User's Guide

AFE79xx SPI Bringup Guide With Xilinx FPGAs

ABSTRACT

This tutorial guides through the process of using Xilinx Vivado and Vitis development environments along with Texas Instruments supplied custom IP to bring up Serial Peripheral Interface (SPI) and non-timing critical General-Purpose Outputs (GPOs) for Texas Instruments AFE79xx EVM along with the companion LMK series clocking chip, thereby enabling an easier integration of the AFE79xx device into a system design. This guide will demonstrate how to use a Xilinx ZCU102 setup as an example.

Table of Contents

Trademarks

All trademarks are the property of their respective owners.

Copyright © 2022 Texas Instruments Incorporated

1 Introduction

This user guide is a walk-through of complete hardware and software flow to bring up SPI and GPO with TI supplied AFE SPI IP. The hardware in this case refers to the AFE SPI IP supplied by TI which uses a Microblaze processor along with AXI SPI, AXI GPIO, and other required peripherals. Shielding end-customer from the nuances of this setup is one of the core objectives in packaging these in a single custom IP container.

The specific step-wise objectives are as follows:

- Instantiate TI supplied AFE SPI IP in a Vivado project
- Map the supplied IP's required signals to FPGA IOs
- Import hardware design and building a new Vitis application project for software development
- Compile, link, and download C program to processor along with bit file for FPGA

2 Prerequisites

For effective use of this documentation, ensure to have the following prerequisites:

- Xilinx Vitis IDE v2020.1.0 (or higher)
- Xilinx Vivado v2020.1.0 (or higher)
- Xilinx FPGA board along with TI AFE EVM
- FPGA bit file download/debug programmer
- USB-UART cable for debug terminal
- TI supplied AFE SPI IP
- TI supplied C-APIs

Table 2-1. Prerequisites

TI AFE	AFE79xx
Sample Configuration	2T-2R-1FB
Lanes	2 RX lanes (1RX, 1FB) and 2 TX lanes at 5 Gbps
AFE EVM	AFE79xx EVM
FPGA Board	Xilinx ZCU102 EVM

Copyright © 2022 Texas Instruments Incorporated

3 Typical Bare-Metal Design Flow

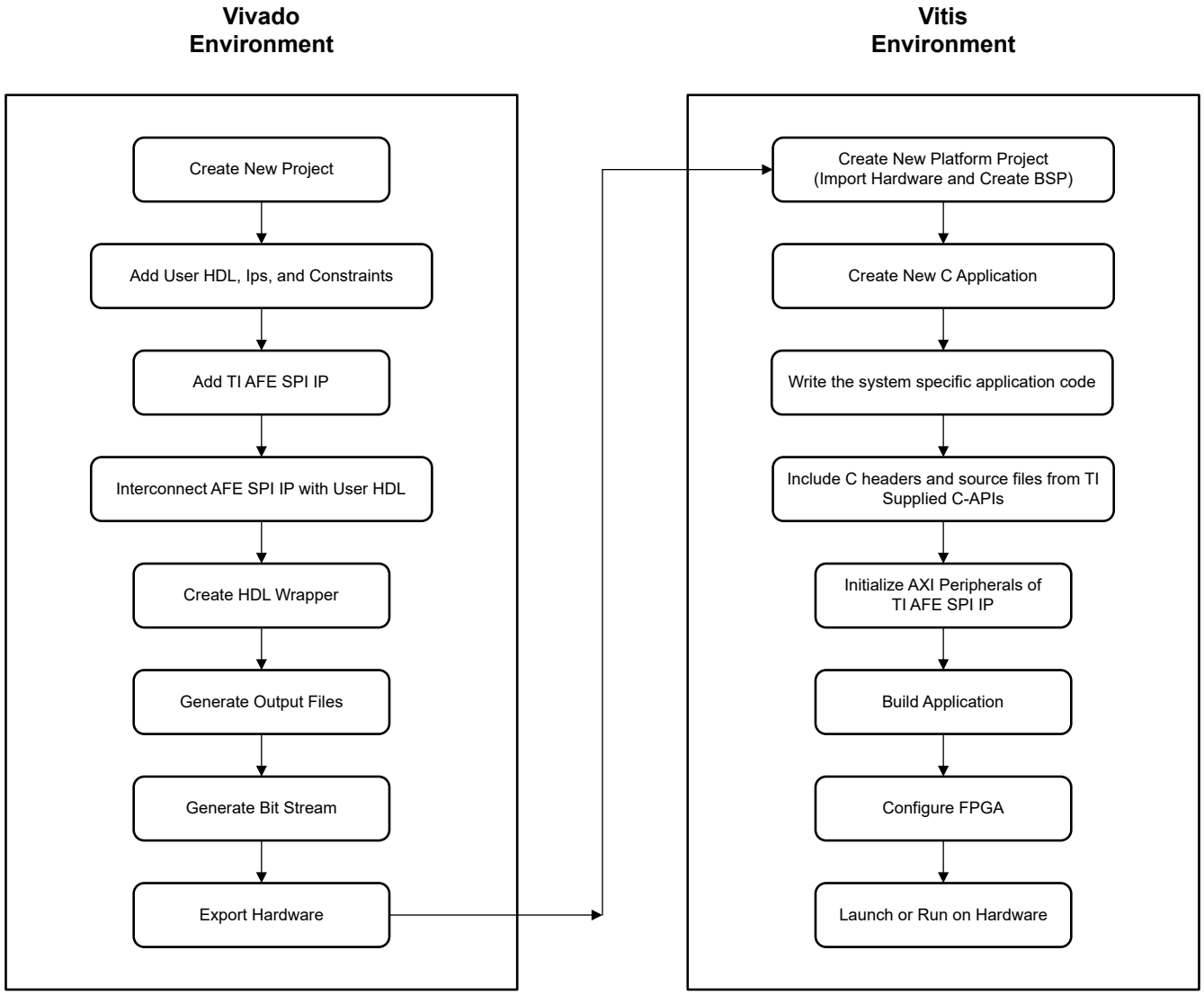


Figure 3-1. Bare-Metal Design Flow

Copyright © 2022 Texas Instruments Incorporated

4 Background

This example uses a soft core Microblaze because the Microblaze can be instantiated in most of the Xilinx FPGA families. The SPI, UART, and GPIO AXI blocks run on relatively lower frequency AXI clocks. As seen in Figure 4-1, the AXI peripherals are controlled by a Microblaze block through smart interconnects.

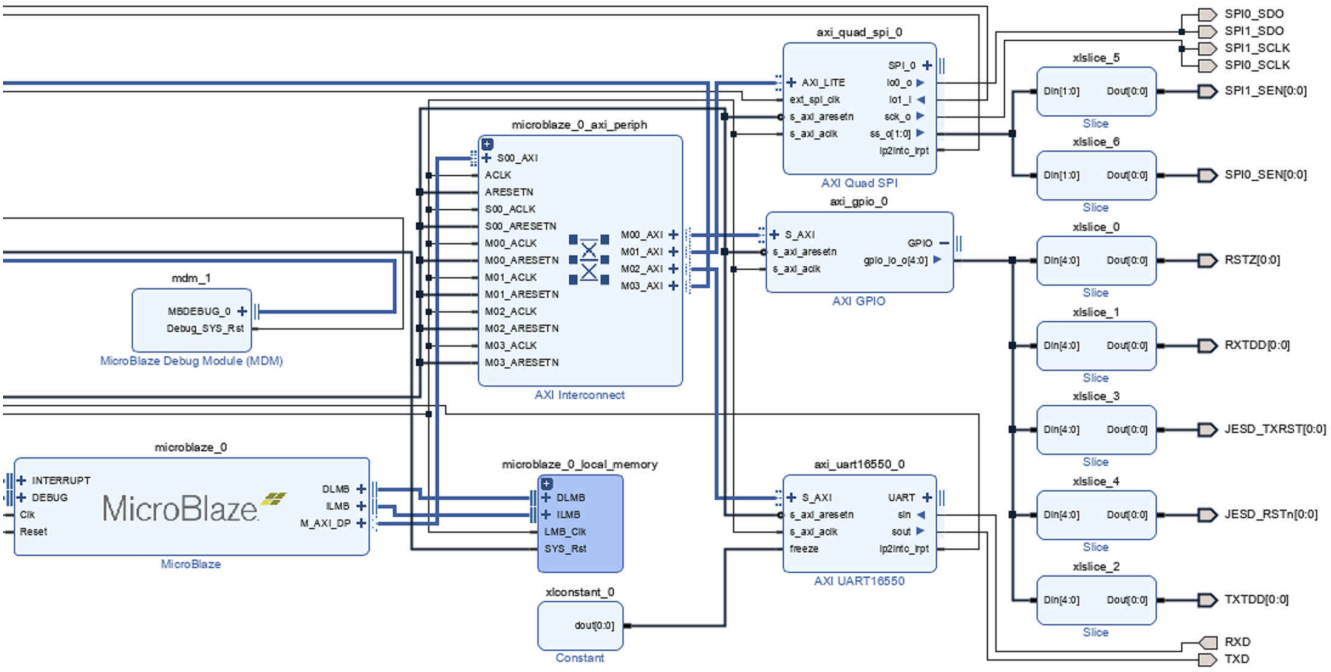


Figure 4-1. Internal Block Diagram of TI AFE SPI IP

The interconnection of various submodules within this IP block diagram is similar to typical Microblaze implementations. The HP port of the Microblaze drives AXI peripherals. The clocking of the entire IP is expected from a 100-MHz differential clock source. This example uses a 100-MHz differential clock source because this clock is typically available as ‘user clock’ in most FPGA EVMs. All other clock frequencies are derived internally through a clocking wizard.

Copyright © 2022 Texas Instruments Incorporated

5 AFE SPI IP Container Pinout

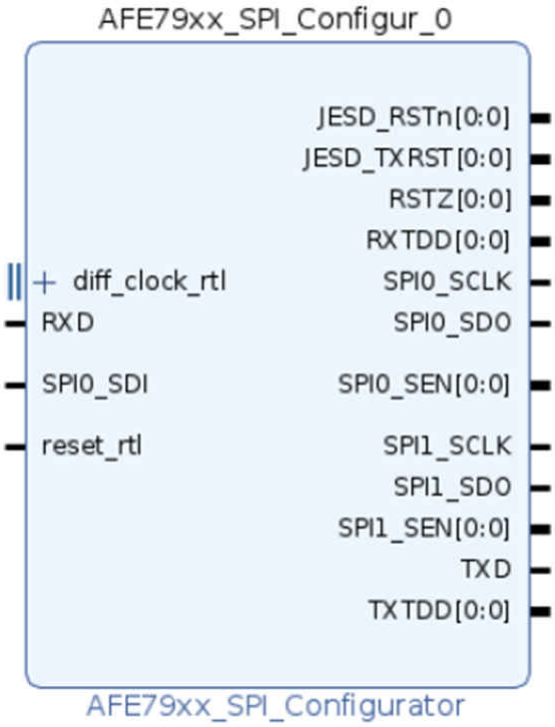


Table 5-1. Pinout Signals and Connections

SIGNALS	DIRECTION	EXTERNAL CONNECTIONS
SPI0_SCLK, SPI0_SDO*, SPI0_SEN	Output	AFE SPI lines (_SDO* to SDI of AFE)
SPI0_SDI	Input	AFE SDO
SPI1_SCLK, SPI1_SDO, SPI1_SEN	Output	LMK SPI lines
RSTZ	Output	RESETn of AFE
JESD RSTn JESD TXRST	Output	JESED IP Cores RSTn and TX Rst
RXD	Input	UART Terminal TX for debug
TXD	Output	UART Terminal RX for debug
diff_clock_rtl	Input	100-Mhz differential clocking
Reset_rtl	Input	Reset (Active High) typically connected to FPGA board reset

Copyright © 2022 Texas Instruments Incorporated

6 TI AFE SPI IP Container

By default, custom and user-added IPs are not visible in IP catalog of Vivado. The locations for these IPs must be configured manually. To configure the IP locations manually, follow these steps:

1. Open the *Project Manager* menu and click *Settings* (see Figure 6-1).

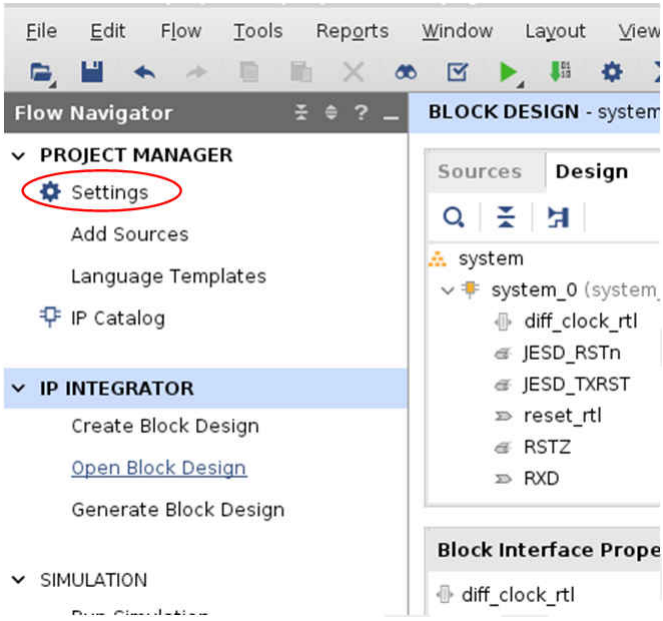


Figure 6-1. Project Manager Settings

2. Click the + icon to add the location of TI provided IP (see Figure 6-2).

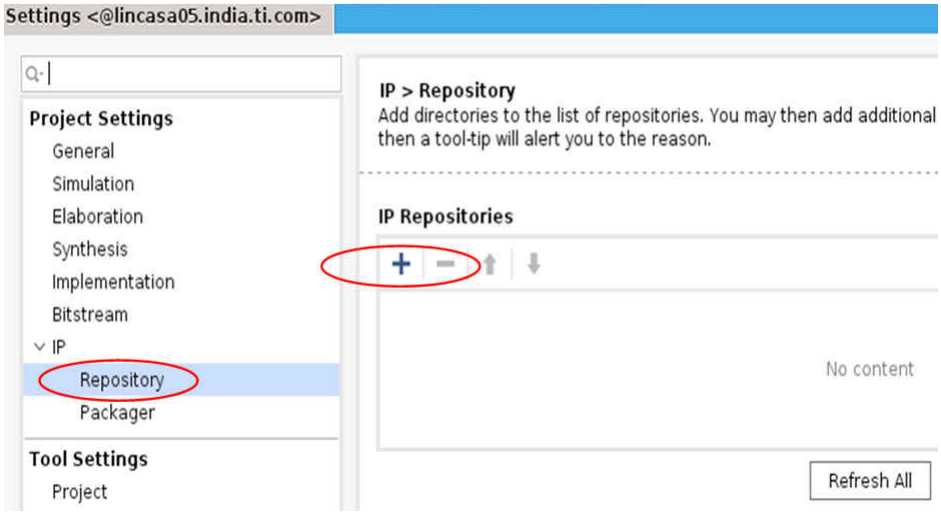


Figure 6-2. Adding IP Repository

Copyright © 2022 Texas Instruments Incorporated

7 Create Block Designs With TI AFE SPI IP

The IP is expected to be used only within a block design. Users can create a new block design or instantiate the IP within an existing one. To create a new block design with the IP, follow these steps:

1. Open the *IP INTEGRATOR* menu and click *Create Block Design* (see Figure 7-1).

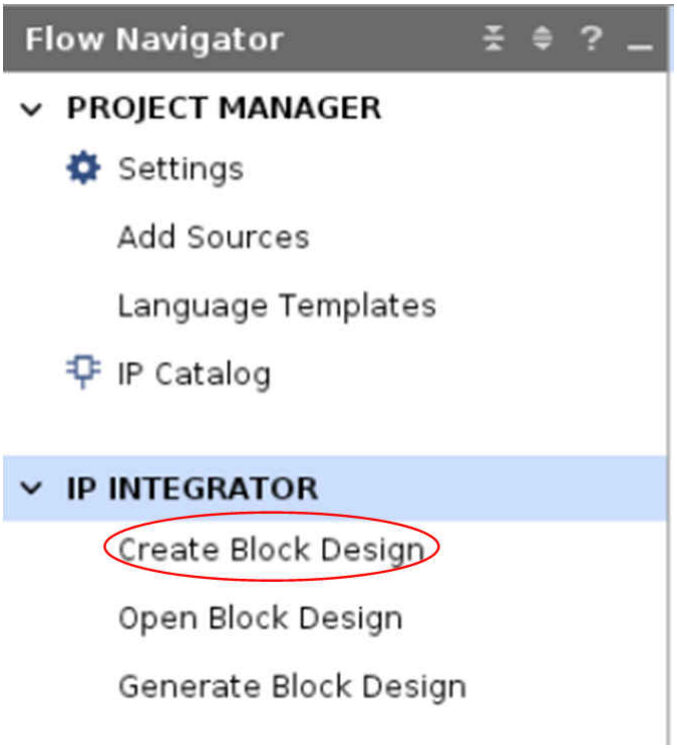


Figure 7-1. Creating Block Design

2. Enter the desired design name (see Figure 7-2) and keep the other two options as defaults. Click OK.

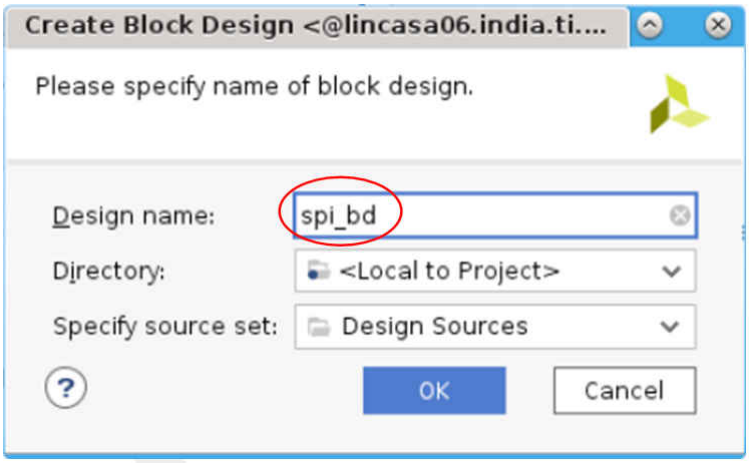


Figure 7-2. Naming of Block Design

3. Click the *IP Catalog* tab and notice the *User Repository* header (see Figure 7-3). The header appears only if the inclusion of the IP repository is done correctly, as explained in previous steps.

Copyright © 2022 Texas Instruments Incorporated

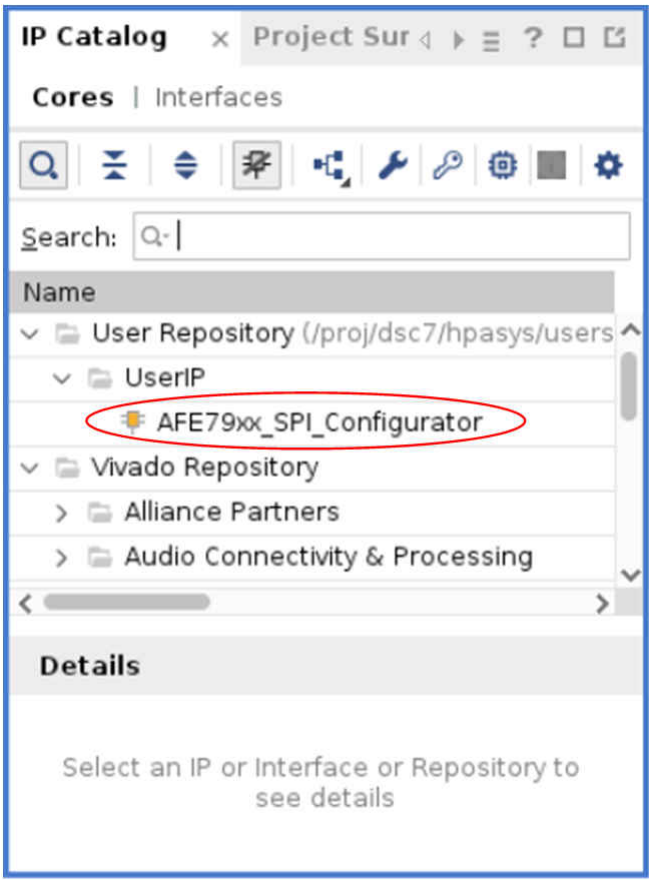


Figure 7-3. User Repository

4. Double-click *AFE79xx SPI Configurator* to launch the *Add IP* window.
5. Click *Add IP to Block Design* (see Figure 7-4).

Figure 7-4. Adding IP to Block Design

6. The AFE SPI IP shows up in the block design (see Figure 7-5) if the previous steps were followed correctly.

Copyright © 2022 Texas Instruments Incorporated

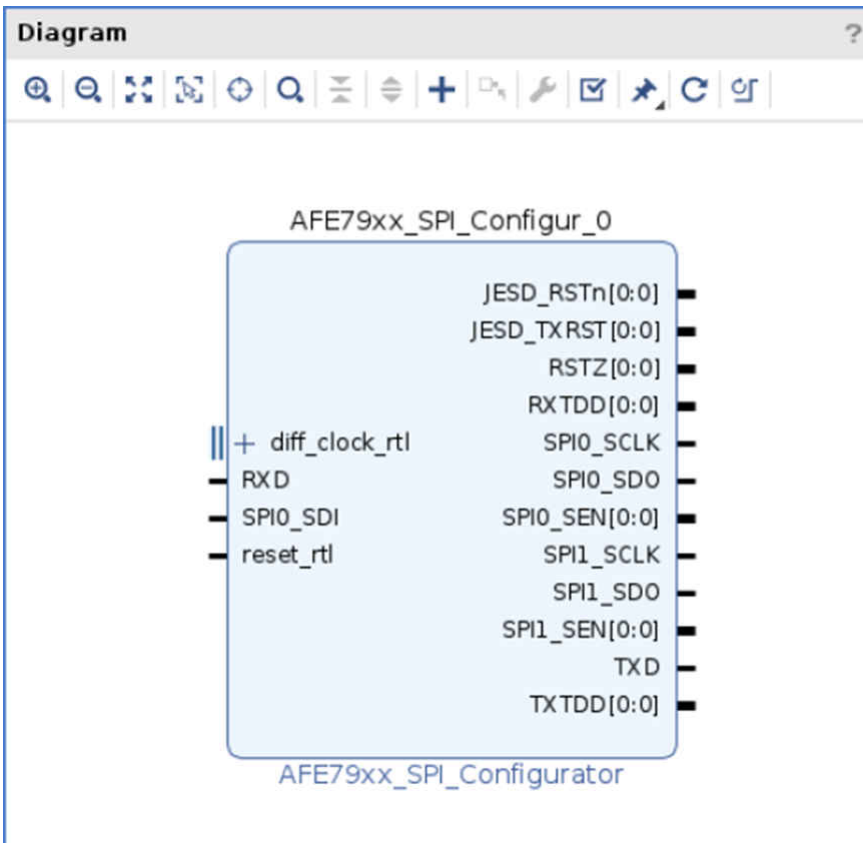


Figure 7-5. Final Diagram

The required IOs from the IP must be brought to the top-level file in the FPGA hdl design hierarchy. To do this, first make the required IOs as external. To make the required IOs as external, right-click the individual IOs of interest and select *Make External* (see Figure 7-6).

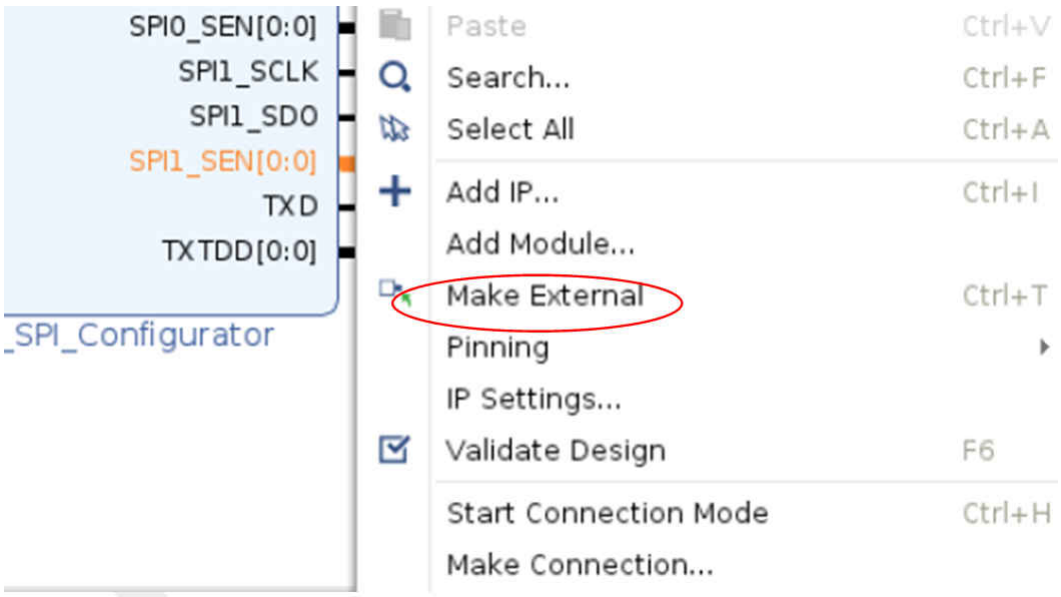


Figure 7-6. Making IOs External

In the above implementation example, the bare minimum required IOs are brought out as external.

The block design must then be validated for any errors and output products to generate. To validate the block diagram, right-click the block design under the *Design Sources* header.

Copyright © 2022 Texas Instruments Incorporated

Similarly, right-click the *View Instantiation Template* to use as a reference for wiring this block design on the top level hdl (see Figure 7-7).

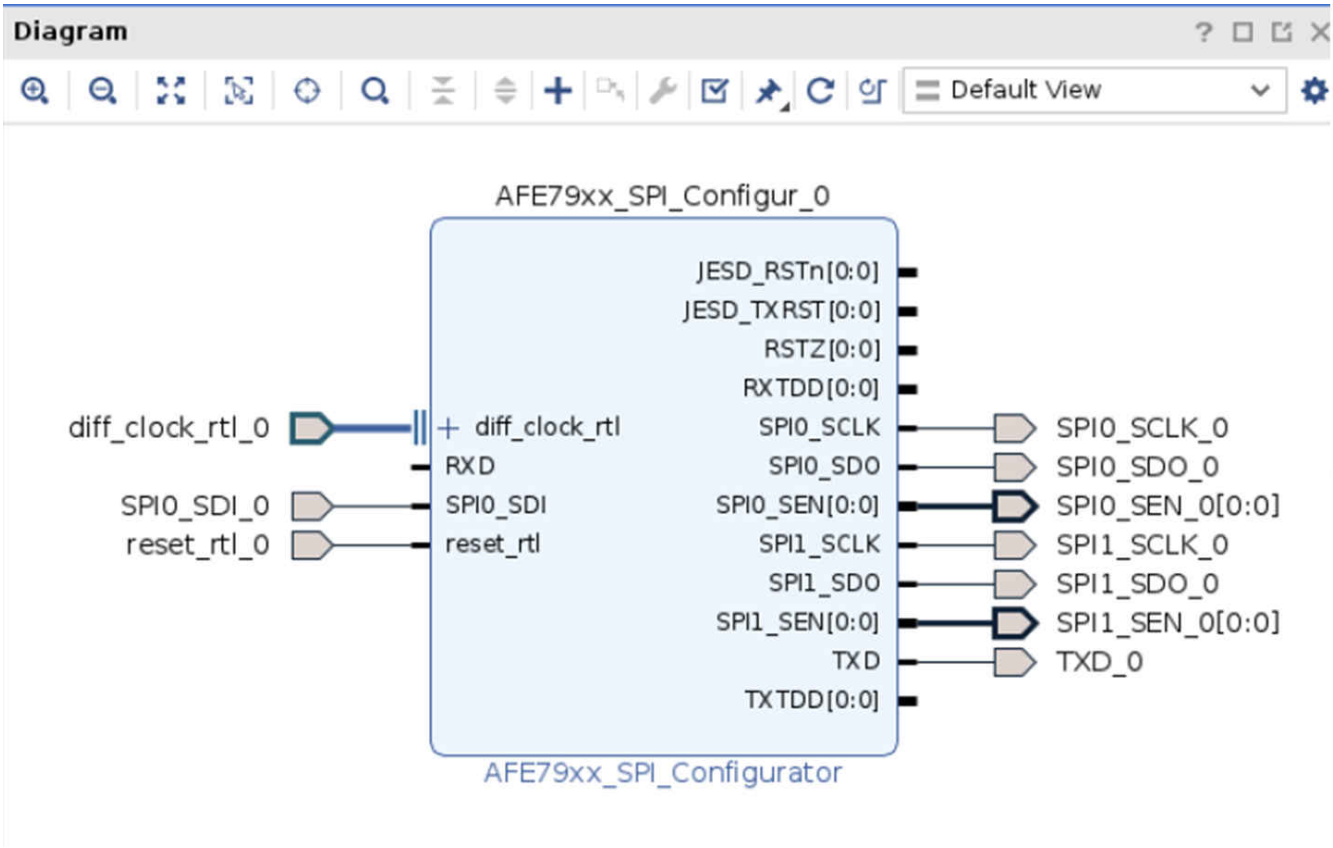


Figure 7-7. TI SPI Configurator Pinout

The IOs from the top level hdl which are connected to this IP must be declared in a *Constraints* file for them to be mapped to the hardware design, specific FPGA pins, and the correct IO levels.

The above steps complete the hardware design using TI AFE SPI IP.

Copyright © 2022 Texas Instruments Incorporated

8 Create New Platforms in Vitis

To create a new platform in Vitis, follow these steps below:

1. Open the *File* menu, go to *New*, and then click *Platform Project* (see Figure 8-1).

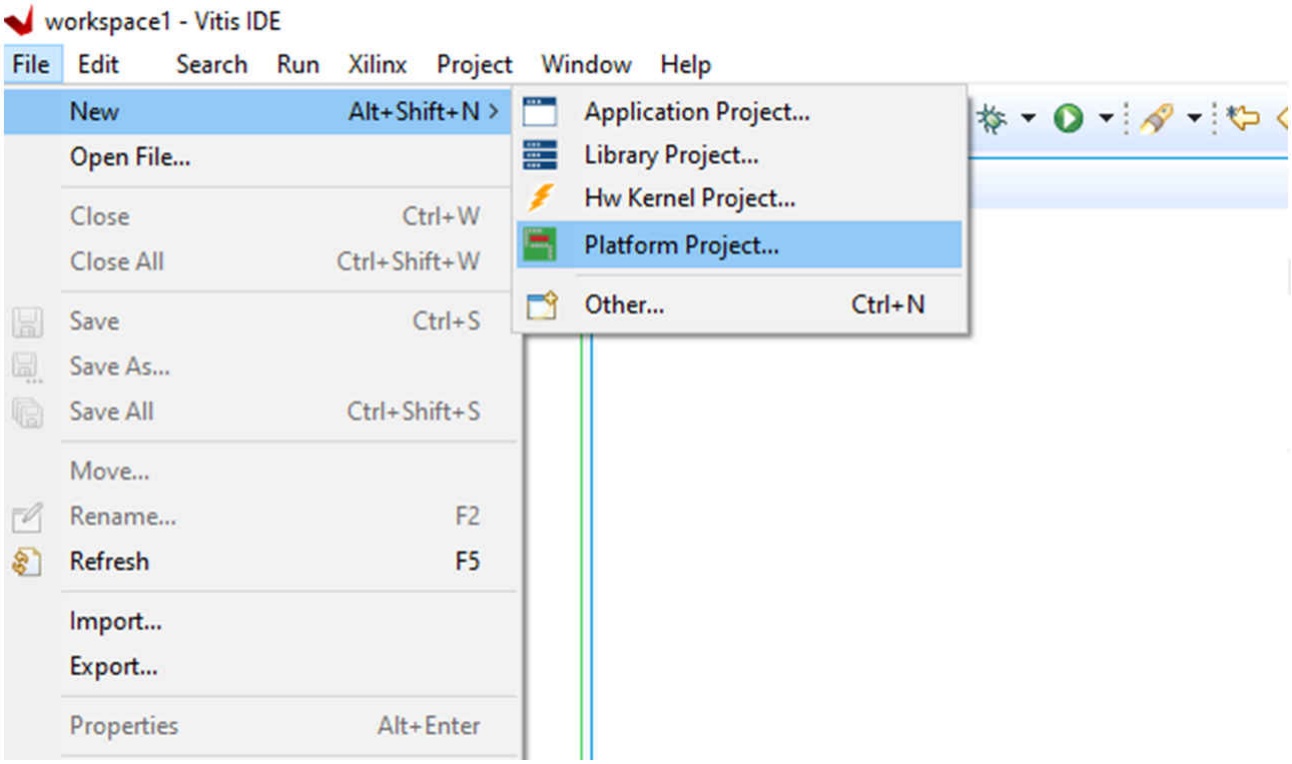


Figure 8-1. New Platform Project

2. Enter the desired platform name. The name *ZCU102ps* was used as an example (see Figure 8-2).

Copyright © 2022 Texas Instruments Incorporated

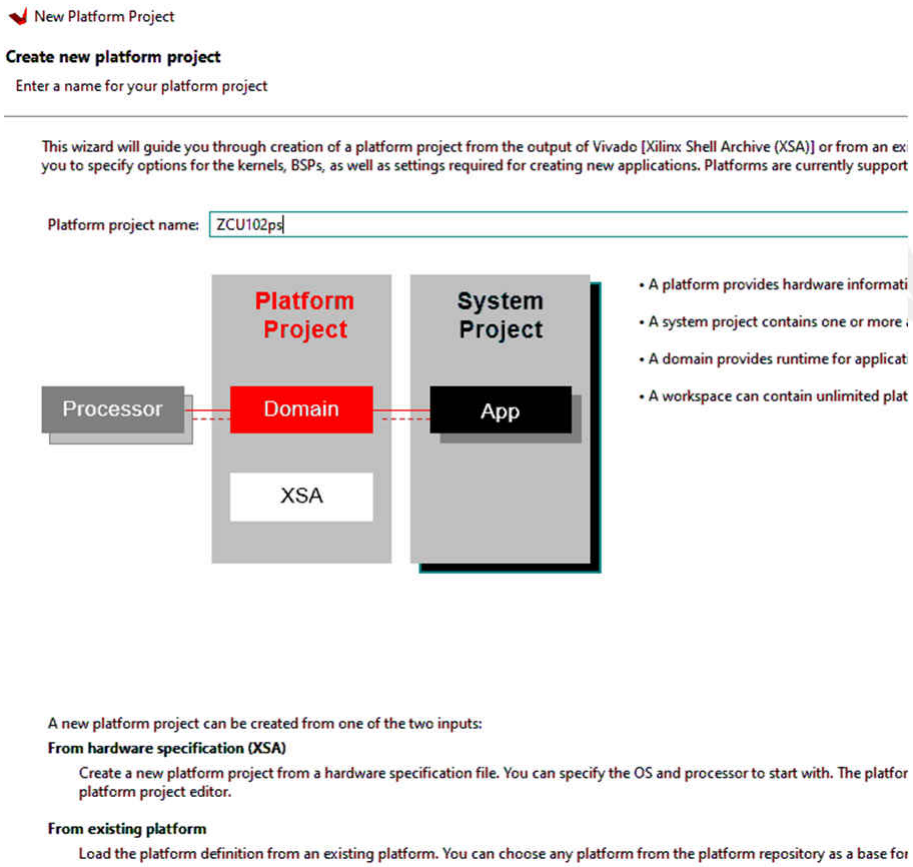


Figure 8-2. Naming of Platform Project

3. After the new platform is named, a menu appears (see Figure 8-3). Select the *XSA* (Xilinx Support Archive) file.

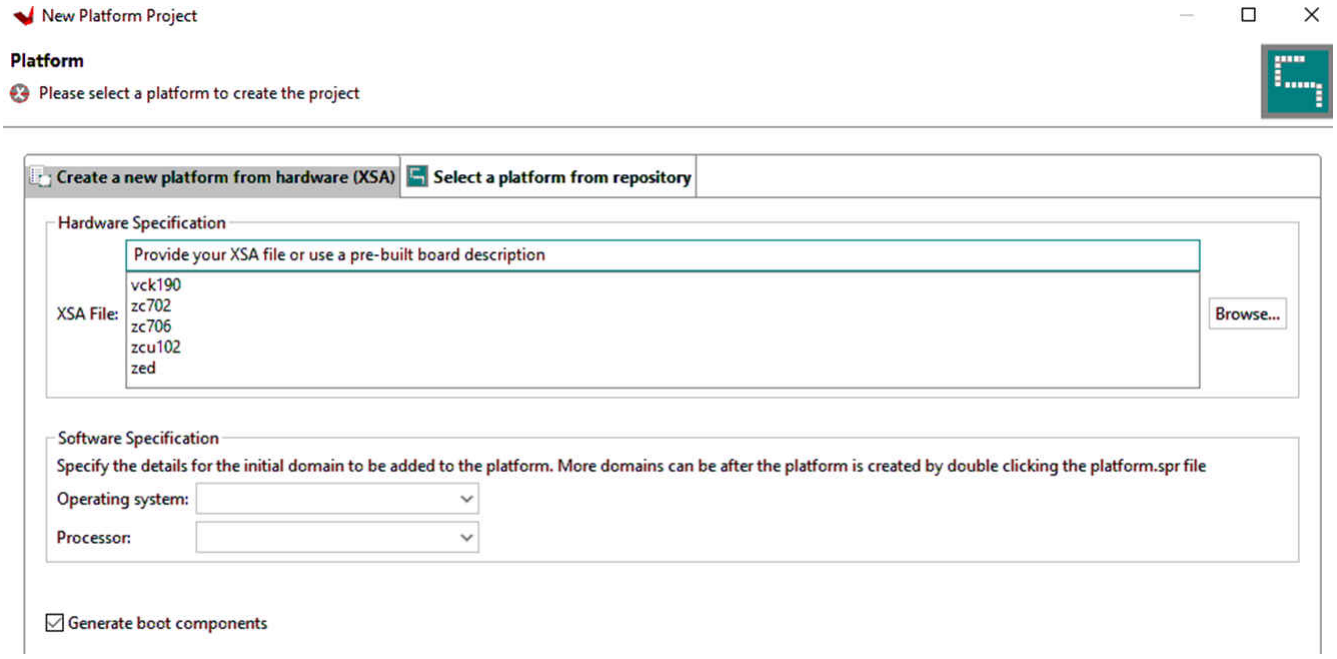


Figure 8-3. Hardware Specification

4. Browse and select the *.XSA* file from the FPGA folder shared along with this document (see Figure 8-4).

Copyright © 2022 Texas Instruments Incorporated

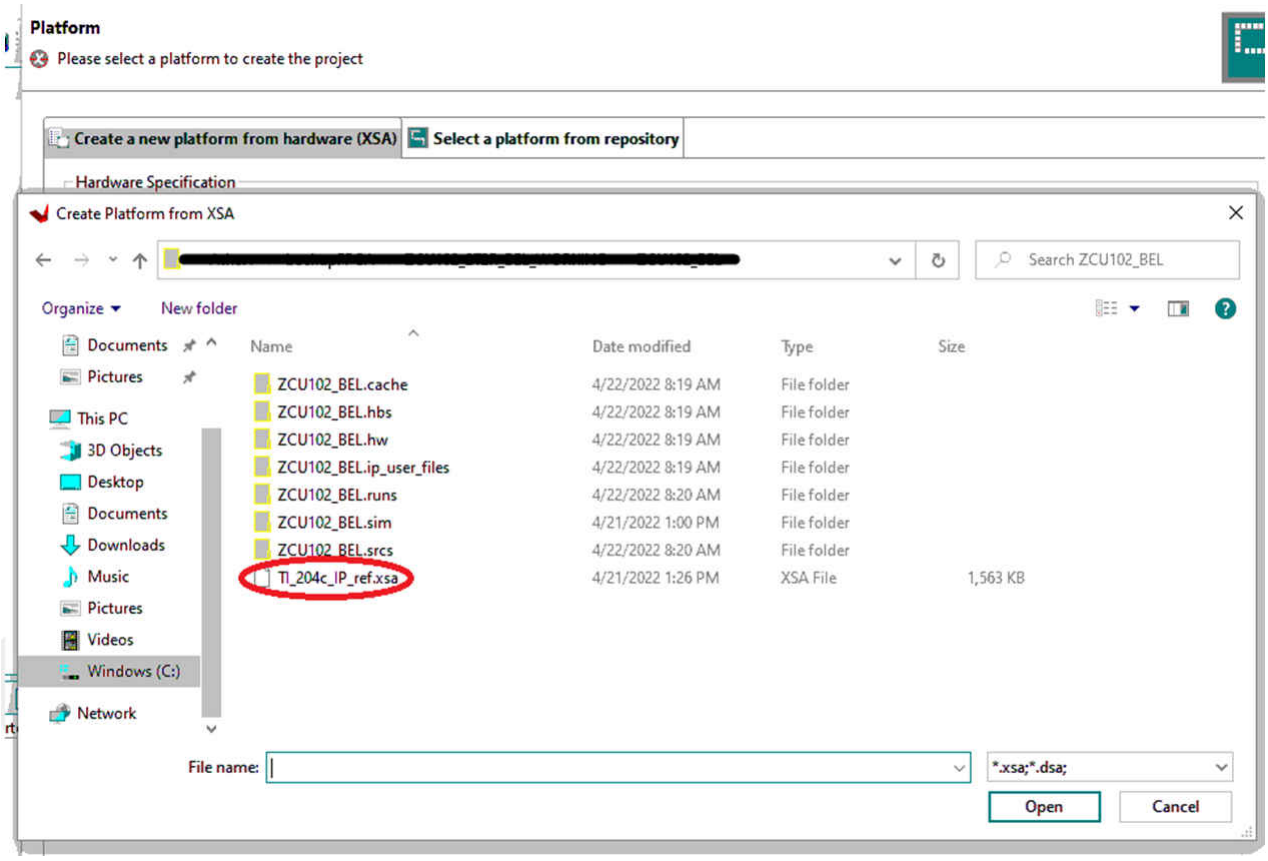


Figure 8-4. Selecting the Platform

5. Right-click the new platform project to open the drop-down menu. Click *Build Project* to start the build (see Figure 8-5). This can take some time to complete the build.

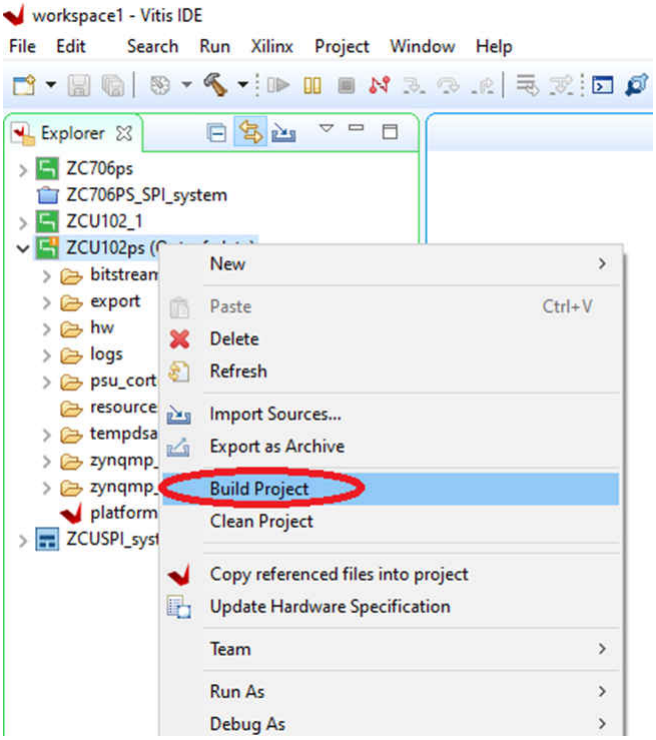


Figure 8-5. Building the New Project

Copyright © 2022 Texas Instruments Incorporated

9 Create New Application Projects in Vitis

After the build is complete, create a new application project in Vitis. To create a new project, follow these steps:

1. Right-click the *Platform* project, go to *New*, then click *Application Project* (see Figure 9-1).

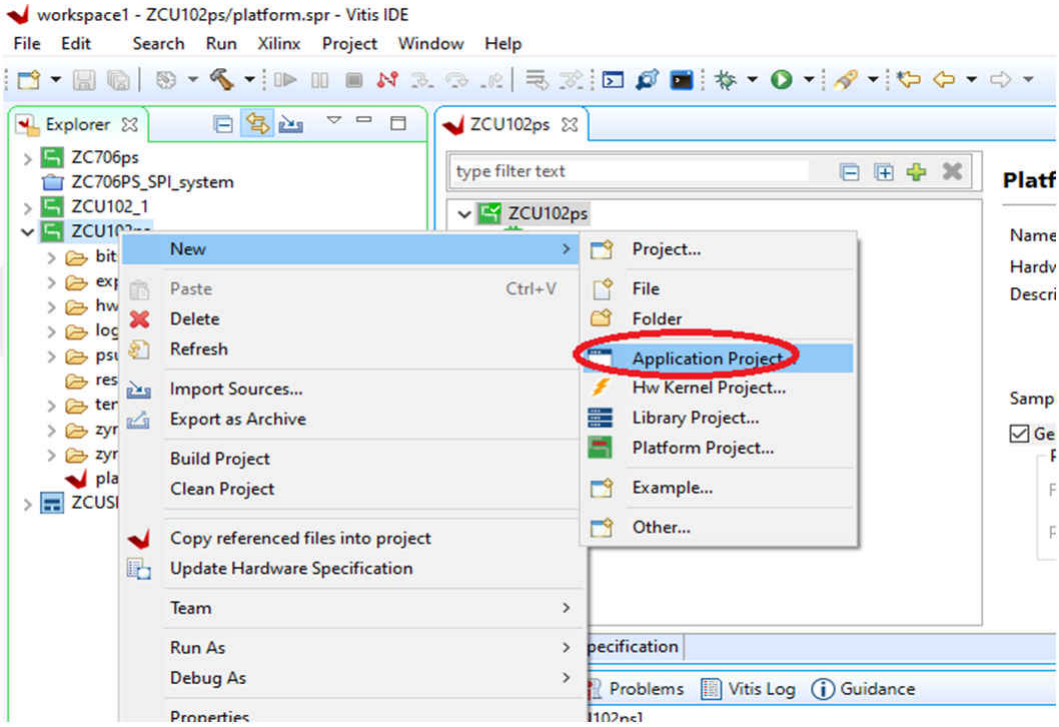


Figure 9-1. Creating New Application Project

2. When the *New Application Project* window appears (see Figure 9-2), click *Next*.

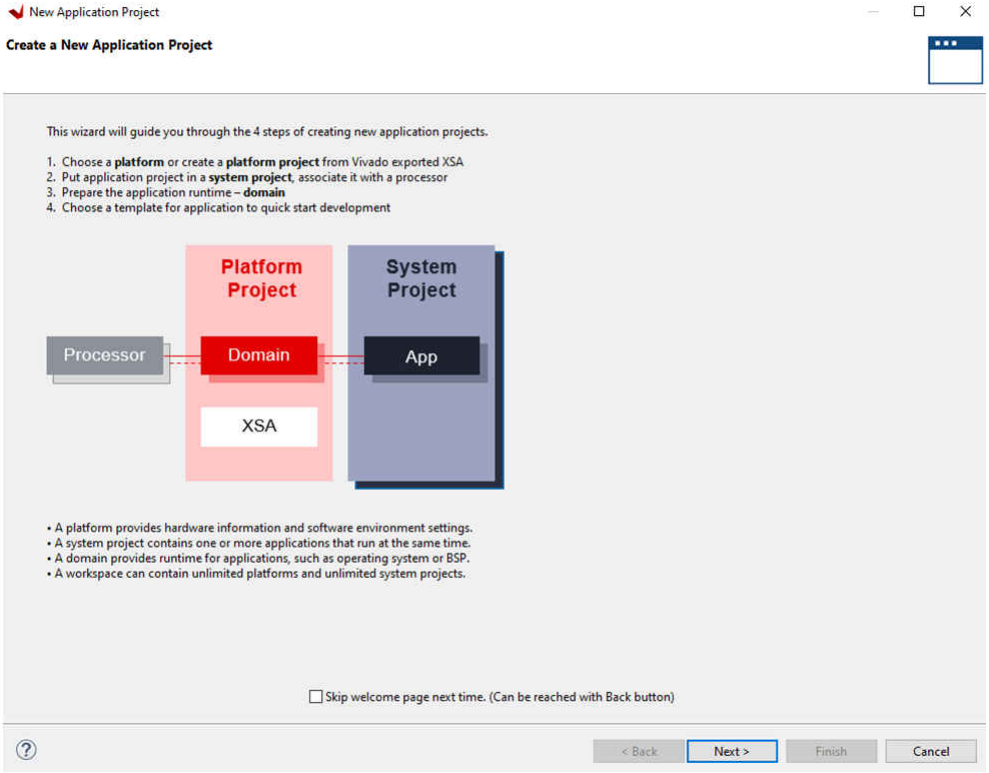


Figure 9-2. New Application Project

3. Select the newly created platform *ZCU102ps* and click *Next* (see Figure 9-3).

Copyright © 2022 Texas Instruments Incorporated

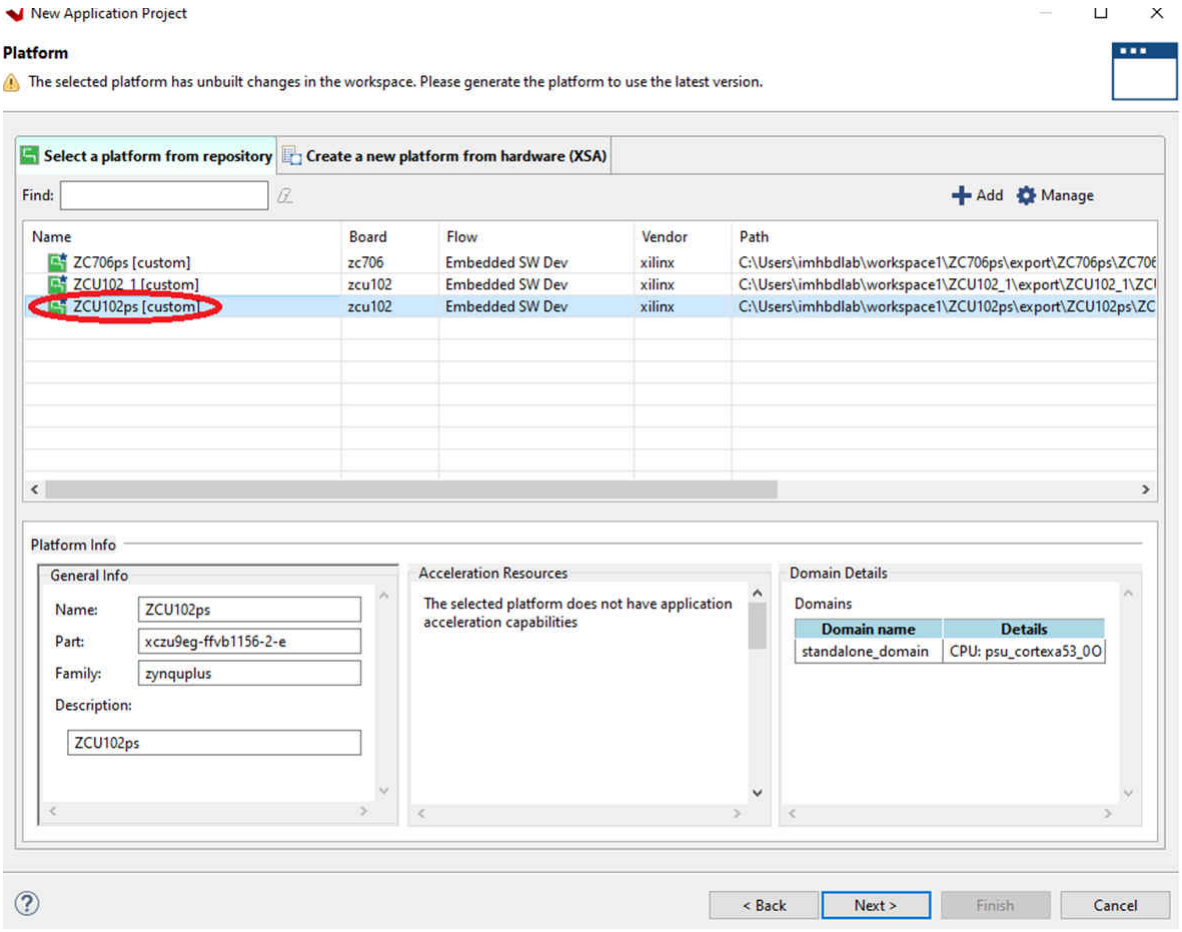


Figure 9-3. Selecting the Application Project

4. Type a new application name. *ZCU102ps_SPI* was used as an example.

Copyright © 2022 Texas Instruments Incorporated

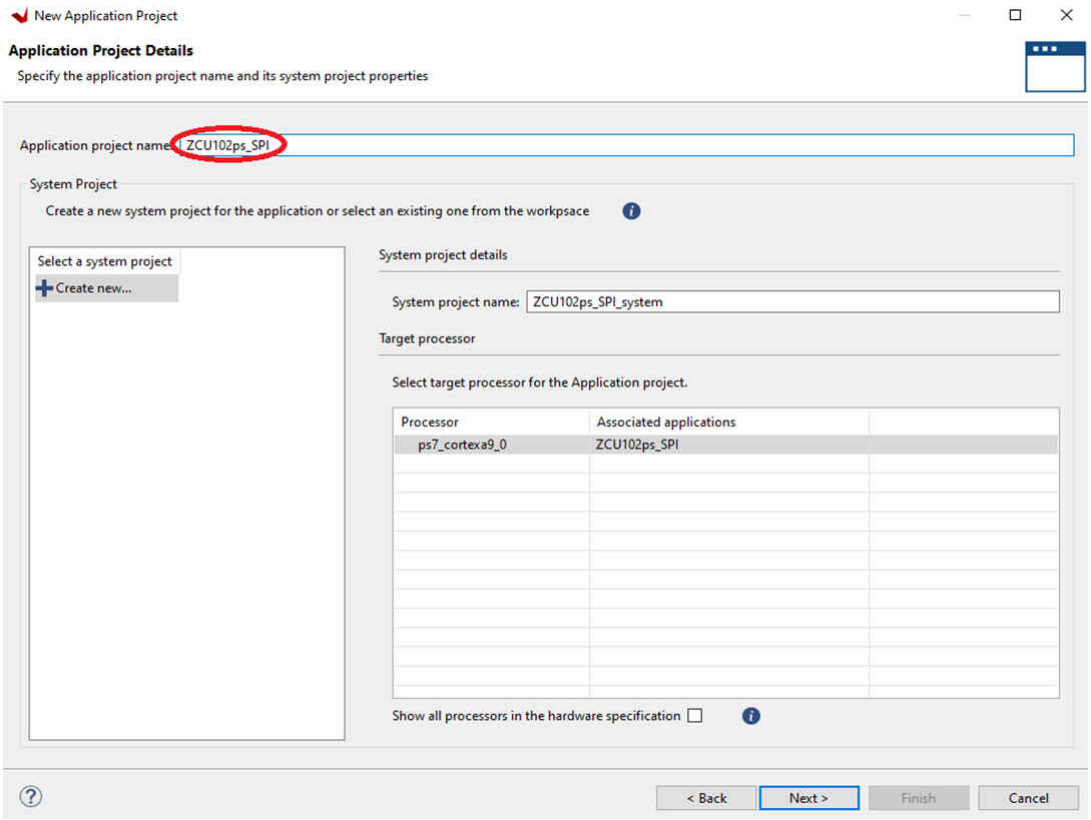


Figure 9-4. New Application Project Name

5. Select *standalone on microblaze_0* and click *Next* (see Figure 9-5).

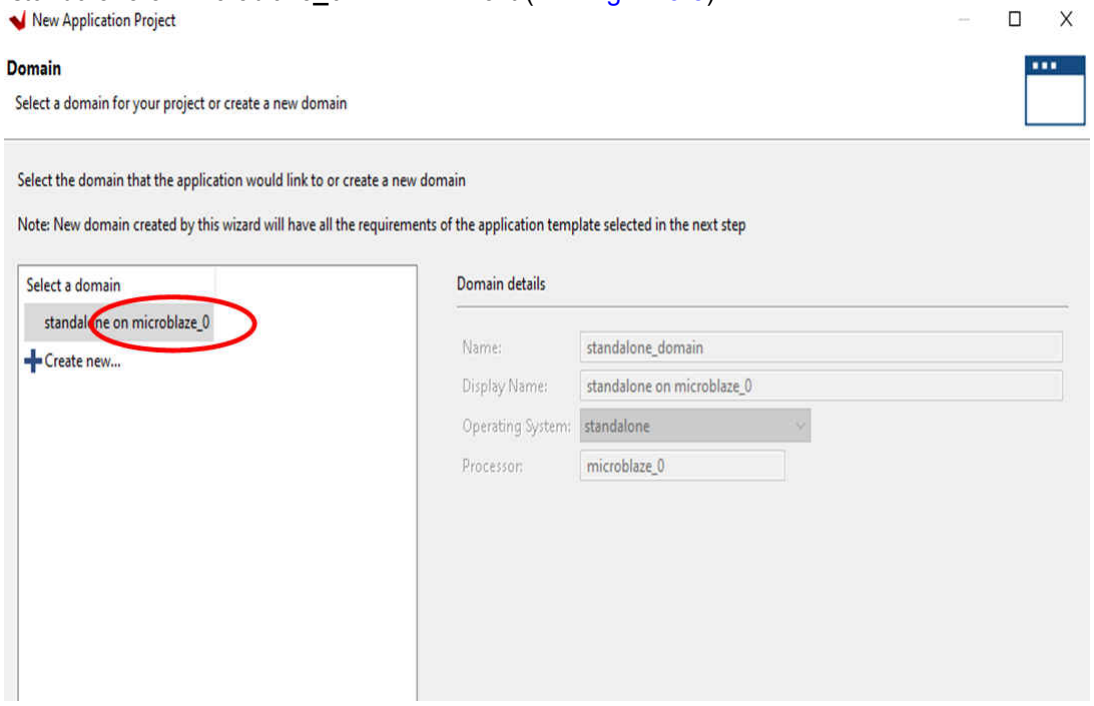


Figure 9-5. Application Project Name

6. Select *Hello World* from the list of templates and click *Finish* (see Figure 9-6).

Copyright © 2022 Texas Instruments Incorporated

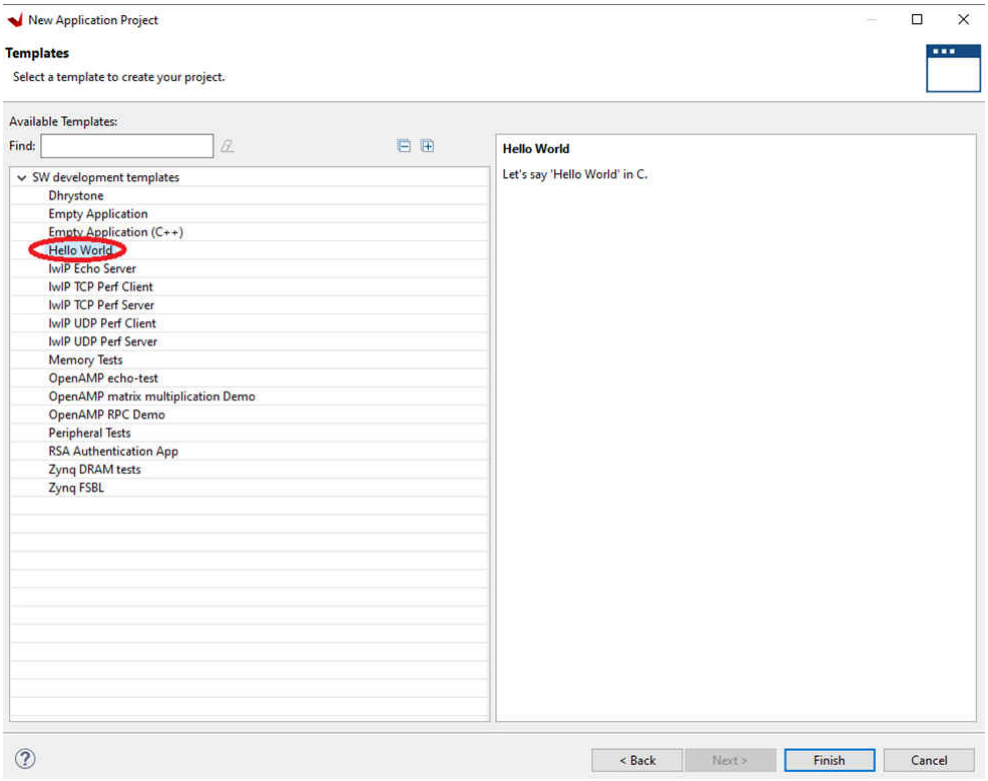


Figure 9-6. Selecting Template

7. A fresh C project appears on top where the actual application development can start.

Copyright © 2022 Texas Instruments Incorporated

10 Build Application Projects

To build an application project, follow these steps:

1. Right-click the application name and select *Build Project* (see Figure 10-1).

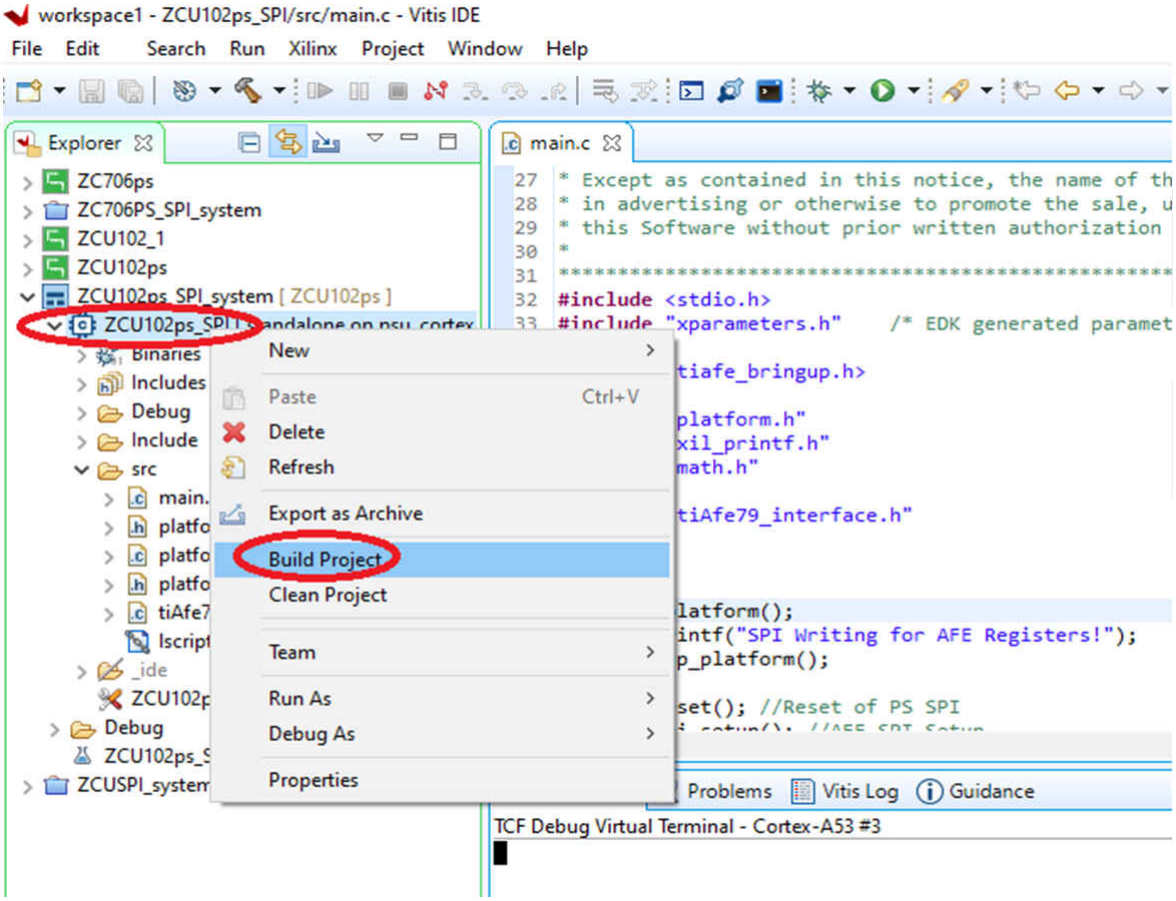


Figure 10-1. Building Project

2. Ensure that the project builds without any errors.

Copyright © 2022 Texas Instruments Incorporated

11 Configure the AXI GPIO

Sequence for AXI GPIO configuration should be:

1. Initialize GPIO module
2. Set Direction
3. Set High or Low for corresponding bits

The above sequence must be completed before initializing AFE and LMK devices.

11.1 Initializing the GPIO

The C syntax for initializing the AXI GPIO is in two steps:

1. XGpio GPOs: Initialize a pointer (GPOs) to the GPIO configuration register.
2. XGpio_Initialize(&GPOs, XPAR_AXI_GPIO_0_DEVICE_ID): Refer to *Xparameters.h* to find the correct AXI GPIO DEVICE ID.

11.2 Setting the Direction

XGpio_SetDataDirection(&GPOs, 1, 0);

First argument *&GPOs* point to the GPIO instance initialized in previous command.

Second argument 1 indicates the GPIO bank. In **TI SPI AFE IP**, only the first bank is used.

Third argument 0 indicates all GPIO bits are set for outputs.

11.3 Setting High or Low for Corresponding Bits

XGpio_DiscreteWrite(&GPOs, 1, regval);

First argument *&GPOs* point to the GPIO instance initialized in previous command.

Second argument 1 indicates the GPIO bank. In **TI SPI AFE IP**, only the first bank is used.

Table 11-1. Bit Mapping of IP I/Os

BIT DESCRIPTION	BIT POSITION
JESD RSTn	4
JESD TXRST	3
RSTn	2
RXTDD	1
TXTDD	0

For Example:

XGpio_DiscreteWrite(&GPOs, 1, 0x14);

This command sets *JESD RSTn* and *RSTn* to 1, sets all other bits to 0

Copyright © 2022 Texas Instruments Incorporated

12 Configure the AXI SPI

The AXI SPI instance in TI AFE SPI IP is used in *Standard Mode*.

Peripherals *select 0* and *select 1* are used as chip selects for AFE and LMK clocking device, respectively

SCL frequency is hard coded to 10 MHz within TI IP

Key commands for SPI initialization and usage in Vitis are as explained below:

1. *XSpi_Config *ConfigPtr*;

   Initialize a pointer (ConfigPtr).
2. *ConfigPtr = XSpi_LookupConfig(XPAR_AXI_QUAD_SPI_0_DEVICE_ID)*;

   Refer ‘Xparameters.h’ to find the correct AXI QUAD SPI DEVICE ID.
3. *XSpi_CfgInitialize(&Spidev, ConfigPtr, ConfigPtr->BaseAddress)*;

   Initialize a new instance of SPI (Spidev).
4. *XSpi_SetOptions(&Spidev, XSP_MASTER_OPTION)*;

   Set the Spidev instance to be in Controller mode.
5. *XSpi_Start(&Spidev)*;
6. *XSpi_SetSlaveSelect(&Spidev, 1)*;

   Select Peripheral: AFE.
7. *XSpi_SetSlaveSelect(&Spidev, 2)*;

   Select Peripheral: LMK.
8. *XSpi_Transfer(&Spidev, WrBufdev, RdBufdev, 3);*

   Second argument WrBufdev is an array with 3 bytes (24-bit data to be transmitted on SPI).

   Third argument RdBufdev is an array with 3 bytes, the last byte has the SPI read value.

   Fourth argument is number of bytes to be transmitted/received...3 in our case.

The D23 is the MSB bit of the 24-bit data because the D23 indicates whether it is a Read or a Write SPI operation:

- If D23 is set to 1, then it is a read operation and RdBufdev[2] stores the read back address contents
- If D23 is set to 0, then it is a write operation and RdBufdev[2] has no significance

Copyright © 2022 Texas Instruments Incorporated

13 Create Boot Images to Run on SD Card

This section of steps is applicable only if there is a need to create a bootable SD Card. Skip these steps if not applicable.

To create a bootable SD Card with the active application, follow these steps:

1. Right-click the application name to open the drop-down menu and select *Create Boot Image* (see Figure 13-1).

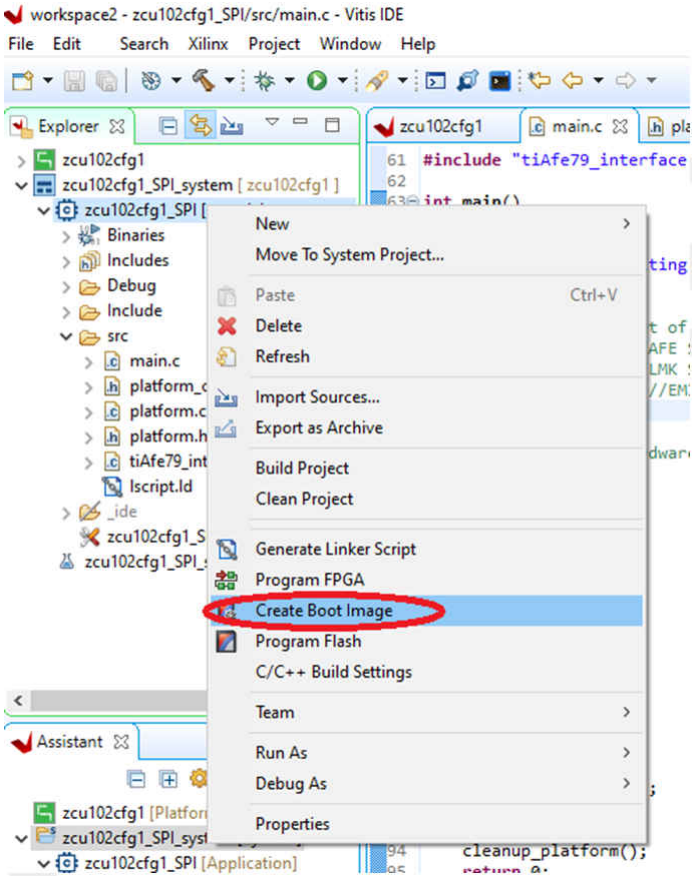


Figure 13-1. Creating Boot Image

2. Select *Zynq MP* from the *Architecture* menu and select *Import from existing BIF file* option (see Figure 13-2).

   The *Output path* shows the file location of the *BOOT.bin* to be generated.

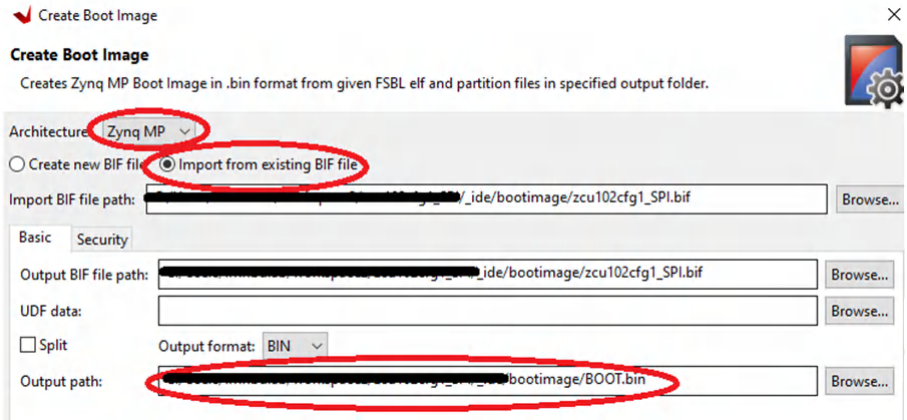


Figure 13-2. Boot Image Input and Output Pathnames

3. After reviewing the locations of the boot elf file, FPGA bit file, and application elf, click *Create Image* (see Figure 13-3).

Copyright © 2022 Texas Instruments Incorporated

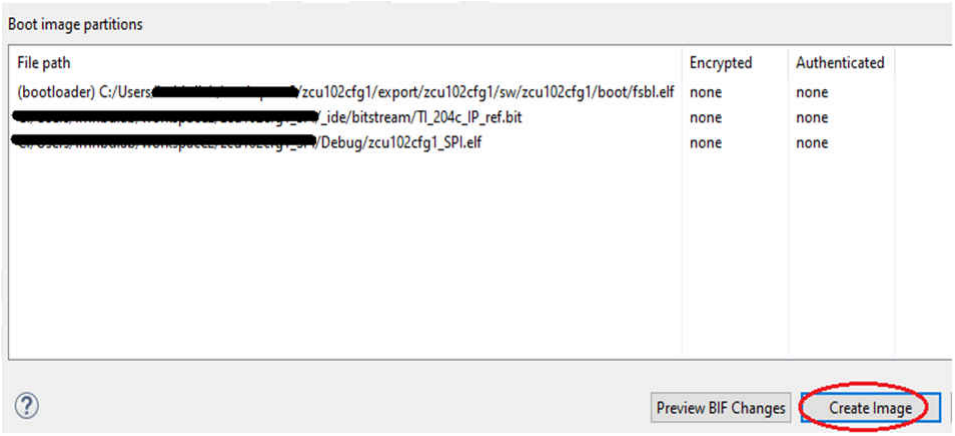


Figure 13-3. Boot Image Partitions

The above step generates the *BOOT.bin* in the output path as selected in Figure 13-2.

4. Copy *BOOT.bin* to a FAT16 or FAT32 formatted SD Card.
5. To boot from SD Card, ensure the SW6 switch positions on the ZCU102 is as per Figure 13-4.

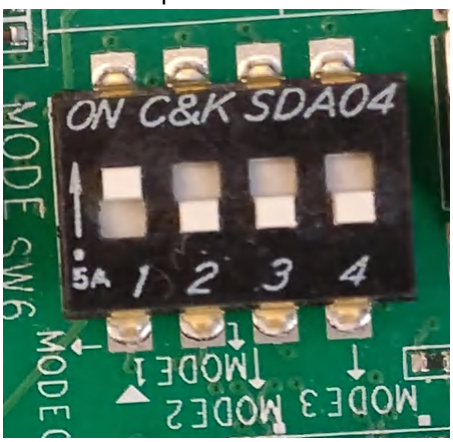


Figure 13-4. SW6 Switch Positions

6. With the above setting on SW6 and SD card inserted, ZCU102 can now directly boot with the application.

Copyright © 2022 Texas Instruments Incorporated

14 Set up and Power on Hardware

To set up and power on the hardware, follow these steps:

1. Dock the AFE EVM to J5 (HPC0) FMC on the ZCU102 board.
2. Connect a 1.5-GHz signal through a clock source to RFROM EVM at J14.
3. Connect J2 (JTAG) and J83 (UART) USB connectors from ZCU102 FPGA board to the computer.
4. Connect the 12-V Xilinx EVM Adapter for the ZCU102 at J52.
5. After all the above connections are made, power up the setup. Note that the AFE EVM in this example is completely powered by the ZCU102 FMC interface.

Copyright © 2022 Texas Instruments Incorporated

15 Set up ZCU102 Board Interface for VADJ_FMC

To set the ZCU102 board interface for VADJ_FMC, follow these steps:

1. Execute the *ZCU102-Board User Interface* software (available for download from Xilinx.com).
2. Select the appropriate COM Port to enable communication between the onboard MSP430 of the ZCU102 and the PC. This software is required to turn on the FMC_AUX supply of 1.8 V for the FMC bank of FPGA (see Figure 15-1).

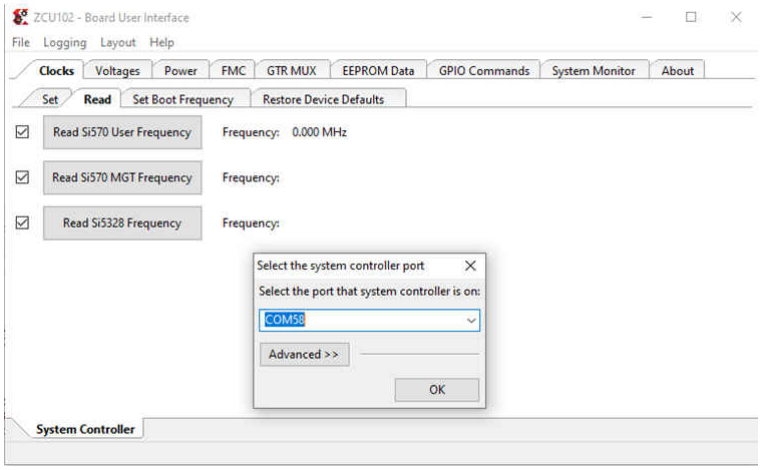


Figure 15-1. ZCU102 Board User Interface

3. Select the *Set VADJ to 1.8 V* check box (see Figure 15-2).

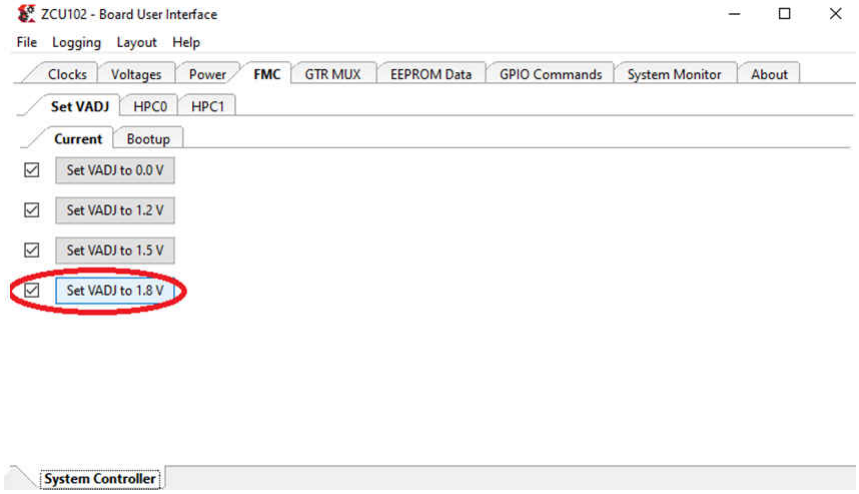


Figure 15-2. Setting VADJ

4. Confirm the same by reading the VADJ_FMC voltage. The voltage value must be 1.80 V (see Figure 15-3).

Copyright © 2022 Texas Instruments Incorporated

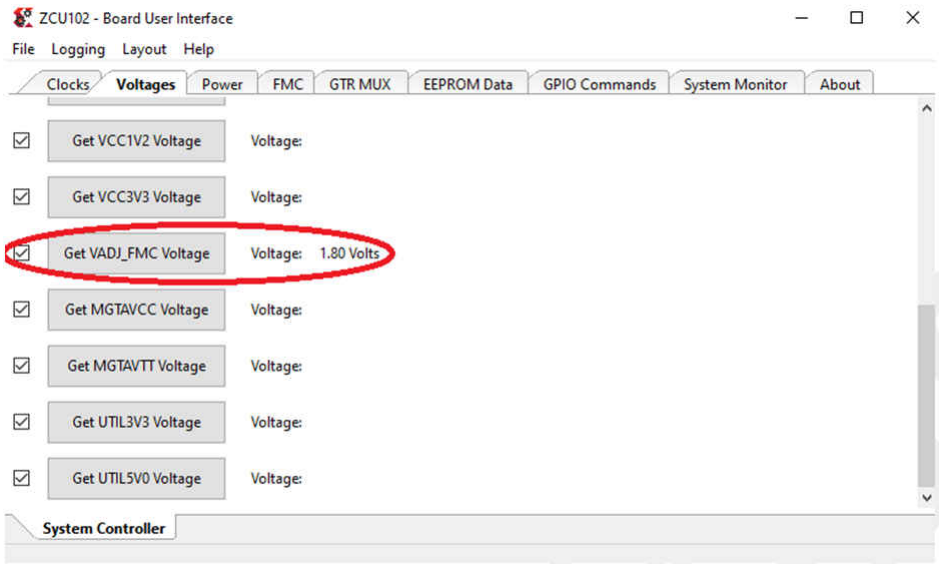


Figure 15-3. VADJ_FMC Voltage

Copyright © 2022 Texas Instruments Incorporated

16 Debug Application Projects and Set up Vitis Serial Terminal

To debug the application project and set up the Vitis serial terminal. follow these steps:

1. Right-click the project name and go to *Debug As* from the drop-down menu. Click *Launch on Hardware (Single Application Bug)* to run the debug (see Figure 16-1).

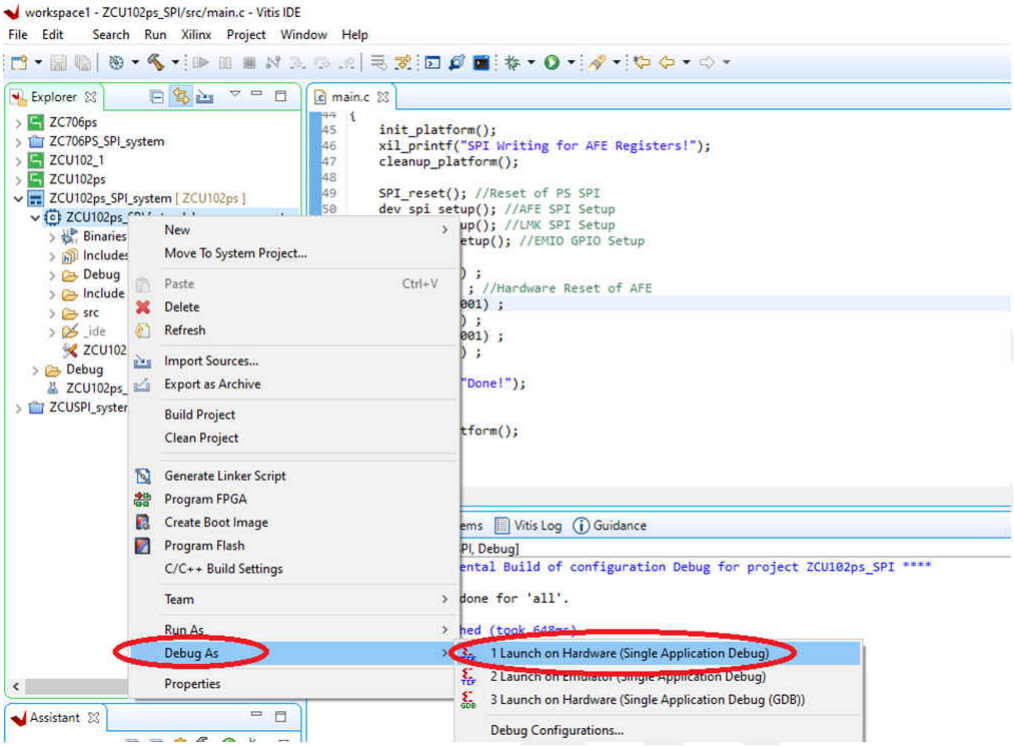


Figure 16-1. Debugging Application Project

2. Connect the Vitis Serial terminal (see Figure 16-2) with baudrate 115200 (this can be used to see SPI write or read status).

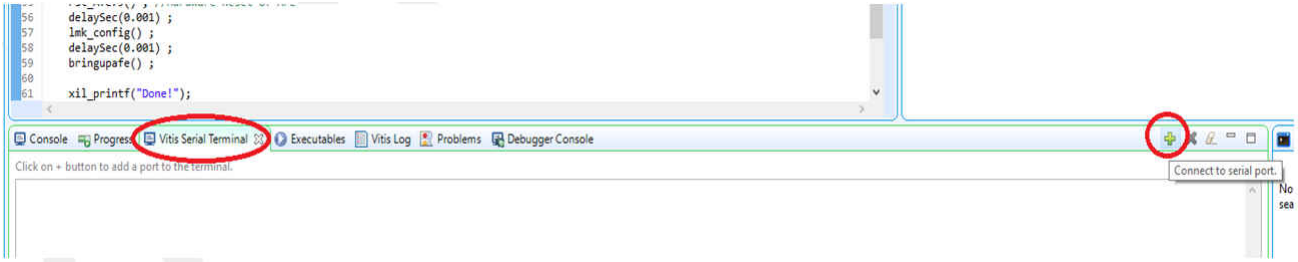


Figure 16-2. Vitis Serial Terminal

Copyright © 2022 Texas Instruments Incorporated

17 Execute the Application

To execute an application run, follow these steps:

1. Click the right arrow button as shown in Figure 17-1.

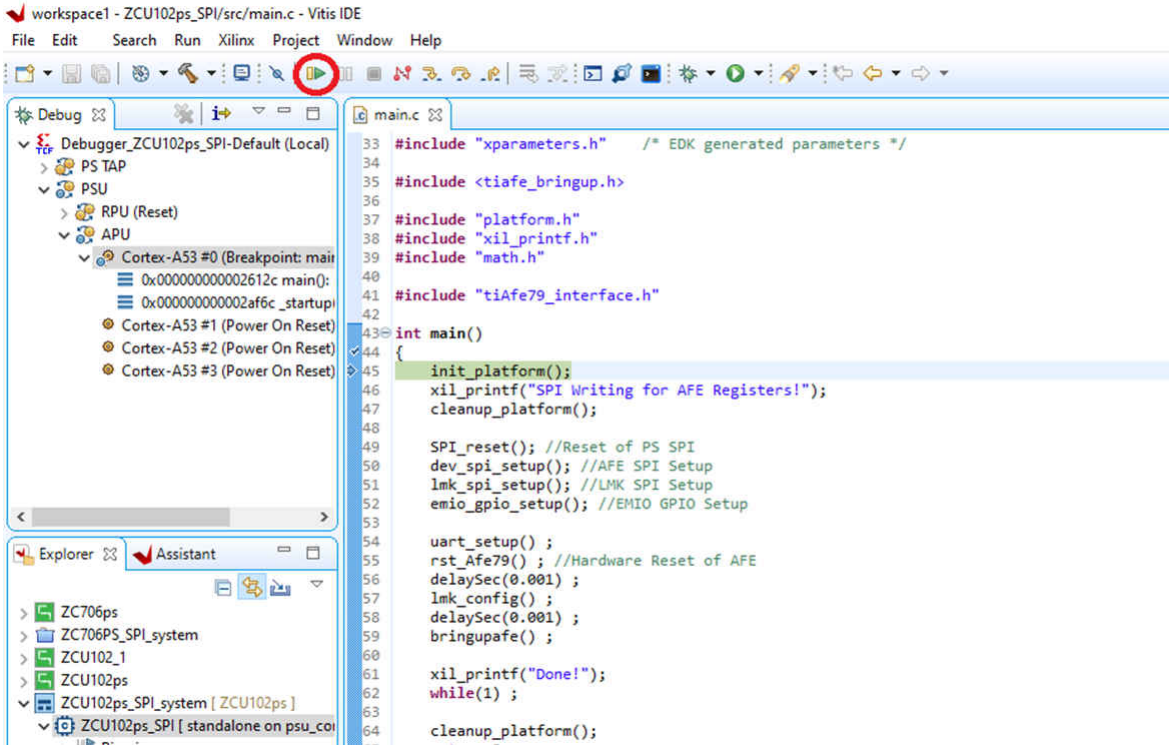
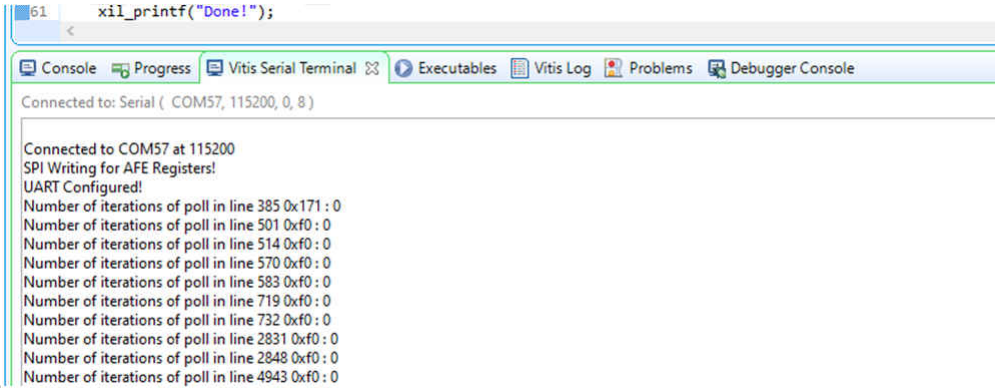


Figure 17-1. Executing Application

2. Notice the SPI logs being printed on the UART terminal.



Copyright © 2022 Texas Instruments Incorporated

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 will 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 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.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated