Boot and Configuration

AMD
together we advance_

Introduction

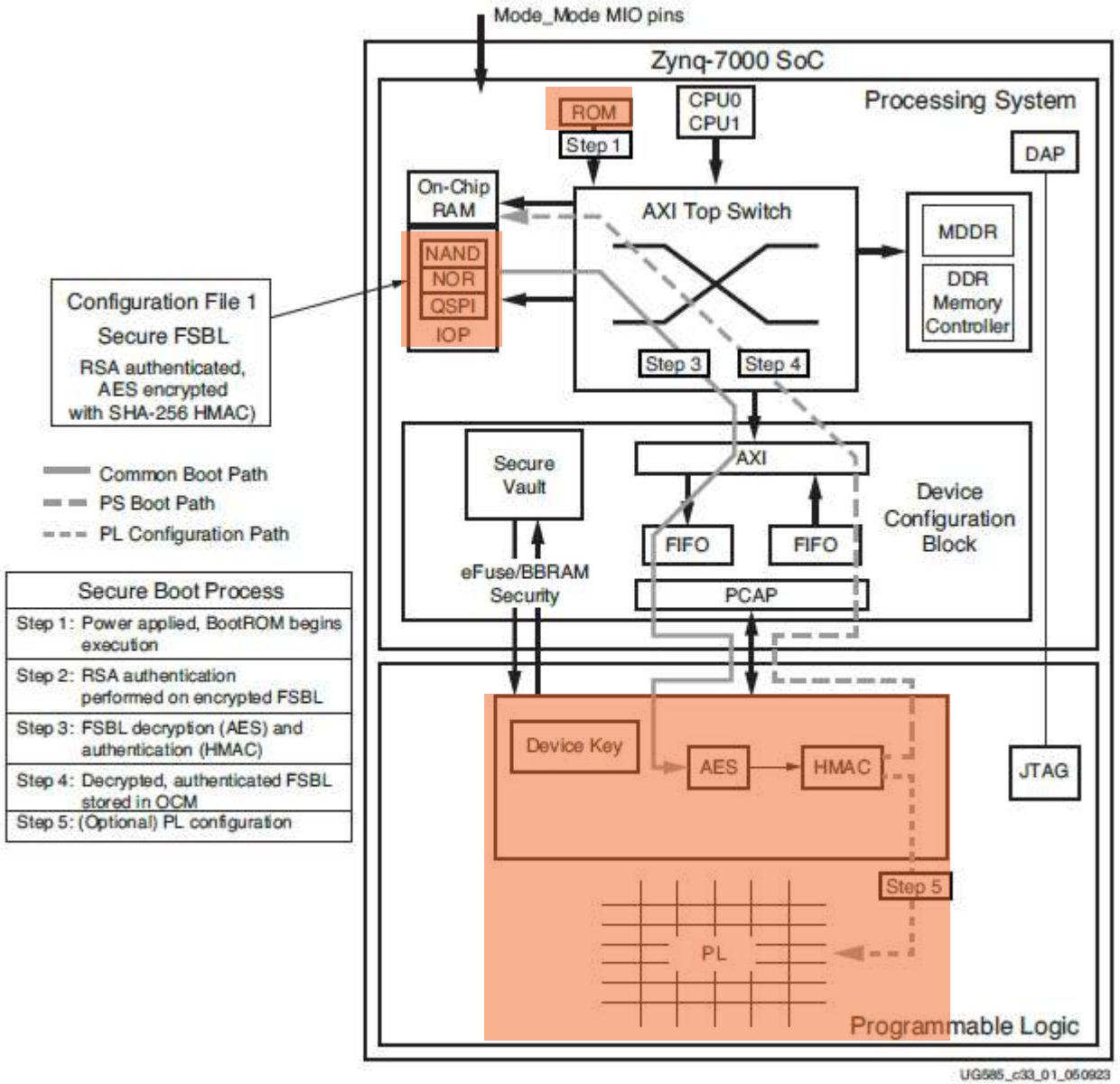
Immediately after the PS_POR_B reset pin deasserts, the hardware samples the boot strap pins and optionally enables the PS clock PLLs. Then, the PS begins executing the BootROM code in the on-chip ROM to boot the system. The POR resets the entire device with no previous state saved. The non-POR type 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.

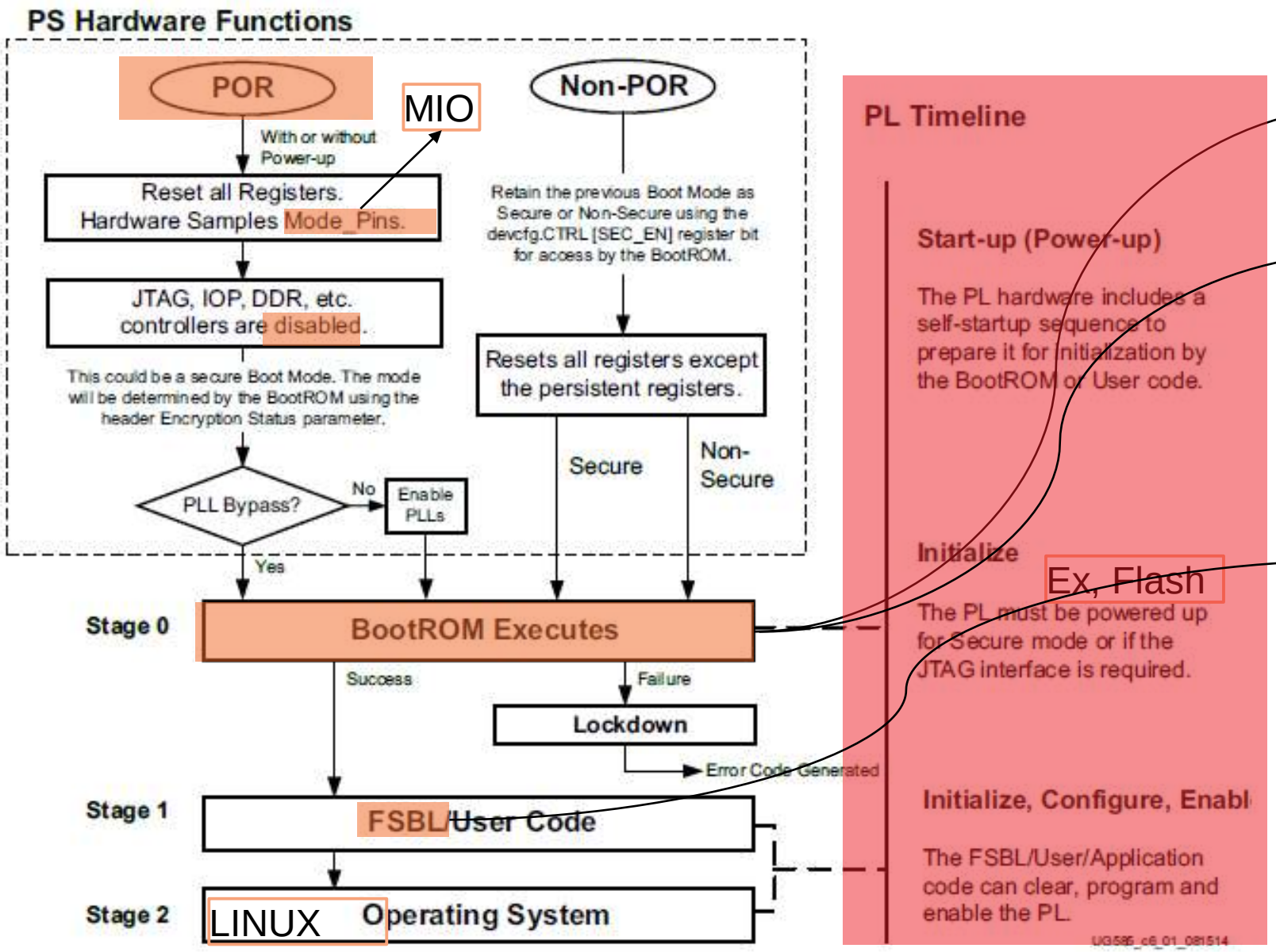
The BootROM is the first software to run in the APU. The BootROM executes on CPU 0 and CPU 1 executes the wait-for-event (WFE) instruction. The main tasks of the BootROM are to configure the system, copy the Boot Image FSBL/User code from the boot device to the OCM, and then branch the code execution 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.





做最基礎的初始化

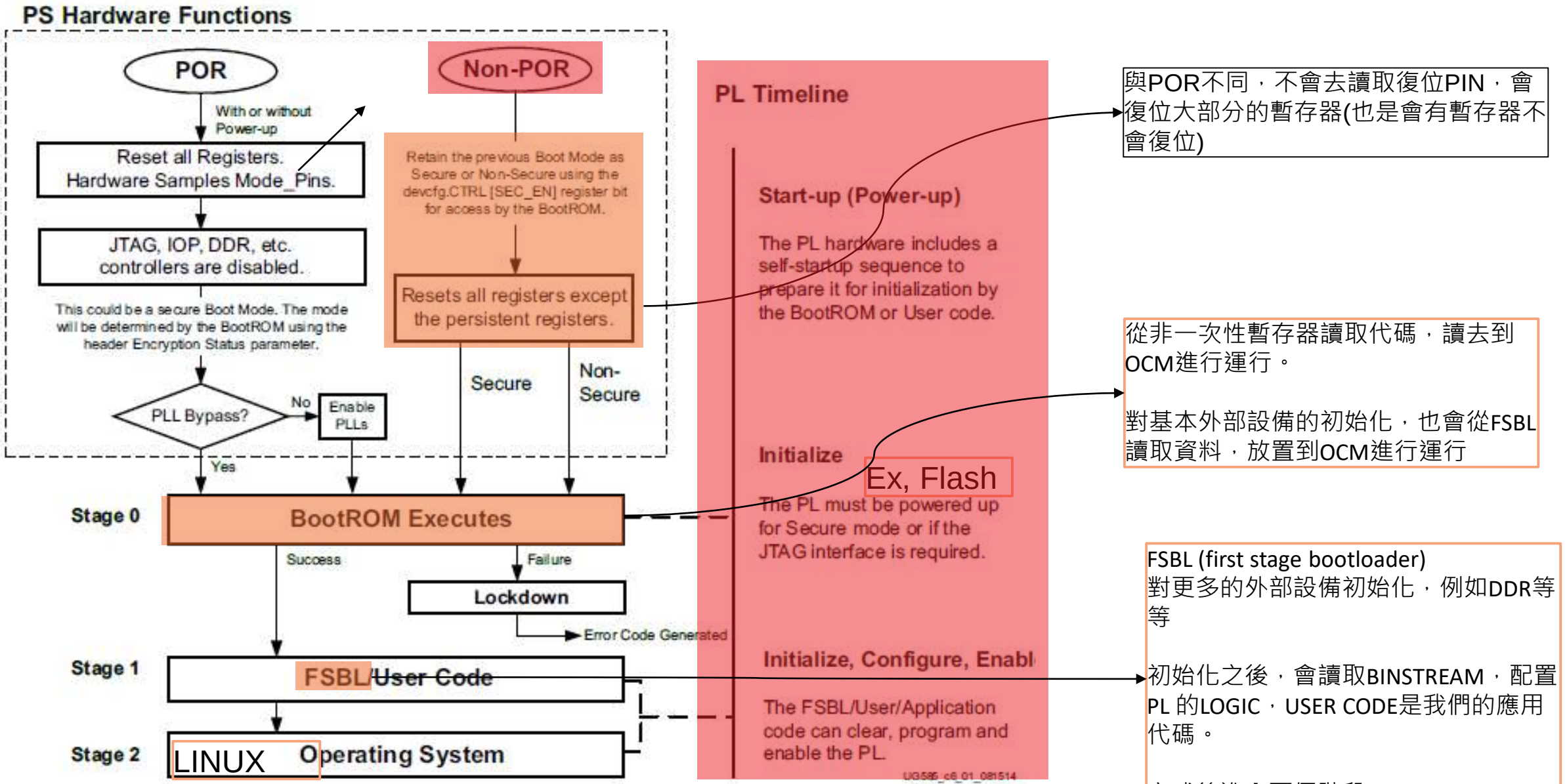
從非一次性暫存器讀取代碼，讀去到OCM進行運行。

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



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)

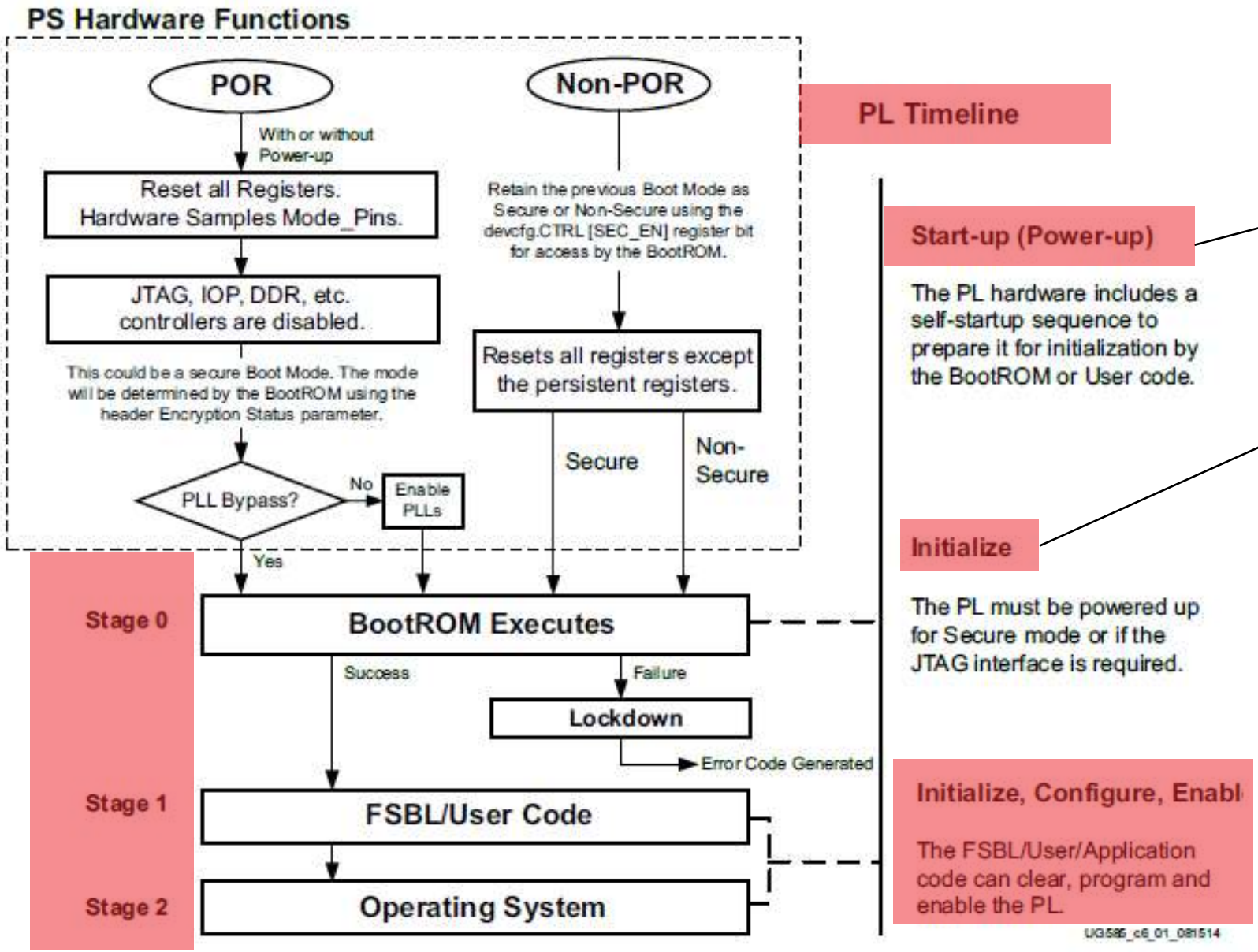
從非一次性暫存器讀取代碼，讀去到OCM進行運行。

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



準備的狀態

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。

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)

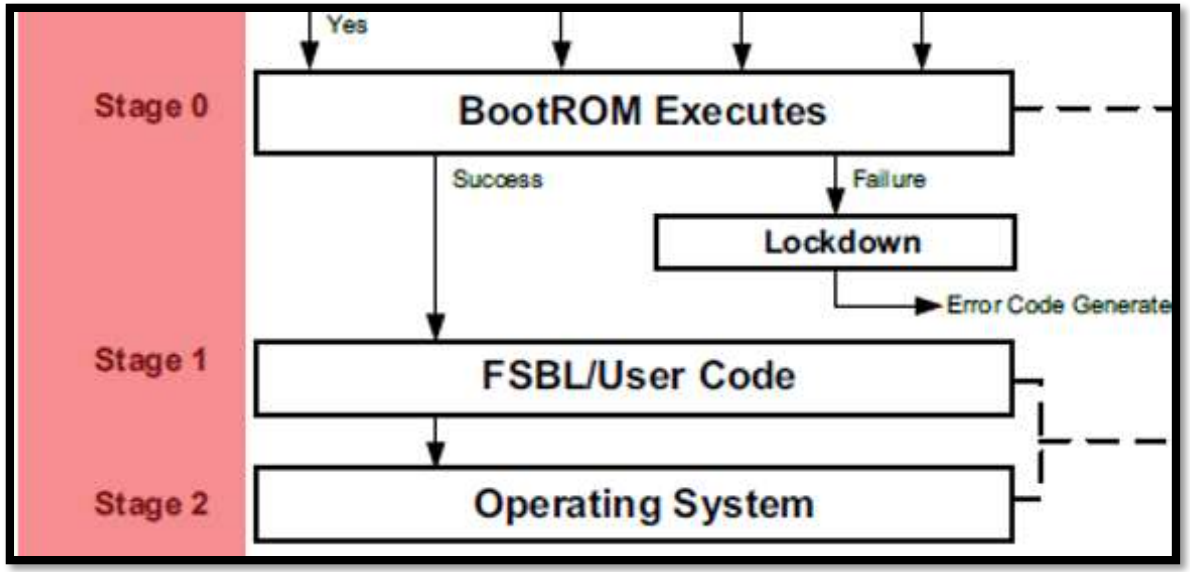
Hard-coded BootROM executes on the primary CPU (CPU 0) after a power-on reset (POR) or non-POR system reset (PS_SRST_B, debug, watchdog, software). The BootROM reads the BootROM Header programmed into the boot flash device to determine the boot flow and transitions to stage 1. After the hardware boot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

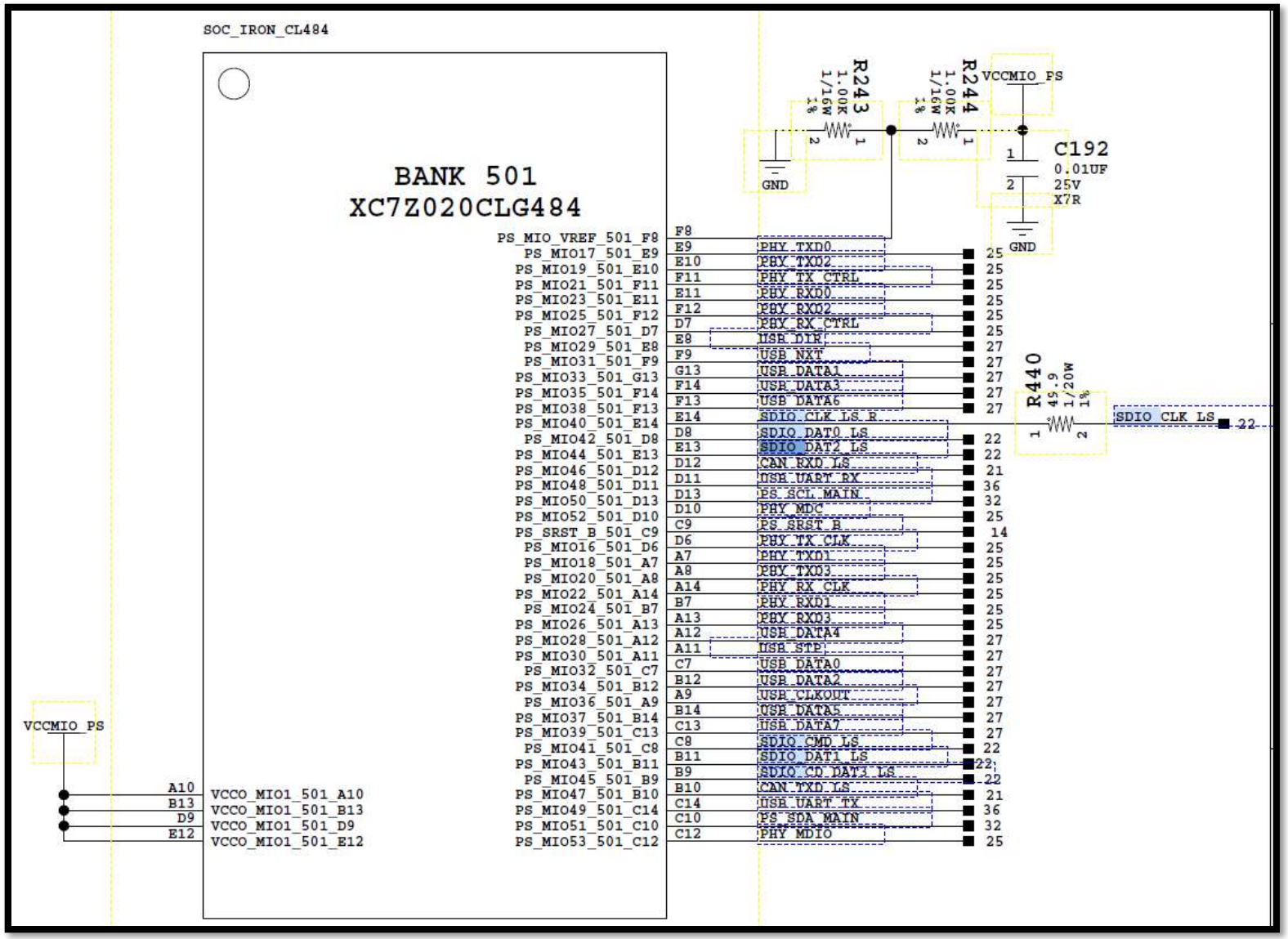
Five pins, BOOT_MODE[4:0], are used to select the boot mode, JTAG chain config, and if the PLLs are bypassed. The sampled values of these pins are written into the slcr.BOOT_MODE [BOOT_MODE] and [PLL_BYPASS] bit fields.

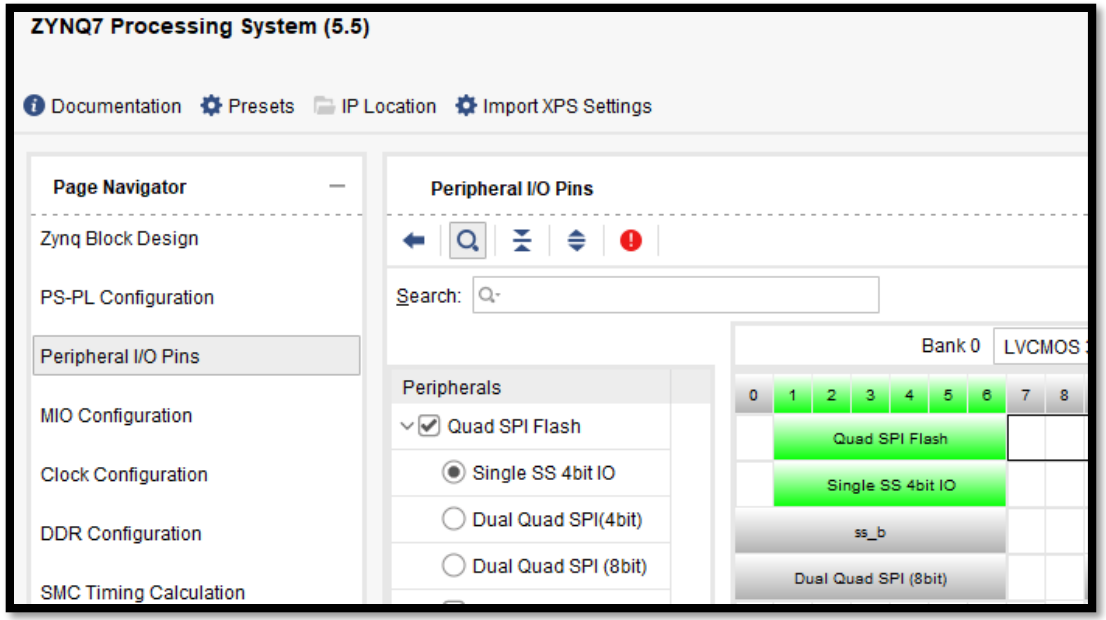
- Boot modes are explained in section 6.3 BootROM Code.
- Boot strap pins are listed in Table 6-4.
- JTAG chains are described in section 6.4.5 PL Control via User-JTAG.
- PLLs are described in section 6.2.3 Clocks and PLLs.

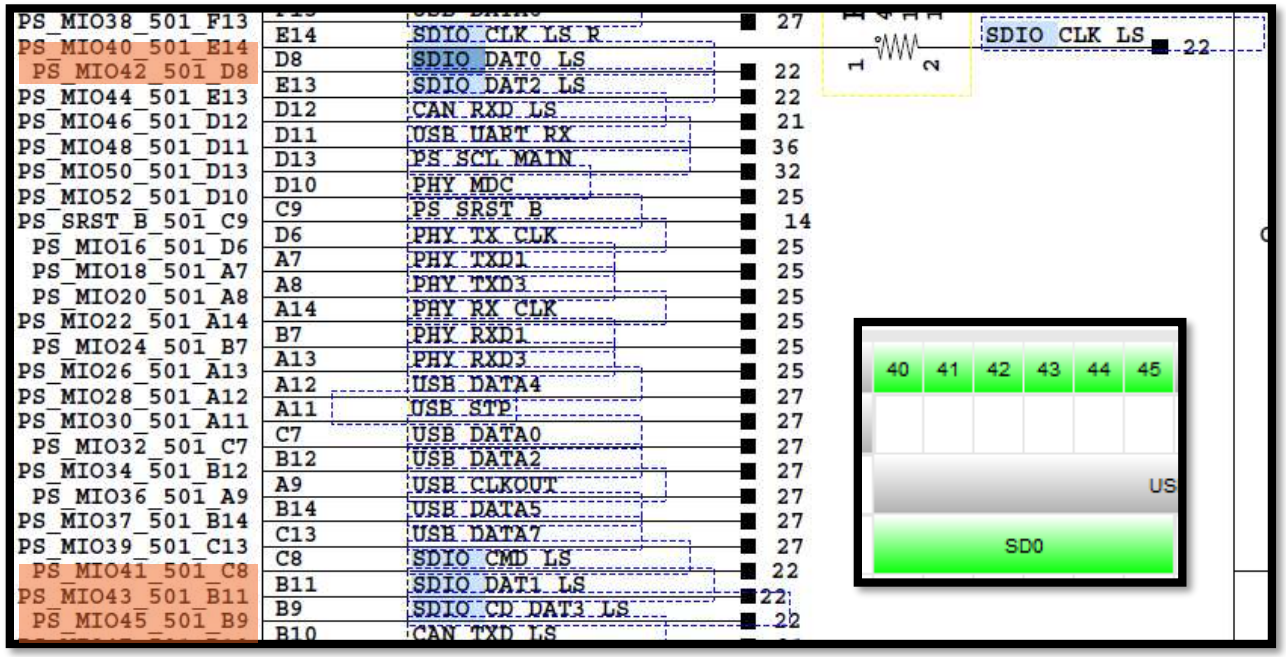
Boot Mode MIO Strapping Pins

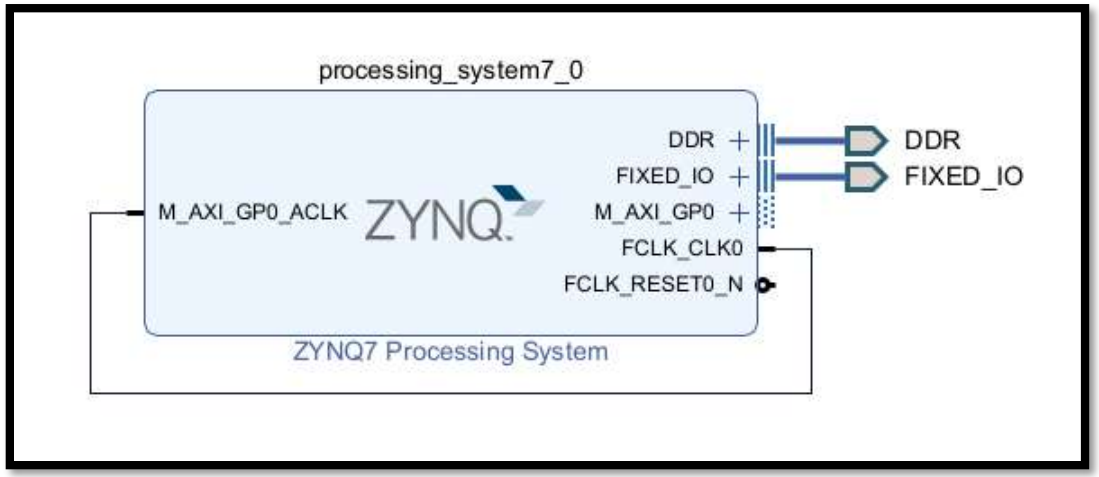
Table 6-4: **Boot Mode MIO Strapping Pins**

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					

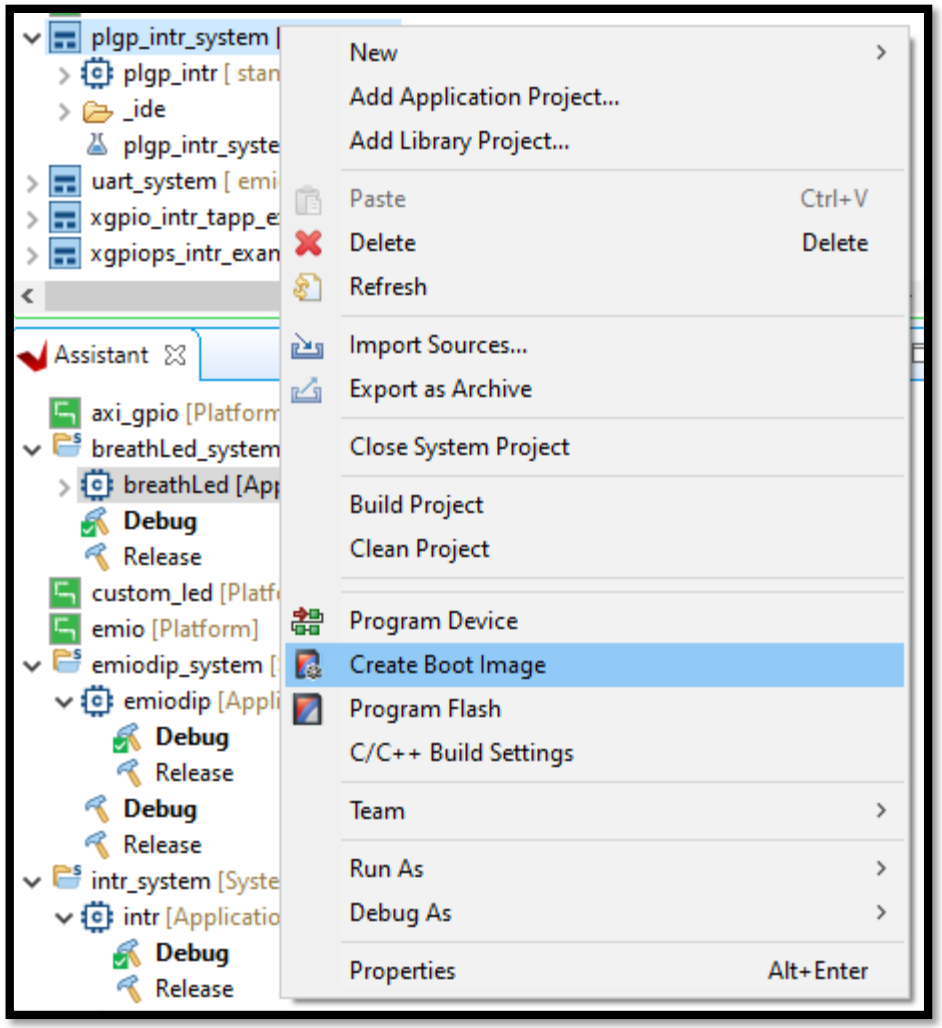


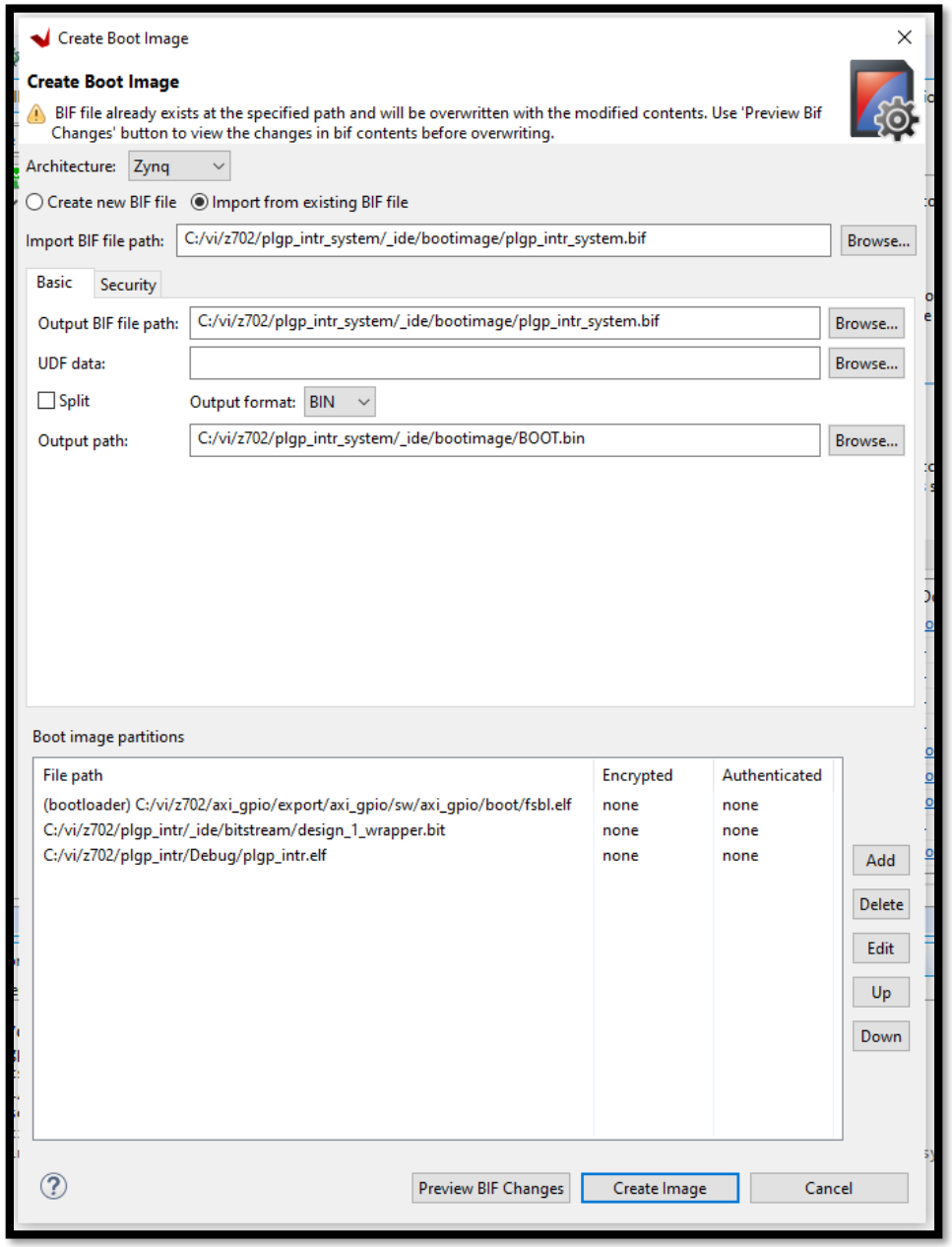




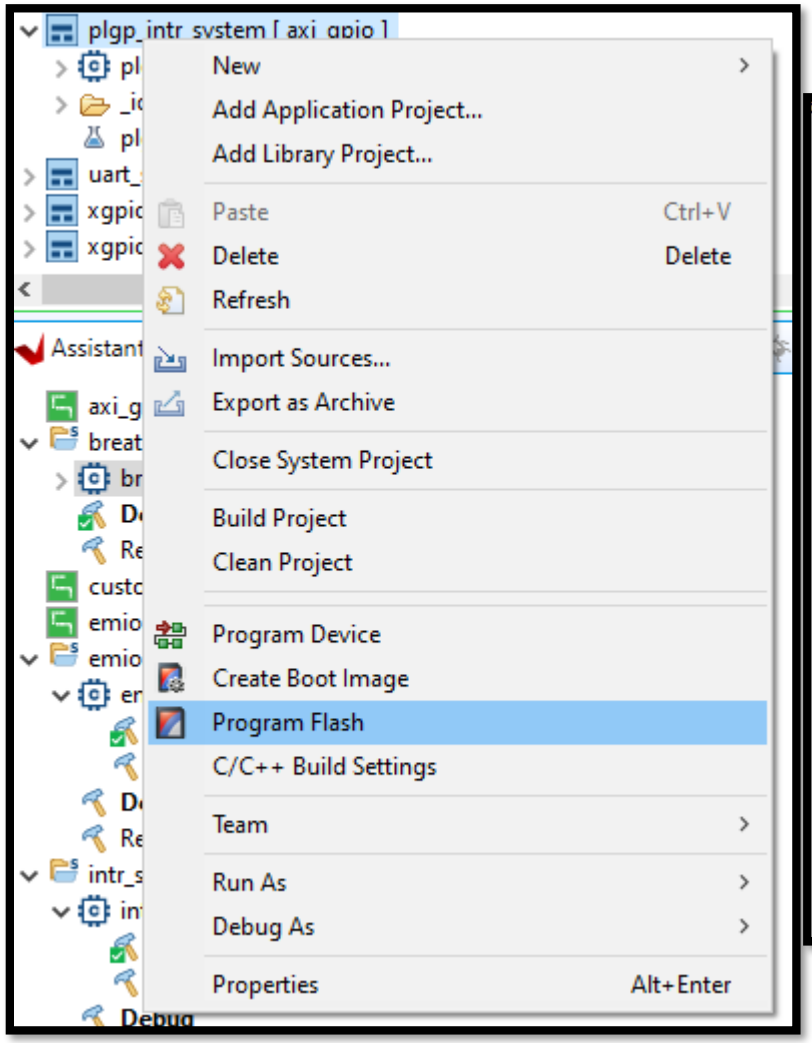


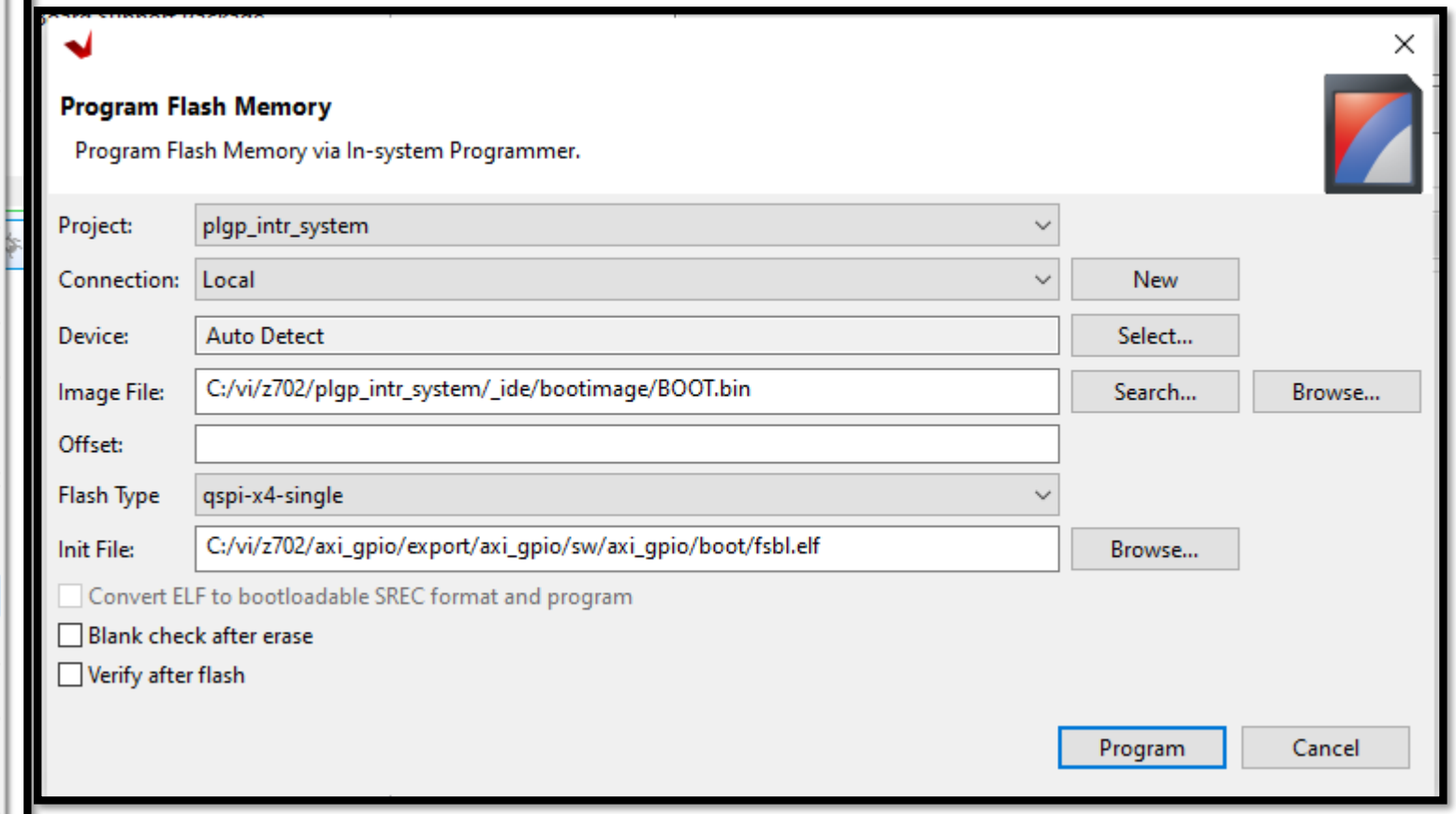
Create Boot Image





Program Flash





Reference

ug585


AMD