Xilinx Standalone Library Documentation

XilFPGA Library v2.1

UG1229 (2017.2) June 7, 2017

XILINX
ALL PROGRAMMABLE™

Table of Contents

Chapter 1

Overview

The XilFPGA library provides an interface to the Linux or bare-metal users for configuring the programmable logic (PL) over PCAP from PS.
The library is designed for Zynq® UltraScale+™ MPSoC to run on top of Xilinx standalone BSPs. It is tested for A53, R5 and MicroBlaze. In the most common use case, we expect users to run this library on PMU MicroBlaze with PMUFW to serve requests from Linux for bitstream programming. In this release, the XilFPGA library supports full, encrypted, authenticated bitstream download. In subsequent releases, the library may support partial bitstream loading.

Xilfpga library Interface modules

Xilfpga library uses the below major components to configure the PL through PS.

Processor Configuration Access Port (PCAP)

The processor configuration access port (PCAP) is used to configure the programmable logic (PL) through the PS.

CSU DMA driver

The CSU DMA driver is used to transfer the actual Bit stream file for the PS to PL after PCAP initialization.

Xilsecure_library

The LibXilSecure library provides APIs to access secure hardware on the Zynq® UltraScale+™ MPSoC devices. This library includes:

- SHA-256 hash function
- AES for symmetric key encryption
- RSA for authentication

Note

- The current version of library supports only Zynq® UltraScale+™ MPSoC devices.
- The XilFPGA library is capable of loading only .bin format files into PL. The library will not support the other file formats.
- Xilsecure_library is required only for the below use cases:
  - Encrypted bit-stream loading.
  - Authenticated bit-stream loading

- For Zynq® UltraScale+™ MPSoC devices, the required OCM memory for authentication bit-stream loading is 68Kb.

Design Summary

Xilfpga library acts as a bridge between the user application and the PL device. It provides the required functionality to the user application for configuring the PL Device with the required bit-stream. The figure below illustrates an implementation where the Xilfpga library needs the CSU DMA driver APIs to transfer the bit-stream from the DDR to the PL region. The Xilfpga library also needs the XilSecure library APIs to support while programming the authenticated and the encrypted bitstream files.

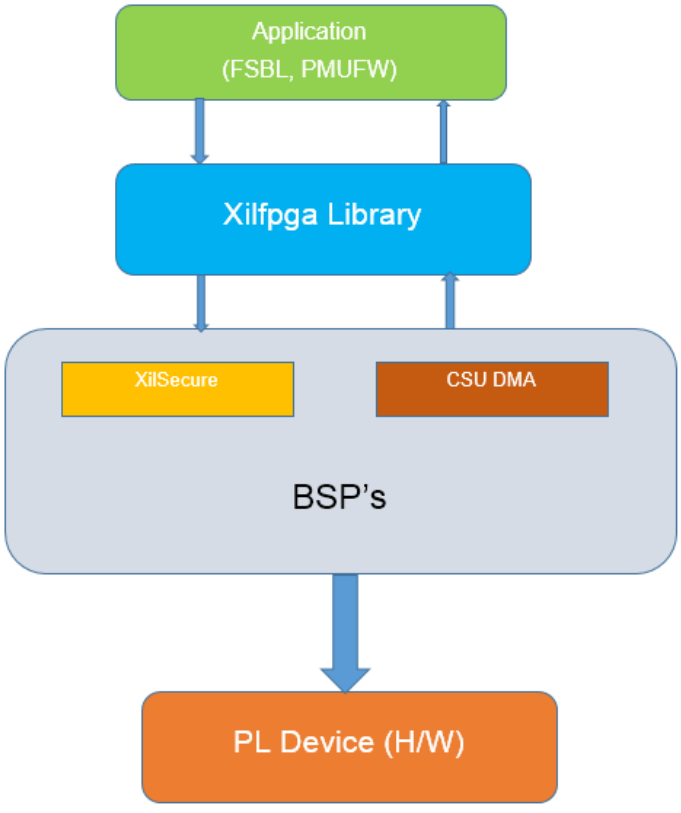

Figure 1.1: XilFPGA Design Summary

Flow Diagram

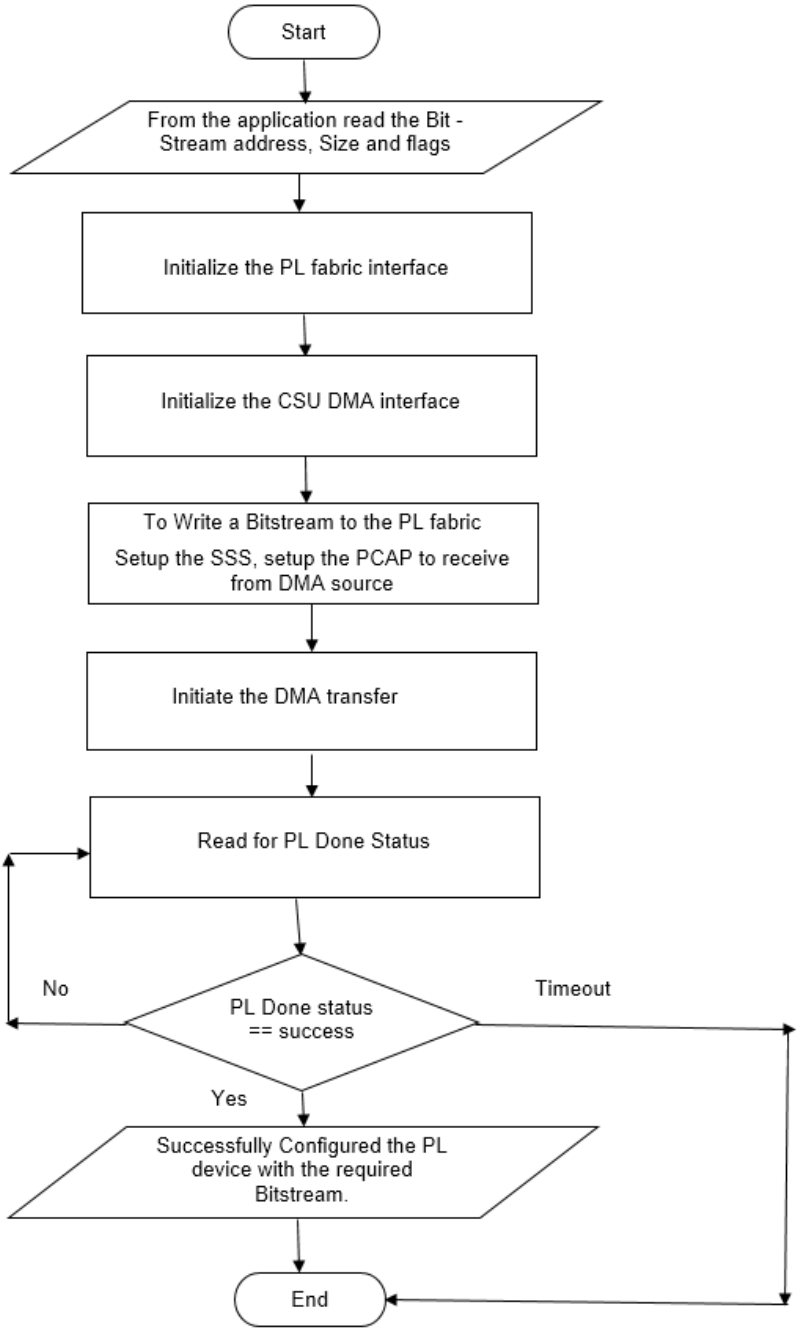

Figure 1.2: XilFPGA Library Workflow

Setting up the Software System

To use XilFPGA in a software application, you must first compile the XilFPGA library as part of software application.

1. Launch Xilinx SDK. Xilinx SDK prompts you to create a workspace.
2. Select **File** > **New** > **Xilinx Board Support Package**. The **New Board Support Package** wizard appears.
3. Specify a project name.
4. Select **Standalone** from the **Board Support Package OS** drop-down list. The **Board Support Package Settings** wizard appears.
5. Select the **xilfpga** library from the list of **Supported Libraries**.
6. Expand the **Overview** tree and select **xilfpga**. The configuration options for xilfpga are listed.
7. Configure the xilfpga by providing the base address of the Bit-stream file (DDR address) and the size (in bytes).
8. Click **OK**. The board support package automatically builds with XilFPGA library included in it.
9. Double-click the **system.mss** file to open it in the **Editor** view.
10. Scroll-down and locate the **Libraries** chapter.
11. Click **Import Examples** adjacent to the XilFPGA 2.1 entry.

Enabling Secure Mode in PMUFirmware

To support encrypted and authenticated bit-stream loading, you must enable secure mode in PMUFW.

1. Launch Xilinx SDK. Xilinx SDK prompts you to create a workspace.
2. Select **File** > **New** > **Application Project**. The **New Application Project** wizard appears.
3. Specify a project name.
4. Select **Standalone** from the **OS Platform** drop-down list.
5. Select a supported hardware platform.
6. Select **psu_pmu_0** from the **Processor** drop-down list.
7. Click Next. The **Templates** page appears.
8. Select **ZynqMP PMU Firmware** from the **Available Templates** list.
9. Click **Finish**. A PMUFW application project is created with the required BSPs.
10. Double-click the **system.mss** file to open it in the **Editor** view.

11. Click the **Modify this BSP's Settings** button. The **Board Support Package Settings** dialog box appears.
12. Select **xilfpga**. Various settings related to the library appears.
13. Select **secure_mode** and modify its value to **true** .
14. Click **OK** to save the configuration.

Chapter 2

XilFPGA APIs

Overview

This chapter provides detailed descriptions of the XilFPGA library APIs.

Functions

- u32 XFpga_PL_BitSream_Load (u32 WrAddrHigh, u32 WrAddrLow, u32 WrSize, u32 flags)
- u32 XFpga_PcapStatus (void)

Function Documentation

u32 XFpga_PL_BitSream_Load (u32 *WrAddrHigh,* u32 *WrAddrLow,* u32 *WrSize,* u32 *flags*)

The API is used to load the user provided bitstream file into zynqmp PL region.
This function does the following jobs:

- Power-up the PL fabric.
- Performs PL-PS Isolation.
- Initialize PCAP Interface
- Write a bitstream into the PL
- Wait for the PL Done Status.
- Restore PS-PL Isolation (Power-up PL fabric).
- Performs the PS-PL reset.

Note

This function contains the polling implementation to provide the PL reset wait time due to this polling implementation the function call is blocked till the time out value expires or gets the appropriate status value from the PL Done Status register.

Parameters

WrAddrHigh	Higher 32-bit Linear memory space from where CSUDMA will read the data to be written to PCAP interface
WrAddrLow	Lower 32-bit Linear memory space from where CSUDMA will read the data to be written to PCAP interface
WrSize	Number of 32bit words that the DMA should write to the PCAP interface
flags	Flags are used to specify the type of bitstream file. ◦ BIT(0) - Bit-stream type ■ 0 - Full Bit-stream ■ 1 - Partial Bit-stream ◦ BIT(2) - Authentication ■ 1 - Enable ■ 0 - Disable ◦ BIT(3) - Encryption ■ 1 - Enable ■ 0 - Disable

Returns

- Error status based on implemented functionality (SUCCESS by default).

u32 XFpga_PcapStatus (void)

This function provides the STATUS of PCAP interface.

Parameters

None	

Returns

Status of the PCAP interface.

Appendix A

Additional Resources and Legal Notices

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see Xilinx Support .

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

Please Read: Important Legal Notices

Automotive Applications Disclaimer

AUTOMOTIVE PRODUCTS (IDENTIFIED AS ”XA” IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE (”SAFETY APPLICATION”) UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD (”SAFETY DESIGN”). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.