO_CLK_LS

ZYNQ7 Processing System (5.5)

Documentation Presets IP Location Import XPS Settings

Page Navigator

- Zynq Block Design
- PS-PL Configuration
- Peripheral I/O Pins**
- MIO Configuration
- Clock Configuration
- DDR Configuration
- SMC Timing Calculation

Peripheral I/O Pins

Search: Q-

Peripherals

- ☒ Quad SPI Flash
 - ☒ Single SS 4bit IO
 - ☐ Dual Quad SPI(4bit)
 - ☐ Dual Quad SPI (8bit)

Bank 0 LVCMOS

0	1	2	3	4	5	6	7	8
Quad SPI Flash								
Single SS 4bit IO								
ss_b								
Dual Quad SPI (8bit)								

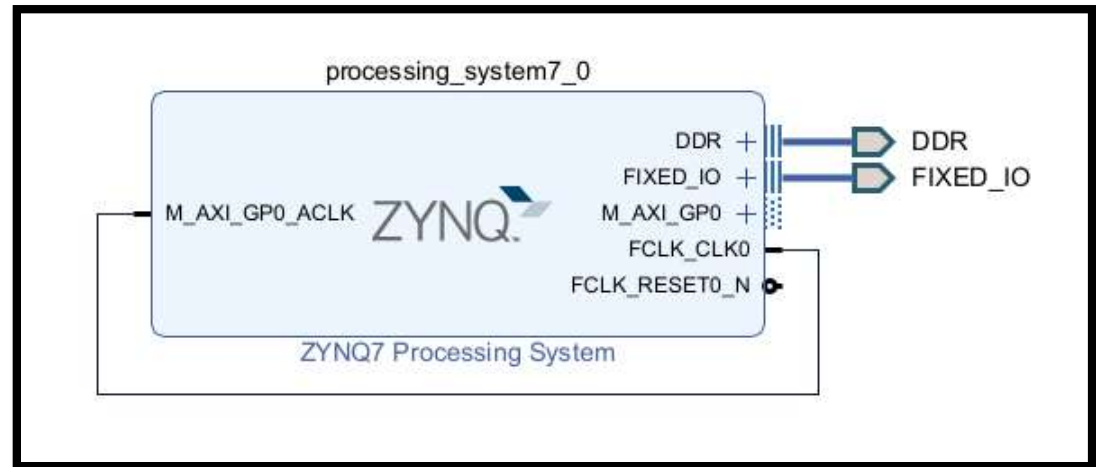
MIO Configuration

Bank 0 I/O Voltage LVCMOS 3.3V

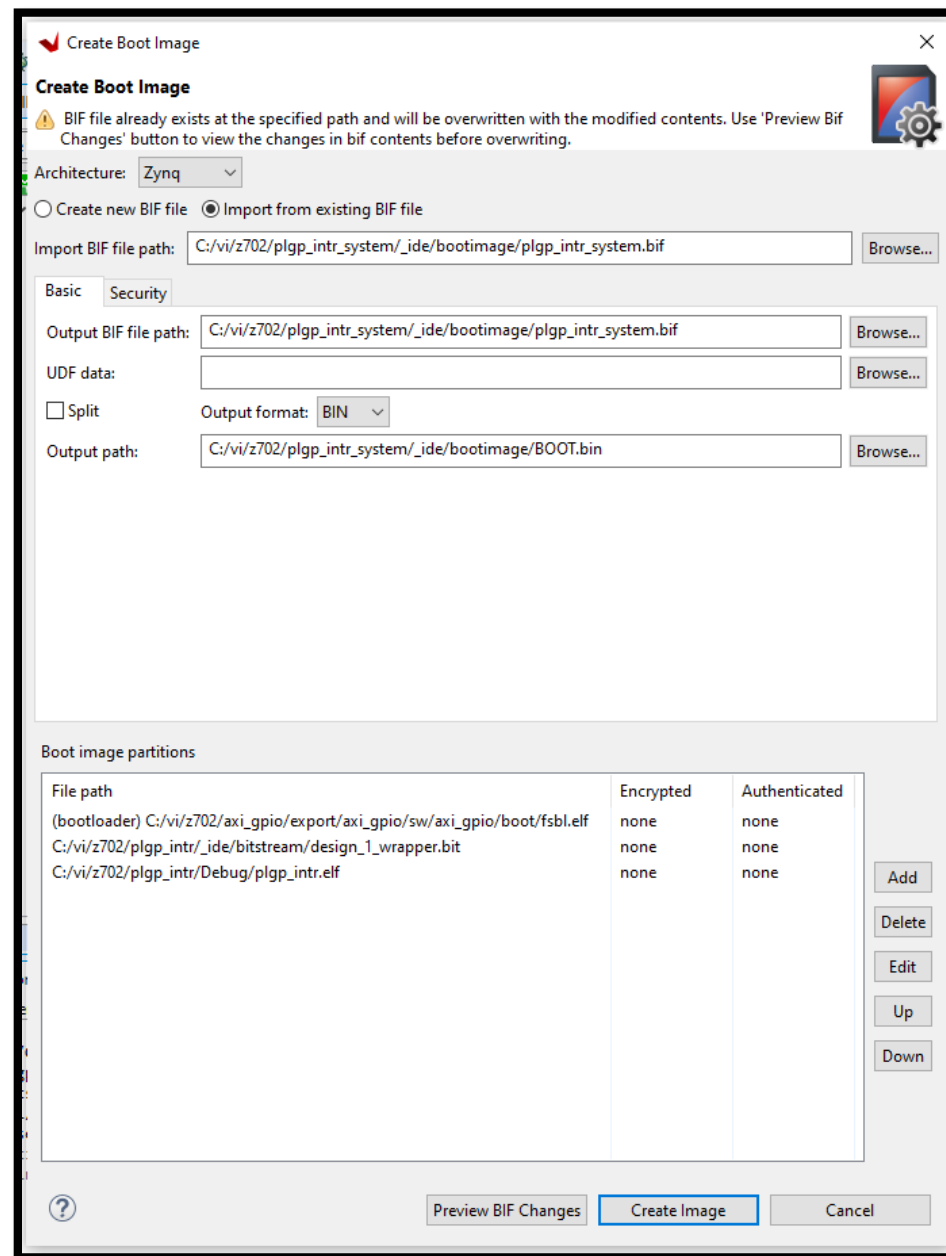
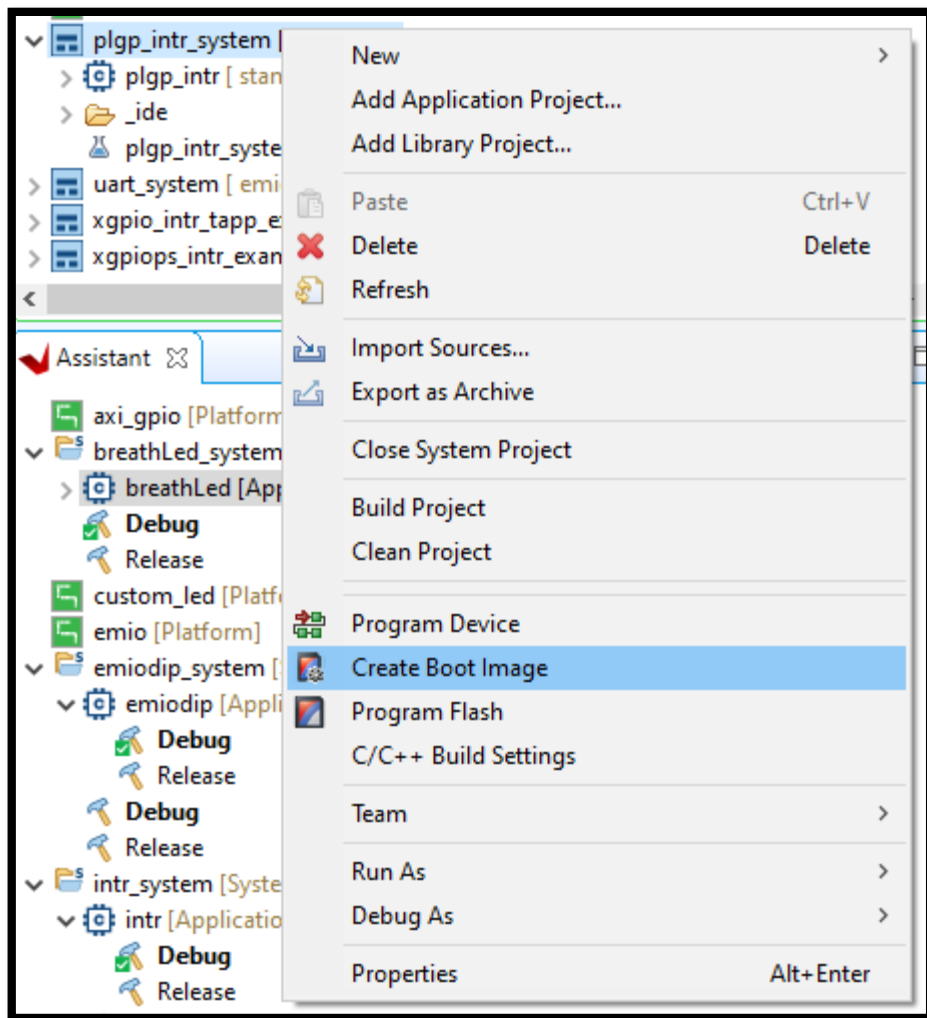
Bank 1 I/O Voltage LVCMOS 1.8V

PS_MIO38_501_F13	E14	SDIO_CLK_LS_R	27
PS_MIO40_501_E14	D8	SDIO_DAT0_LS	22
PS_MIO42_501_D8	E13	SDIO_DAT2_LS	22
PS_MIO44_501_E13	D12	CAN_RXD_LS	21
PS_MIO46_501_D12	D11	USB_UART_RX	36
PS_MIO48_501_D11	D13	PS_SCL_MAIN	32
PS_MIO50_501_D13	D10	PHY_MDC	25
PS_MIO52_501_D10	C9	PS_SRST_B	14
PS_SRST_B_501_C9	D6	PHY_TX_CLK	25
PS_MIO16_501_D6	A7	PHY_TXD1	25
PS_MIO18_501_A7	A8	PHY_TXD3	25
PS_MIO20_501_A8	A14	PHY_RX_CLK	25
PS_MIO22_501_A14	B7	PHY_RXD1	25
PS_MIO24_501_B7	A13	PHY_RXD3	25
PS_MIO26_501_A13	A12	USB_DATA4	27
PS_MIO28_501_A12	A11	USB_STP	27
PS_MIO30_501_A11	C7	USB_DATA0	27
PS_MIO32_501_C7	B12	USB_DATA2	27
PS_MIO34_501_B12	A9	USB_CLKOUT	27
PS_MIO36_501_A9	B14	USB_DATA5	27
PS_MIO37_501_B14	C13	USB_DATA7	27
PS_MIO39_501_C13	C8	SDIO_CMD_LS	22
PS_MIO41_501_C8	B11	SDIO_DAT1_LS	22
PS_MIO43_501_B11	B9	SDIO_CD_DAT3_LS	22
PS_MIO45_501_B9	B10	CAN_TXD_LS	22

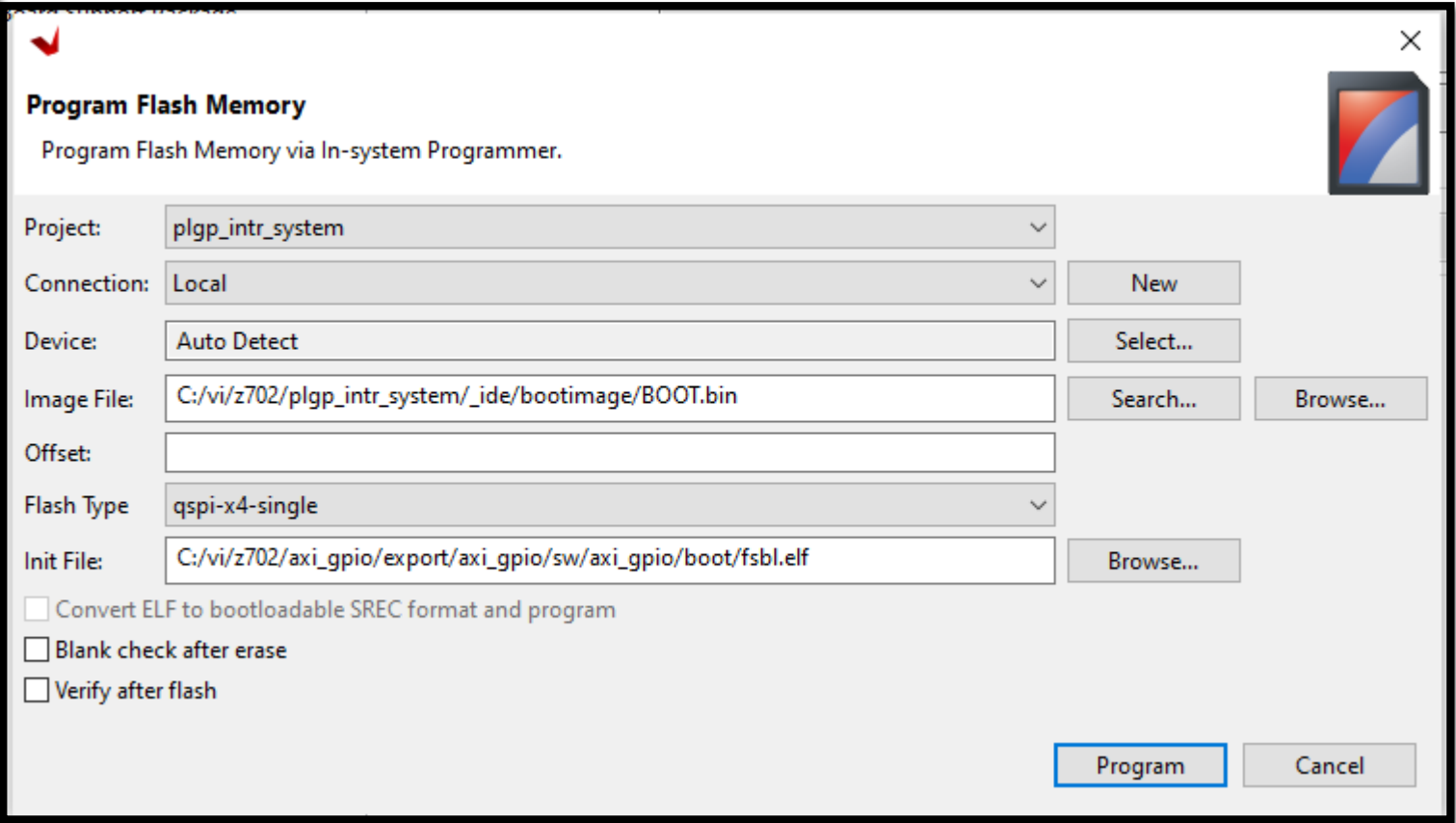
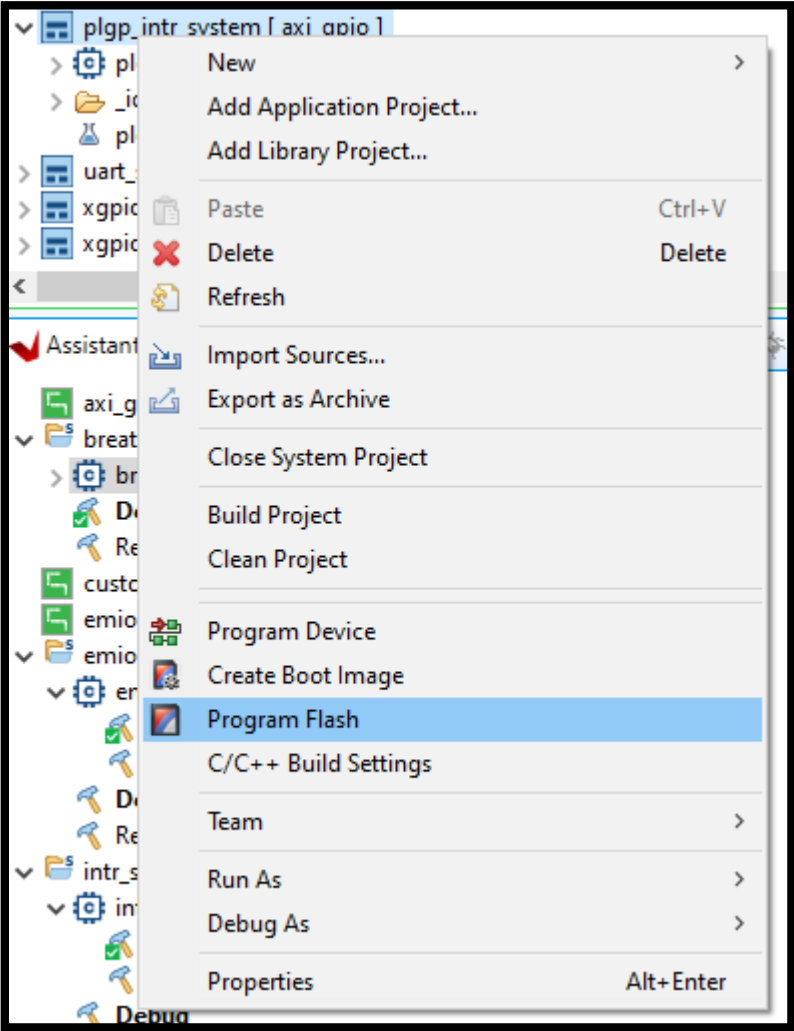
SD0



Create Boot Image



Program Flash



Reference

ug585

