



Programming External SPI Flash through JTAG for ECP5/ECP5-5G

Technical Note

FPGA-TN-02050 Version 1.0

October 2017

Contents

Acronyms in This Document	4
1. Introduction	5
1.1. Features.....	5
2. Programming External SPI Flash	6
2.1. SPI Flash Programming over USB (System A)	6
2.2. SPI Flash Programming through Microcontroller (System B)	7
3. Configuring the Master SPI port using the CFG[2:0] Pins on the ECP5/ECP5-5G Device.....	8
4. Enabling ECP5 Master SPI Port using Diamond Software	9
5. Generating the SPI File using Deployment Tool for External Quad SPI Flash	11
6. Programming the SPI Flash with Bitstream File using Diamond Programmer.....	16
7. Generating Embedded JTAG Programming File using Deployment Tool.....	18
Technical Support Assistance	21
Revision History	21

Figures

Figure 1.1. SPI Flash Programming Block Diagram	5
Figure 2.1. SPI Flash Programming using USB for Lattice Evaluation Board	6
Figure 2.2. Device Configuration through Microcontroller	7
Figure 4.1. Synthesize Design in Process Tab	9
Figure 4.2. Spreadsheet View Button	9
Figure 4.3. Global Preferences Tab	10
Figure 5.1. Getting Started Dialog Box	11
Figure 5.2. Step 1 of 4: Select Input File(s)	12
Figure 5.3. Step 2 of 4: Advance SPI Flash Options	13
Figure 5.4. Step 3 of 4: Select Output File(s)	14
Figure 5.5. Step 4 of 4: Generate Deployment	15
Figure 6.1. Programmer Main Interface	16
Figure 6.2. Device Properties Dialog Box	16
Figure 6.3. Program Button on Main Interface	17
Figure 6.4. Program Button in Design Menu Options	17
Figure 7.1. Deployment Tool – Getting Started Dialog Box	18
Figure 7.2. Step 1 of 4: Select Input File(s)	19
Figure 7.3. Step 2 of 4: JTAG Full VME Embedded Options	19
Figure 7.4. Step 3 of 4: Select Output File(s)	20
Figure 7.5. Step 4 of 4: Generate Deployment	20

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
FTDI	Future Technology Devices International
JTAG	Joint Test Action Group
SPI	Serial Peripheral Interface

1. Introduction

The Lattice Semiconductor ECP5™ device family has a built-in hardware capability to support the programming of an external SPI flash boot device. This internal hardware capability, along with the Lattice Diamond® Programmer software tool, enable the designer to program the external Serial Peripheral Interface (SPI) Flash through the ECP5 JTAG port.

The simplified diagram (Figure 1.1) shows the ECP5 JTAG programming interface port connected to a PC or embedded processor to ultimately drive the ECP5 Master SPI port to program the SPI Flash. As a blank device, the ECP5 hardware default settings enable the JTAG to Master SPI Port interface within the device. When the ECP5 is configured with a user defined bitstream, the FPGA designer has to make sure that the Master SPI Port is enabled through the Diamond Spreadsheet View as part of the user design.

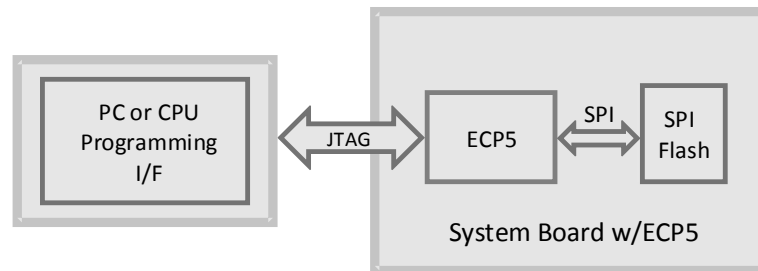


Figure 1.1. SPI Flash Programming Block Diagram

1.1. Features

The following features are supported for external SPI Flash programming:

- Remote/Field upgrade of external SPI Flash
- Parameters to support all standard SPI Flash vendors with different densities
- Software platform for both Windows and Linux OS
- Support for dual boot configuration
- Other operations
 - Program primary and golden images
 - Protect/Unprotect golden image
 - Verify bitstreams
 - Bulk erase
 - Erase primary/golden bitstreams
 - Read and save bitstreams to a file

2. Programming External SPI Flash

ECP5 supports the programming of external SPI Flash using two systems:

- [SPI Flash Programming over USB \(System A\)](#)
- [SPI Flash Programming through Microcontroller \(System B\)](#)

2.1. SPI Flash Programming over USB (System A)

The external SPI flash can be programmed via the JTAG port of the ECP5 device using Diamond Programmer running on a PC with a USB port. There are two types of USB cables used to connect to the ECP5 device:

- Standard passive USB cable with micro Type-B connector; and
- Lattice proprietary programming cables with fly-wire connections (for example, HW-USBN-2B).

The type of USB cable used is determined by the target board architecture.

For typical Lattice evaluation and development boards and kits, the standard USB cable is used. An onboard FTDI device bridges the USB to the JTAG port of the Lattice device (System A.1 as shown in [Figure 2.1](#)). For user designs with a direct JTAG connection to the Lattice device, the Lattice proprietary cable is used (System A.2 as shown in [Figure 2.2](#)).

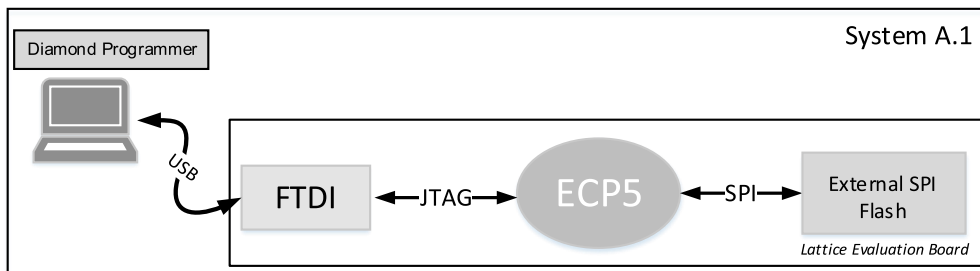


Figure 2.1. SPI Flash Programming using USB for Lattice Evaluation Board

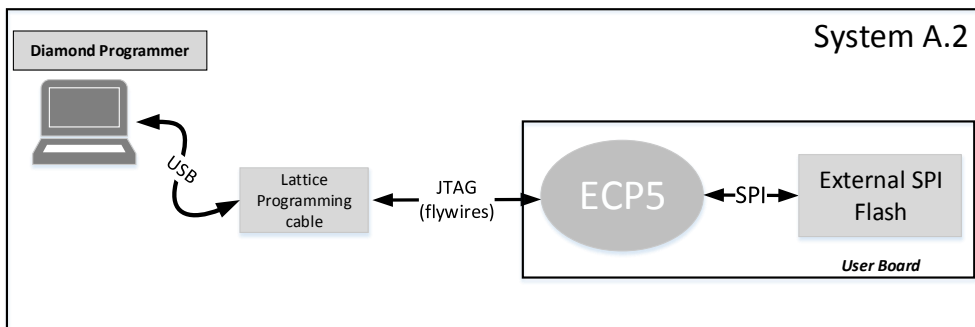


Figure 2.2. SPI Flash Programming using Lattice Programming Cable

To configure the ECP5 device and program the external SPI flash using the Lattice programming cable, perform the procedures in the following sections in the order they are listed below.

- [Configuring the Master SPI port using the CFG\[2:0\] Pins on the ECP5/ECP5-5G Device](#) (Section 3)
- [Enabling ECP5 Master SPI Port using Diamond Software](#) (Section 4)
- [Programming the SPI Flash with Bitstream File using Diamond Programmer](#) (Section 6)

2.2. SPI Flash Programming through Microcontroller (System B)

The external SPI flash can be programmed via the JTAG port of the Lattice ECP5 device using an embedded processor or microcontroller running Lattice ispVMEmbedded code. In Figure 2.2 below, an example system is shown in which a microcontroller is used to program the external SPI Flash through Lattice ECP5 device via JTAG port.

Lattice Diamond SW tool is used to generate a plain bitstream file to program the ECP5 device. The external SPI Flash can either be a x1 (1-bit data output) or x4 (4-bit data output/quad). For x1 SPI flash the *.mcs file is generated by the Diamond SW tool and located under the implementation folder of the appropriate project. In the case of quad SPI flash, the Deployment tool is necessary to generate a *.mcs file from the appropriate bitstream file (.bit file). To generate the micro code for embedded applications this *.mcs file is passed down through the ispVM Embedded (via Deployment Tool) tool.

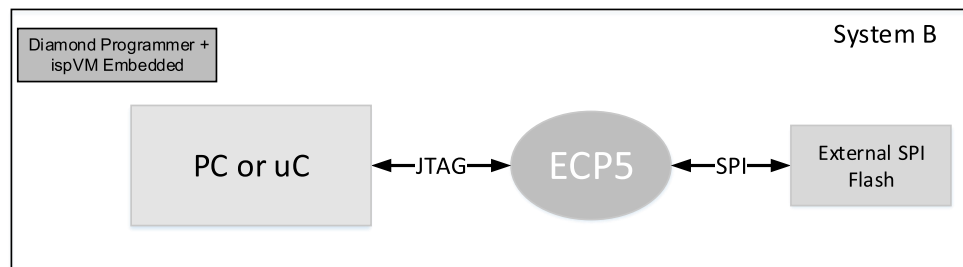


Figure 2.3. Device Configuration through Microcontroller

To configure the ECP5 device and program the external SPI flash through a microcontroller, perform the procedures in the following sections in the order they are listed below.

- [Configuring the Master SPI port using the CFG\[2:0\] Pins on the ECP5/ECP5-5G Device](#) (Section 3)
- [Enabling ECP5 Master SPI Port using Diamond Software](#) (Section 4)
- [Generating the SPI File using Deployment Tool for External Quad SPI Flash](#) (Section 5)
- [Programming the SPI Flash with Bitstream File using Diamond Programmer](#) (Section 6)
- [Generating Embedded JTAG Programming File using Deployment Tool](#) (Section 7)

3. Configuring the Master SPI port using the CFG[2:0] Pins on the ECP5/ECP5-5G Device

CFG[2:0] are dedicated configuration mode pins used to select the configuration mode of the ECP5/ECP5-5G device.

To enable Master SPI configuration port, apply the settings below for the CFG pins on the ECP5/5G device:

CFG[2] = 0

CFG[1] = 1

CFG[0] = 0

Note: “0” = Tie signal to GND, “1” = Pull up signal to VCCIO8.

Refer to the Configuration Considerations section in [ECP5 and ECP5-5G Hardware Checklist](#) (FPGA-TN-02038) for detailed information on the PCB requirements for JTAG and SPI interfaces.

4. Enabling ECP5 Master SPI Port using Diamond Software

Create the ECP5-5G design that is to be programmed into the SPI flash by providing the appropriate RTL source file.

1. In the **Process** tab, run the **Synthesize Design** step to synthesize the design. Check marks, as shown in [Figure 4.1](#), indicate that the step is successfully completed.

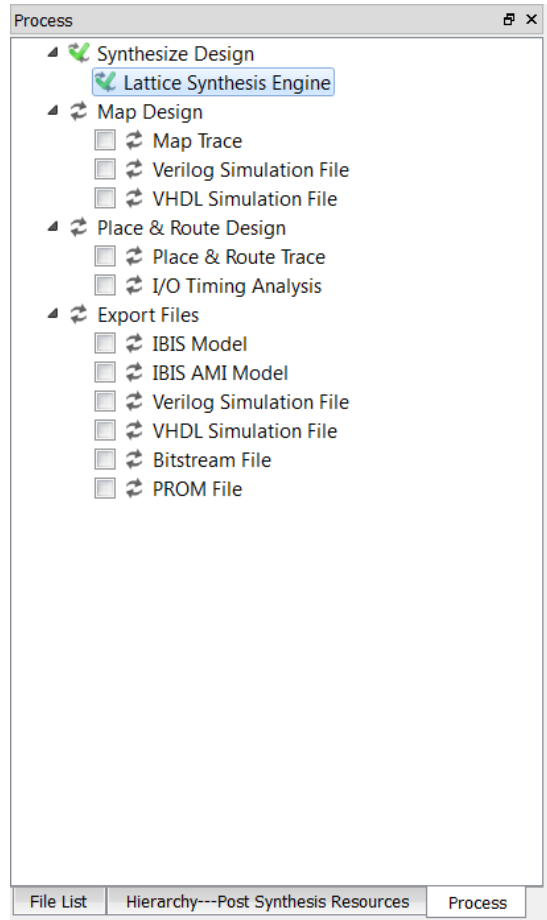


Figure 4.1. Synthesize Design in Process Tab

2. Click the **Spreadsheet View** button or open it from the **Tools** menu on top of the ribbon, as shown in [Figure 4.2](#), to open the Spreadsheet View tab where you can edit the configuration properties of the device.

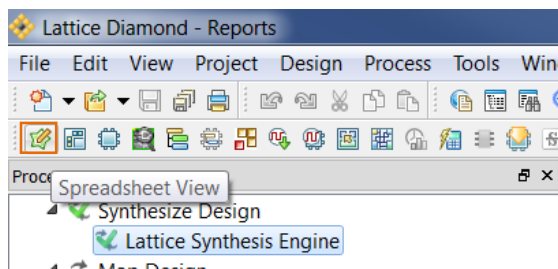


Figure 4.2. Spreadsheet View Button

- In Spreadsheet View, go to the **Global Preferences** tab, shown in Figure 4.3, to edit the configuration settings of the device. Select the options below to enable the Master SPI port of the ECP5/ECP5-5G device. Save these settings.

MASTER_SPI_PORT: Enable

MCCLK_FREQ: 62

- In the **Process** pane, make sure that the **Bitstream File** option is checked and generate the bitstream by double clicking **Export Files**. You can verify the operation in the **Output** pane also shown in Figure 4.3.

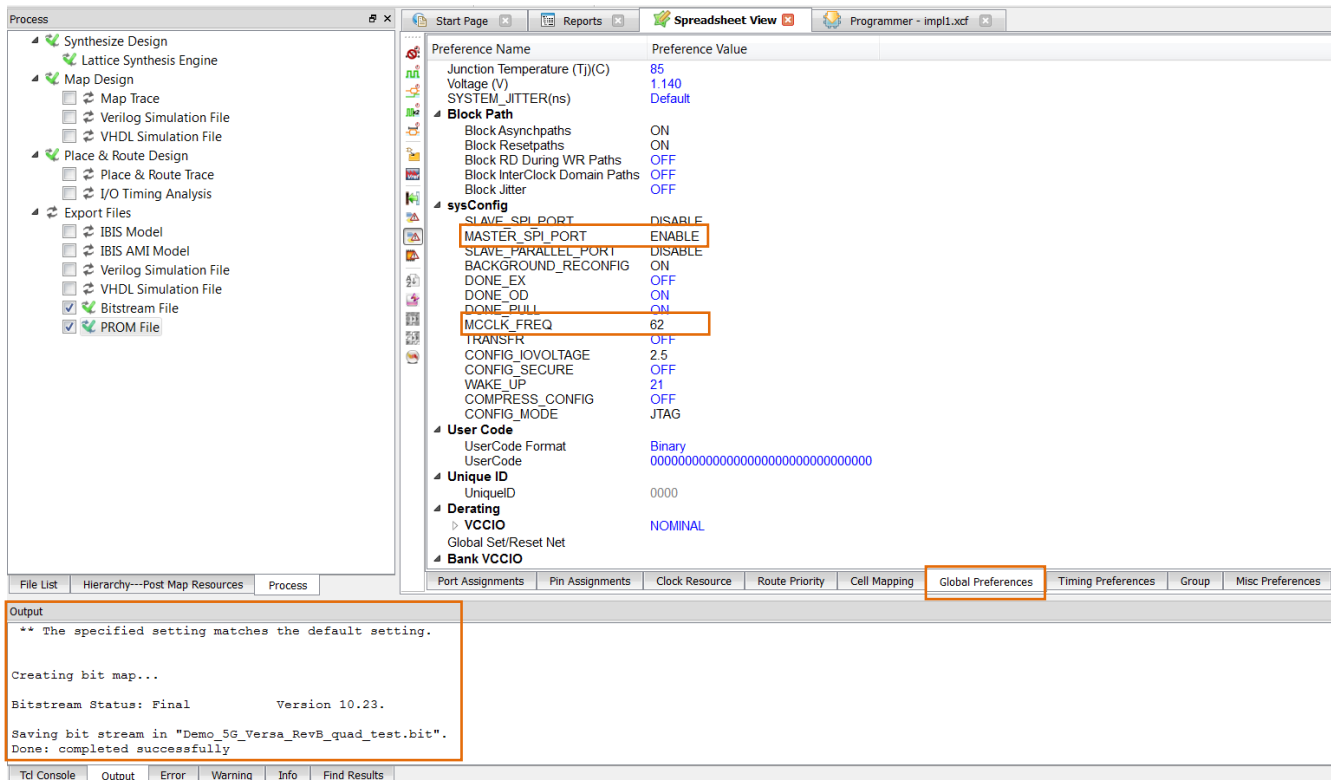


Figure 4.3. Global Preferences Tab

Once the bitstream is generated, the *.bit file is created under the implementation folder.

5. Generating the SPI File using Deployment Tool for External Quad SPI Flash

Diamond software generates a *.bit file. The Deployment tool found under the Accessories of Diamond tool is used to generate the Quad SPI mode *.mcs file used to program the Quad SPI Flash from the *.bit file. Open Deployment tool.

1. In the **Getting Started** dialog box, select the options below, also shown in [Figure 5.1](#), and click **OK**.

Function Type: External Memory and

Output File Type: Advanced SPI Flash

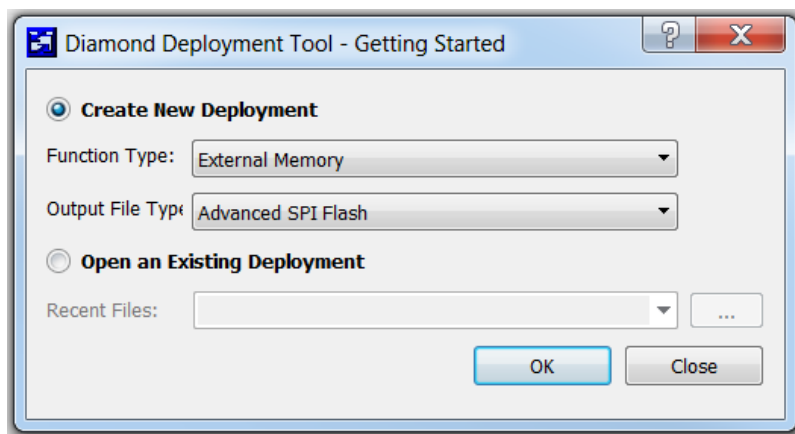


Figure 5.1. Getting Started Dialog Box

2. In **Step 1 of 4: Select Input File(s)**, shown in [Figure 5.2](#), select the input file in **File Name**. The input file is a *.bit file. Click **Next**.

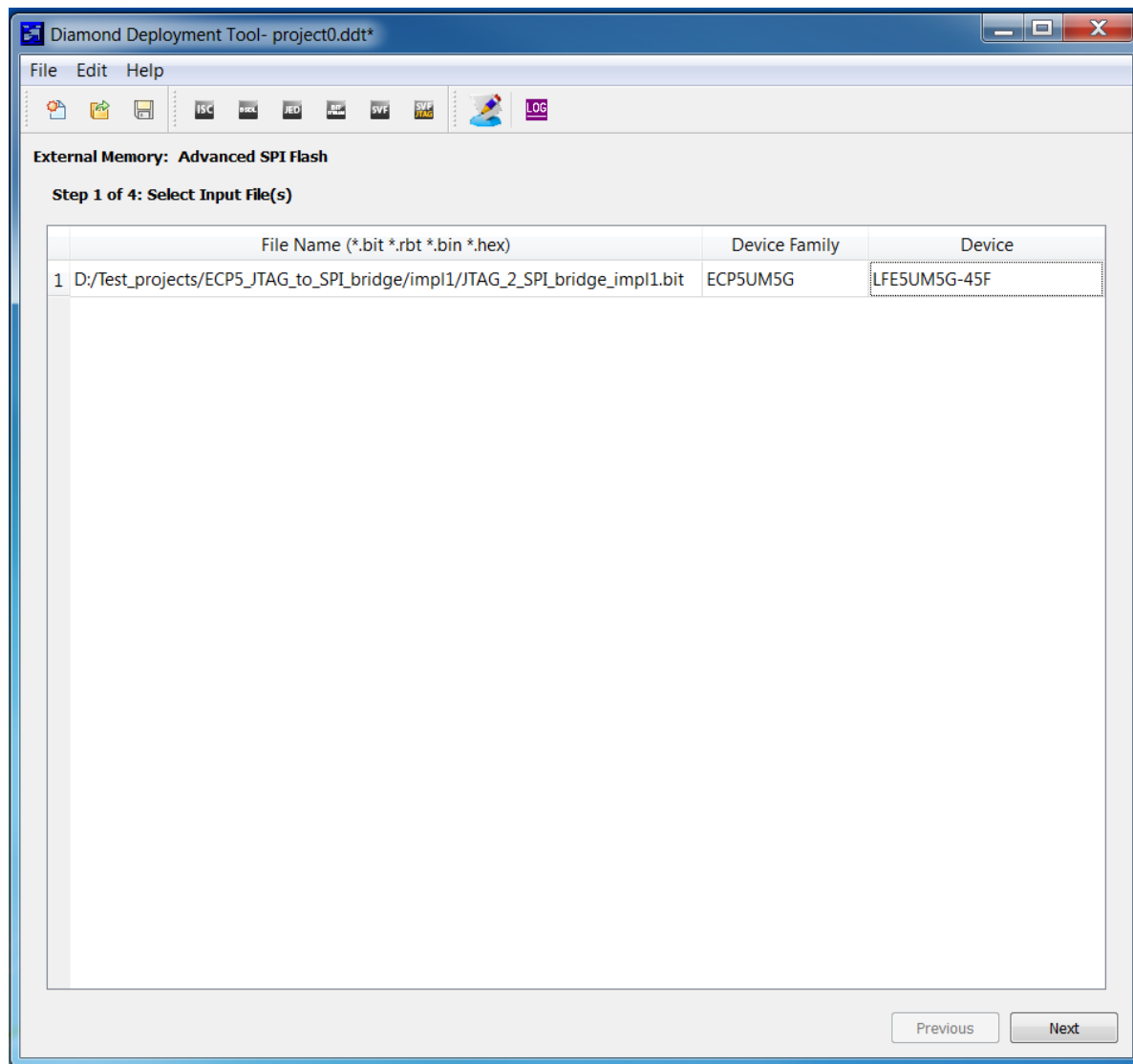


Figure 5.2. Step 1 of 4: Select Input File(s)

3. In **Step 2 of 4: Advance SPI Flash Options**, shown in [Figure 5.3](#), select the options below and click **Next**.
SPI Flash Size (Mb): 128 and
SPI Flash Read Mode: Quad I/O SPI Flash Read

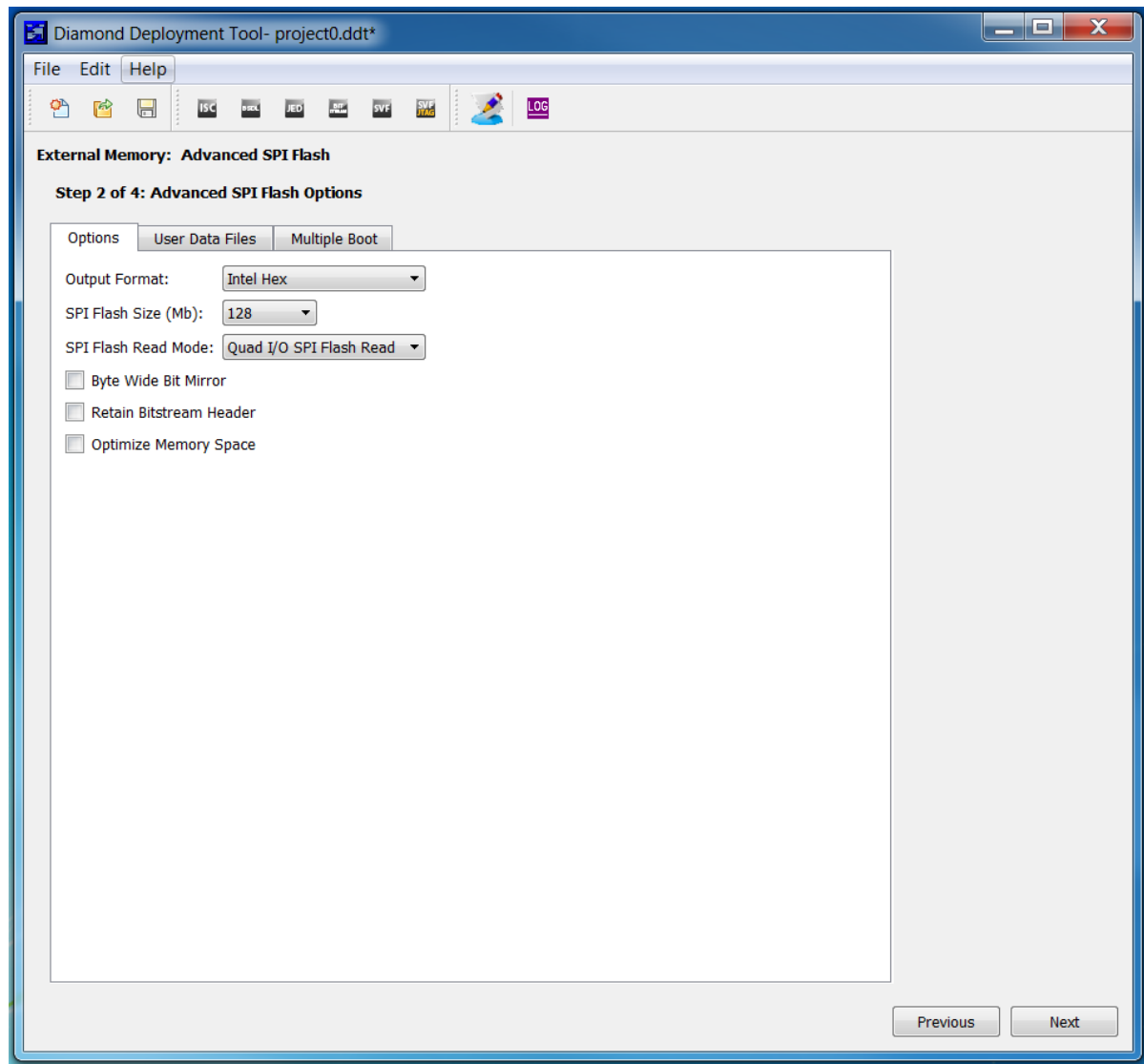


Figure 5.3. Step 2 of 4: Advance SPI Flash Options

4. In **Step 3 of 4: Select Output File(s)**, shown in [Figure 5.4](#), select the output file (*.mcs) in **Output File1** and click **Next**.

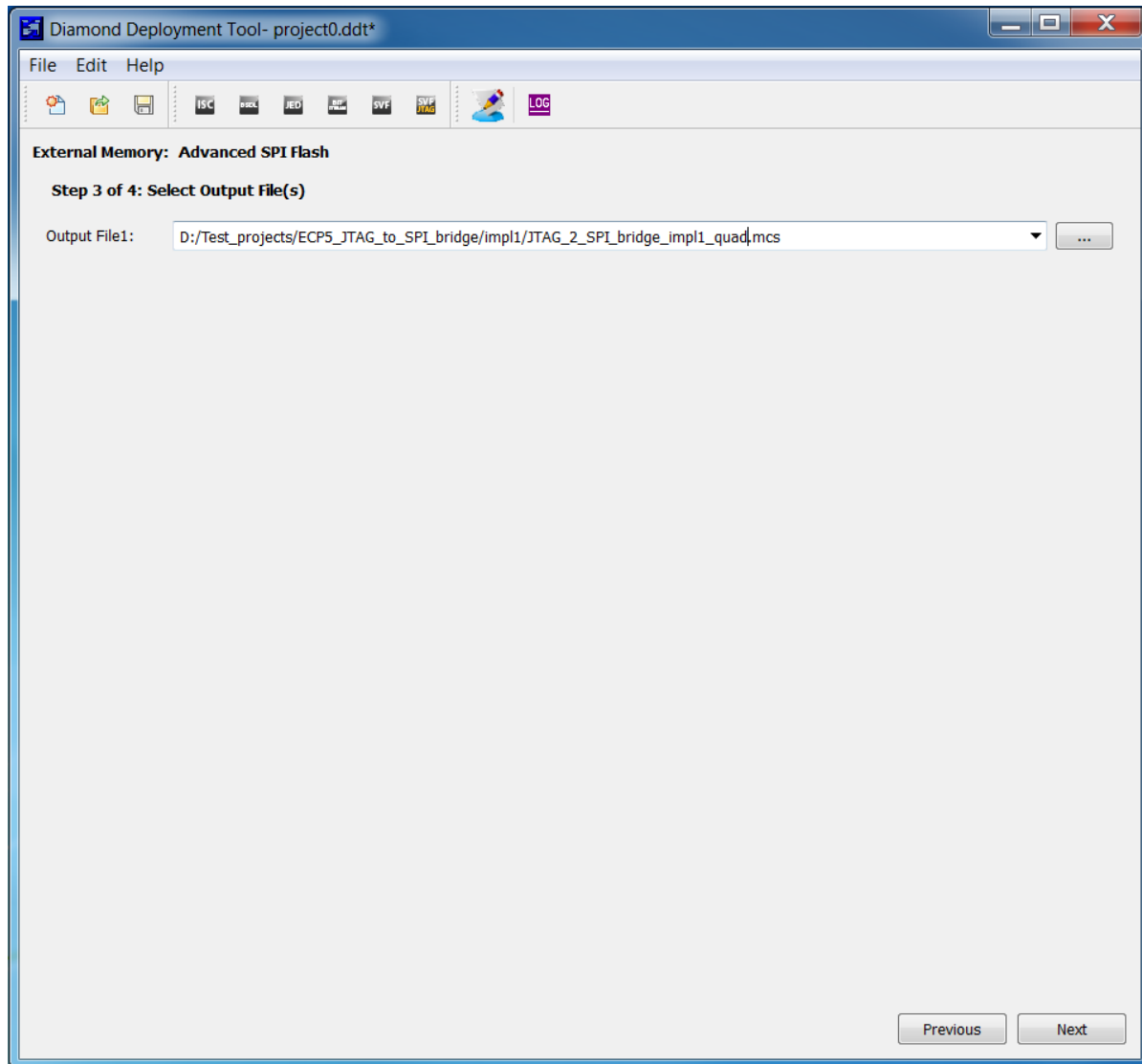


Figure 5.4. Step 3 of 4: Select Output File(s)

5. In **Step 4 of 4: Generate Deployment**, shown in [Figure 5.5](#), click **Generate** to generate the *.mcs file from *.bit file.

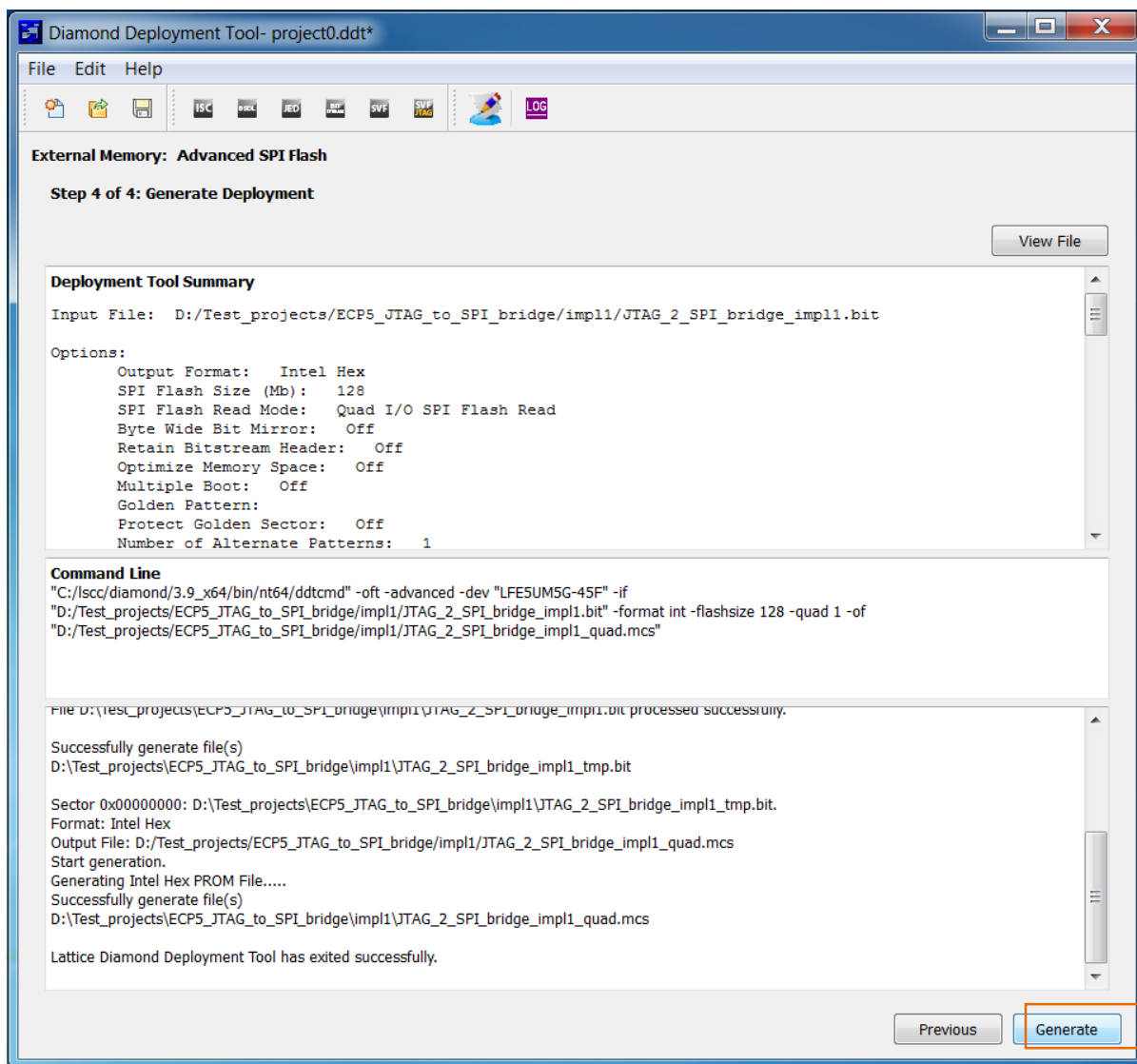


Figure 5.5. Step 4 of 4: Generate Deployment

6. Programming the SPI Flash with Bitstream File using Diamond Programmer

1. Open Diamond Programmer. Select the appropriate device using the Device Family and Device drop down options as shown in [Figure 6.1](#) (example of ECP5-5G versa board).

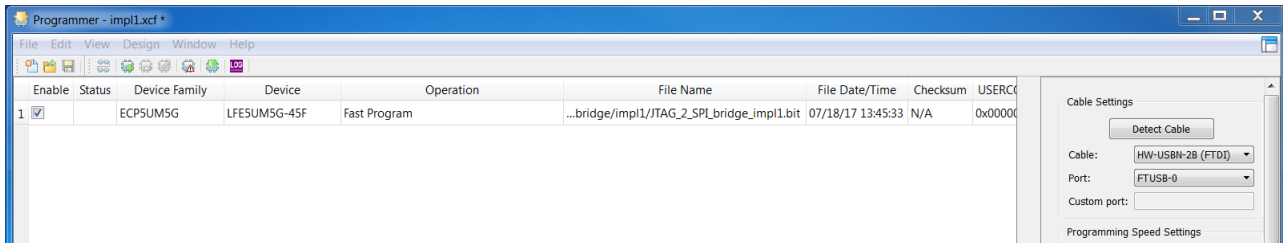


Figure 6.1. Programmer Main Interface

2. Double click the **Operation** tab to open the **Device Properties** dialog box.
3. Select **Programming file** and appropriate vendor and device information from SPI Flash Options section in the GUI. Click **OK**. An example of ECP5 Versa board is shown below, in [Figure 6.2](#).

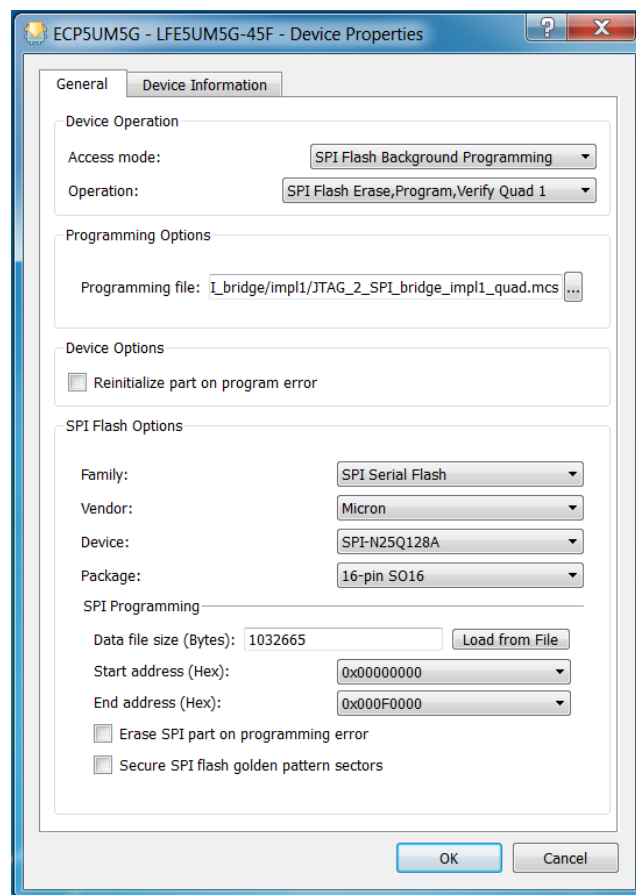


Figure 6.2. Device Properties Dialog Box

4. To program the SPI Flash, click the **Program** button either from the main interface as shown in Figure 6.3 or from the **Design** menu options as shown in Figure 6.4.

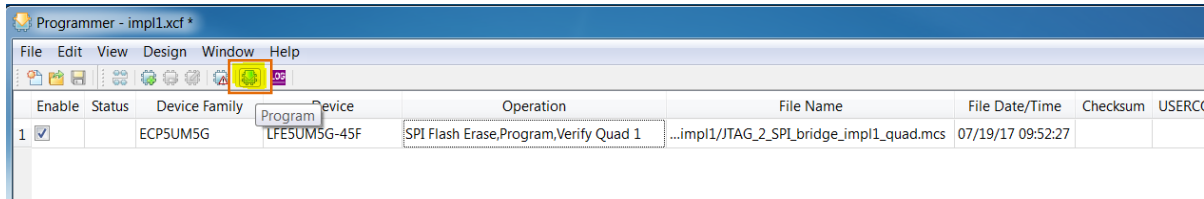


Figure 6.3. Program Button on Main Interface

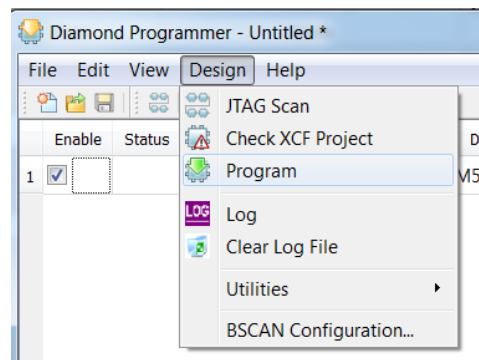


Figure 6.4. Program Button in Design Menu Options

7. Generating Embedded JTAG Programming File using Deployment Tool

The Deployment tool is used to generate the *.vme file for embedded JTAG programming.

To generate the *.vme file in Deployment tool:

1. Open the Deployment tool.
2. In the **Getting Started** dialog box, shown in [Figure 7.1](#), select **Create New Deployment**. Select the following options and click **OK**.

Function Type: Embedded System

Output File Type: JTAG Full VME Embedded

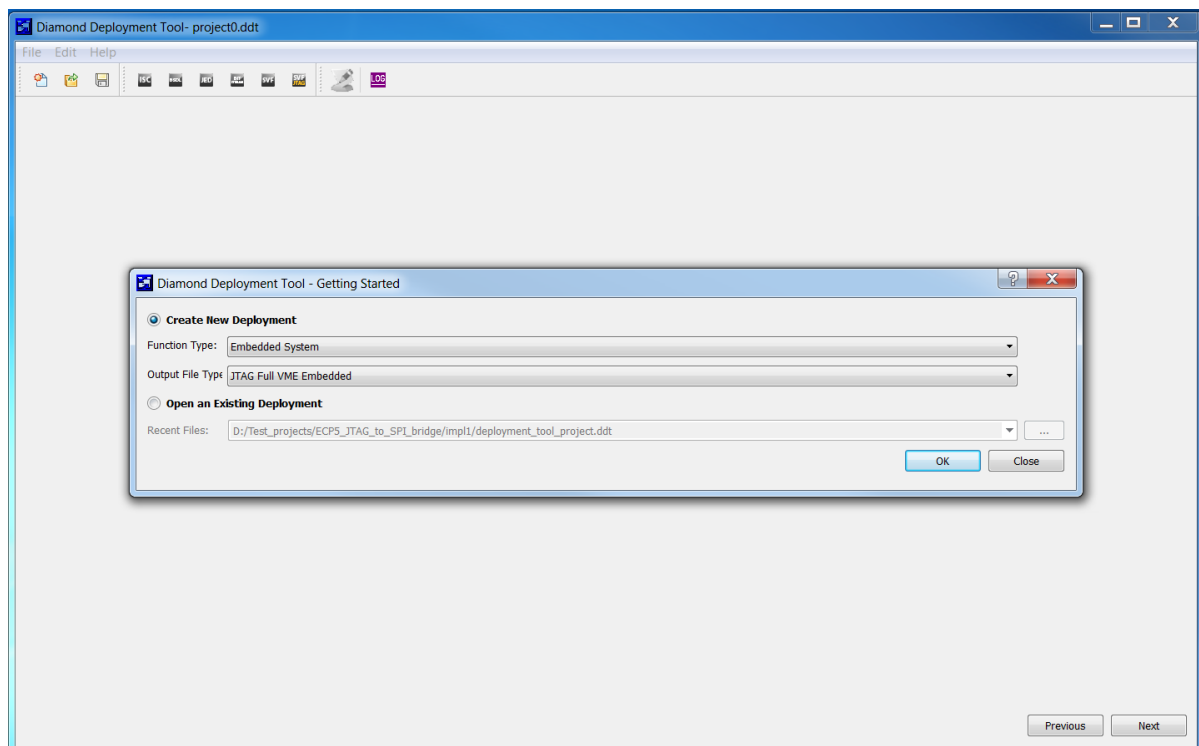


Figure 7.1. Deployment Tool – Getting Started Dialog Box

3. In **Step 1 of 4: Select Input File(s)**, shown in [Figure 7.2](#), select the input file (*.xcf) generated by Diamond Programmer in **Input XCF file**. Verify if the operation is correct and click **Next**.

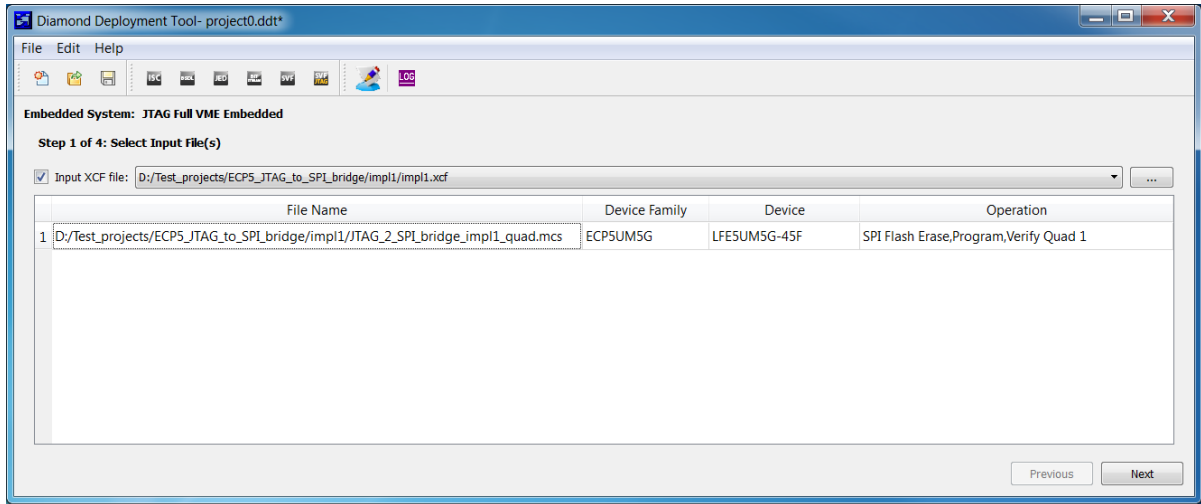


Figure 7.2. Step 1 of 4: Select Input File(s)

4. In **Step 2 of 4: JTAG Full VME Embedded Options**, shown in [Figure 7.3](#), select **Convert VME files to HEX (.c)** for **File-Based Embedded VME** and click **Next**.

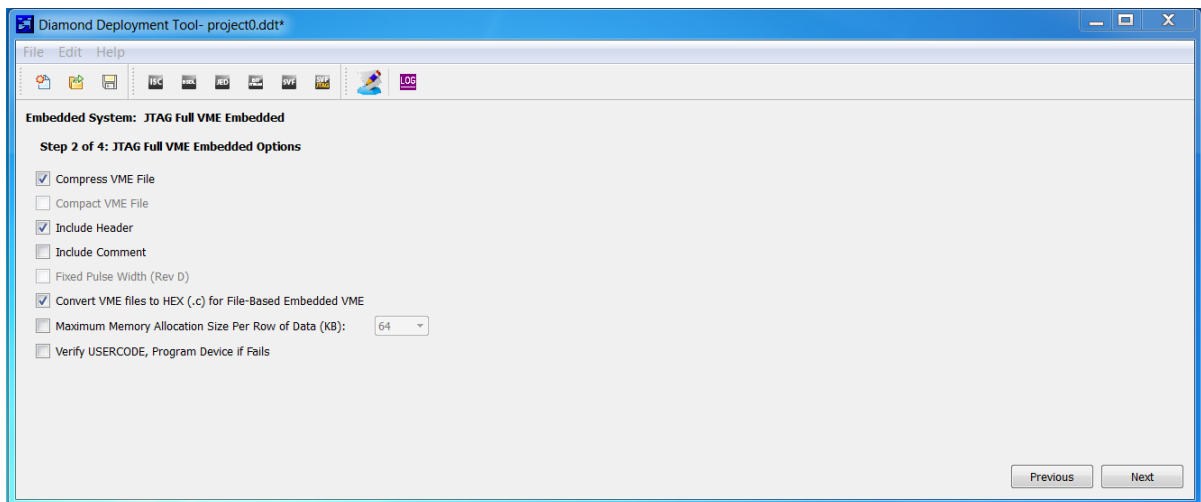


Figure 7.3. Step 2 of 4: JTAG Full VME Embedded Options

- In **Step 3 of 4: Select Output File(s)**, shown in [Figure 7.4](#), check the output file (*.vme) path in **Output File1** and click **Next**.

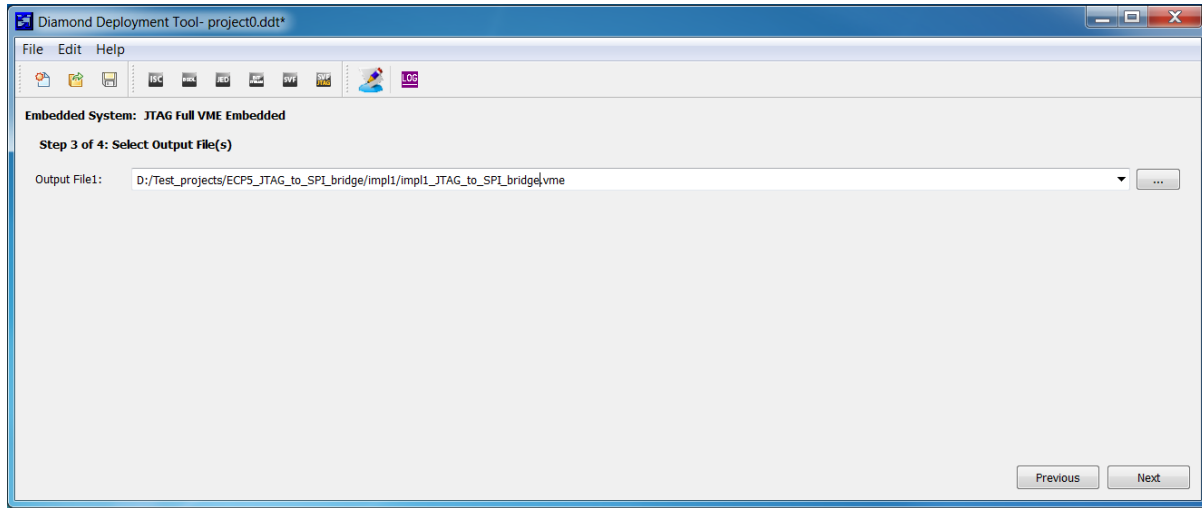


Figure 7.4. Step 3 of 4: Select Output File(s)

- In **Step 4 of 4: Generate Deployment**, shown in [Figure 7.5](#), click **Generate** to generate the *.vme file that is used by the embedded application.

Once the operation is completed, **Build VME File Operation: Successfully** message is displayed.

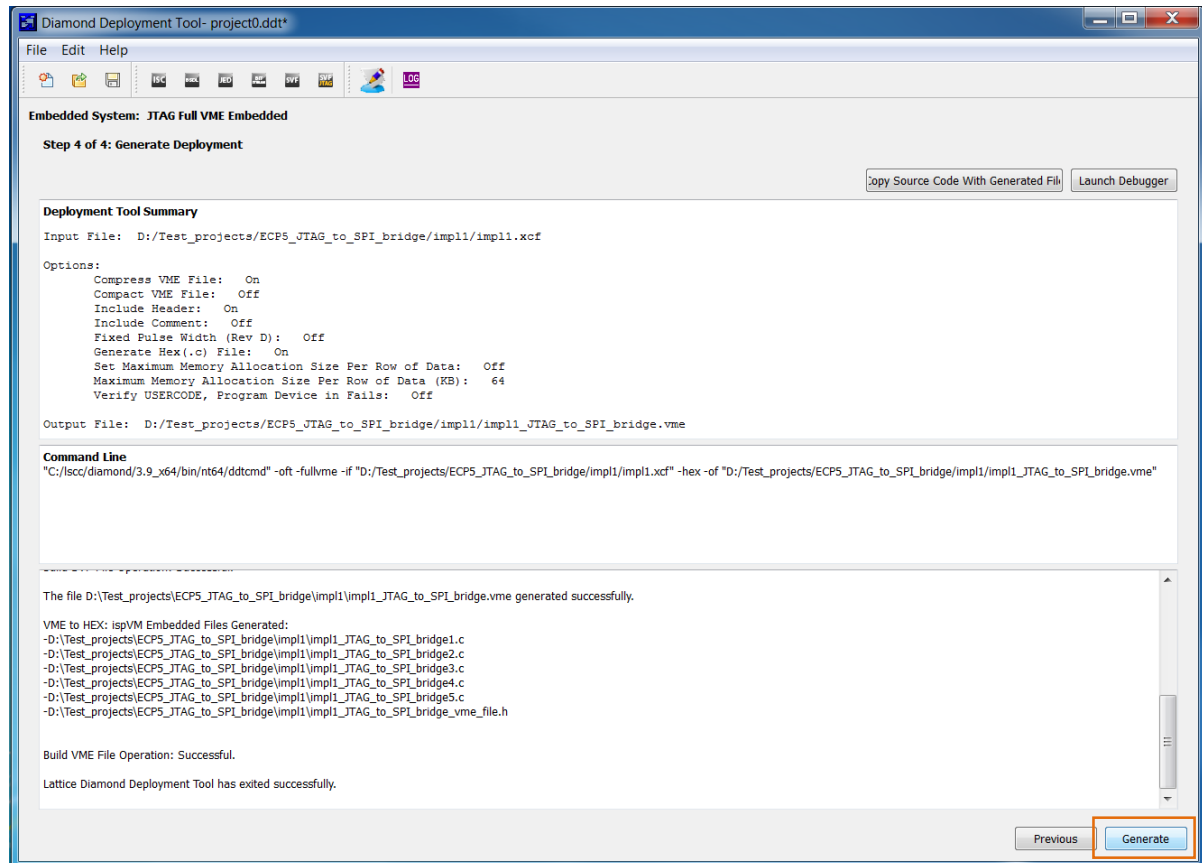


Figure 7.5. Step 4 of 4: Generate Deployment

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Date	Version	Change Summary
October 2017	1.0	Initial release



7th Floor, 111 SW 5th Avenue
Portland, OR 97204, USA
T 503.268.8000
www.latticesemi.com