Programming and Debugging Mixed Platform Devices

Application Note

FPGA-AN-02061-1.0

March 2023

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
SVF	Single vector format programming file
XCF	Chain configuration project file for Programmer
JTAG	Joint Test Action Group

1. Introduction

The purpose of this document is to cover the general software usage flow for programming and debugging a JTAG chain which consists of Lattice Radiant™ and Diamond® devices. Considering both Lattice Diamond and Radiant support different devices, the process for programming and debugging a JTAG chain which consists of mixed devices from both software tools requires some additional setup. At a high level, this process requires user to generate a .SVF serial vector format file to program the devices in the JTAG chain that are not native to the version of the programmer being used. For more detailed information about these steps for programming and debugging, refer to the following two sections of the document.

2. Programming Mixed Platform Devices in a JTAG Chain

2.1. Generating a .SVF File

The first step in programming and debugging a JTAG chain which consists of mixed platform devices from Radiant and Diamond is to generate a .SVF file. This file can be used to program the device that is not native to the version of Programmer being used. Generally, it is recommended that a .SVF programming file is generated for the device in the JTAG chain that is native to Lattice Radiant.

1. Load Programmer from Lattice Radiant:
   a. Select **Tools > Programmer** from the Radiant toolbar.
   b. The standalone Programmer tool can also be used to generate a .SVF file.
2. Select **Create a new blank project** from the popup window that appears, then click **OK.**

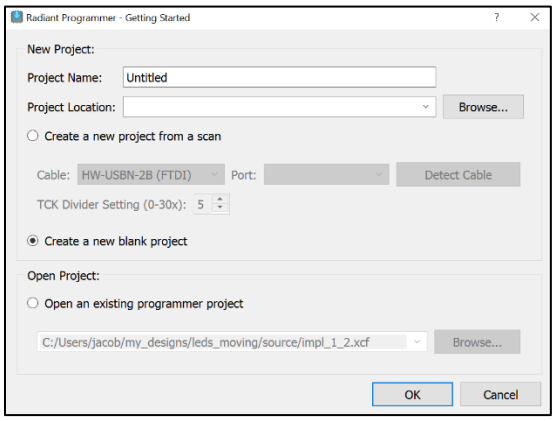

Figure 2.1. New Project Window in Radiant Programmer

3. Select **Tools > Deployment Tool** from the Programmer toolbar.

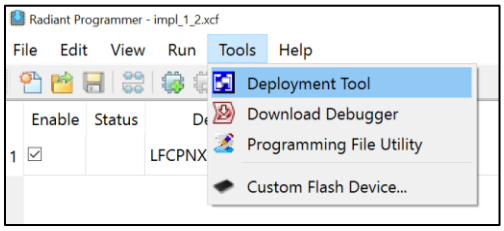

Figure 2.2. Opening Deployment Tool in Radiant Programmer

4. Select **Create New Deployment**.
   - **Function Type: Tester**
   - **Output file type: SVF – Single Device**

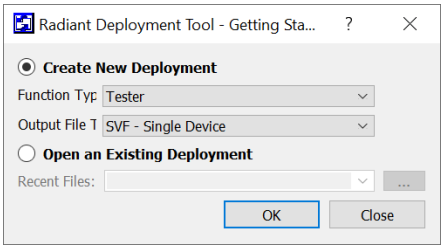

Figure 2.3. Creating New Deployment Tool Project in Radiant Programmer

5. Click **OK.**
6. Using the **Name** column, select the programming file to generate a .SVF. Click **Next.**

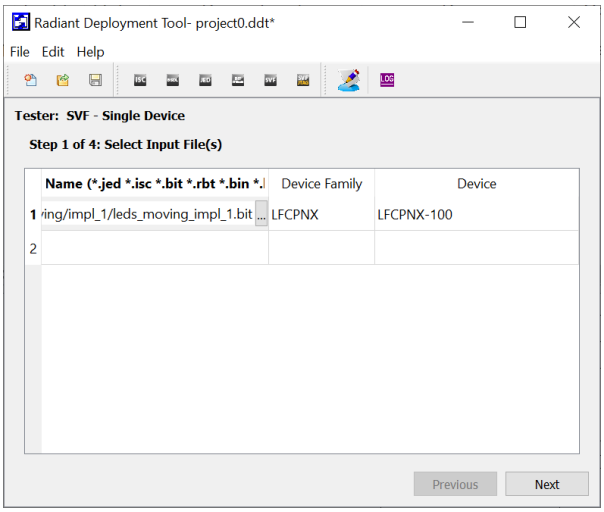

Figure 2.4. Selecting Input File for .SVF File Generation

7. Select additional **SVF Options**. Click **Next**.
   a. **Target Memory** and **Access Mode** should match to the memory of the target device being programmed.
   b. **Operation** must be set to **Erase, Program, Verify**.

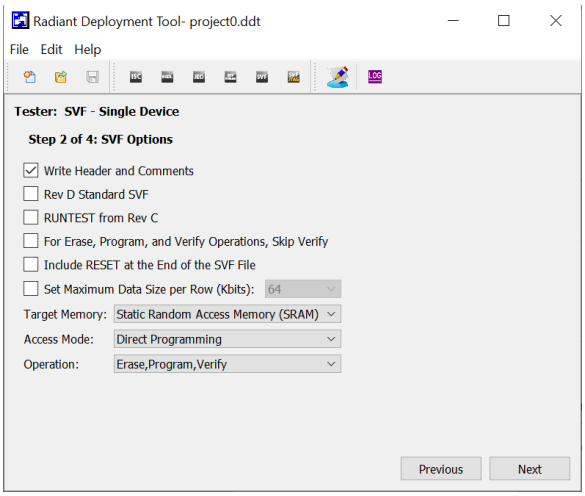

Figure 2.5. Option Selection for .SVF File Generation

8. Select the .SVF file from the directory. Click **Next**.

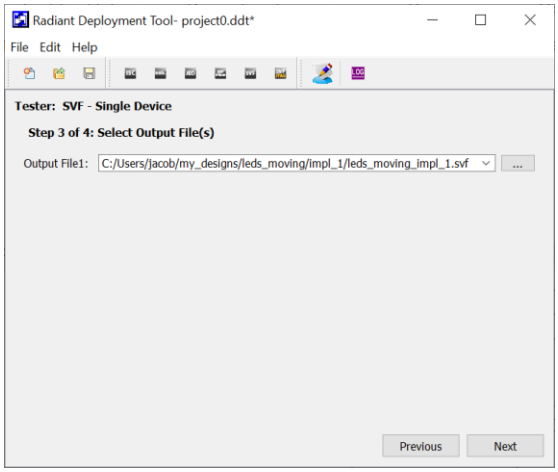

Figure 2.6. Output Directory for .SVF File Generation

9. Review the summary of selections and click **Generate**. This file can be used later in Device Programming section.

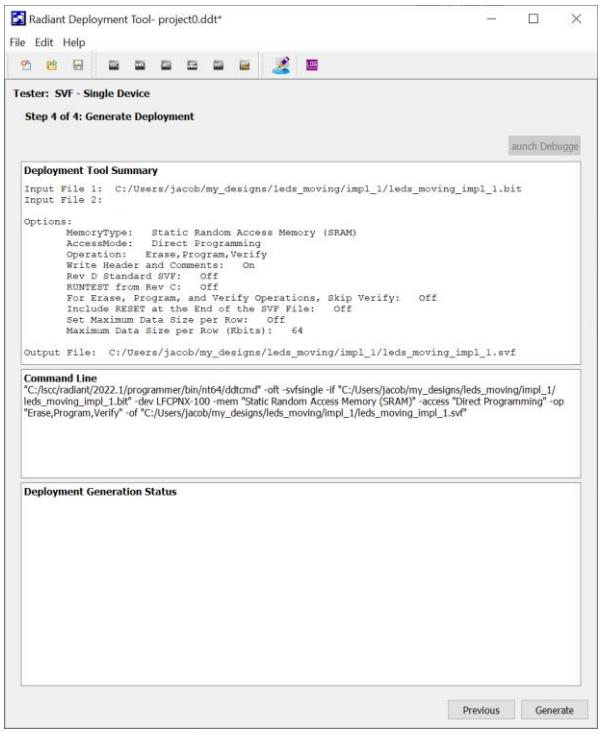

Figure 2.7. Deployment Tool Summary Window for .SVF File Generation

2.2. Configuring the JTAG Chain

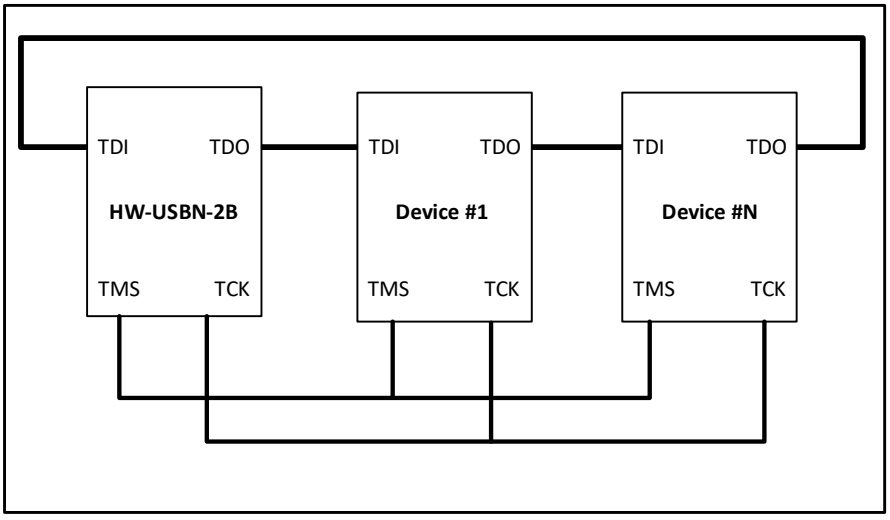

Figure 2.8. General JTAG Pin Connections for Mixed Platform Device JTAG Chain

As shown in Figure 2.8, the general JTAG connections for each device in a mixed device JTAG chain can be simplified to a few connections. The most important thing to note is that the TDO pin of the HW-USBN-2B programming cable should be connected to the TDI pin of the first device in the JTAG chain. From then on, each respective device in the JTAG chain must have its TDI pin connected to the TDO pin of the previous device in the JTAG chain. The TDO port of the last device in the JTAG chain should be connected to the TDI pin of the HW-USBN-2B programming cable. Aside from that, the TMS, TCK, GND, and 3V3 pins for each device in the chain should be common with each other.

2.3. Device Programming

1. Load Programmer from Lattice Diamond.
   a. Select **Tools > Programmer** from the Diamond toolbar.
   b. The standalone Programmer tool can also be used to program the devices in the JTAG chain.
2. Select **Create a new blank project** from the popup window that appears. Click **OK.**
3. Add the devices in the JTAG chain using the add device icon from the Programmer toolbar.
   a. To add a device, use the **Device Family** and **Device** fields to select the correct device.
   - For devices native to Radiant, select **Generic JTAG Device** and **JTAG-SVF**.
   - For devices native to Diamond, select the correct device and part number.

   b. The order devices are listed here should correspond to each respective device's order in the JTAG chain.
4. Select the programming file for devices native to Lattice Diamond.
5. Use the .SVF file generated from Generating a .SVF File section to program the devices from Lattice Radiant in the JTAG chain.

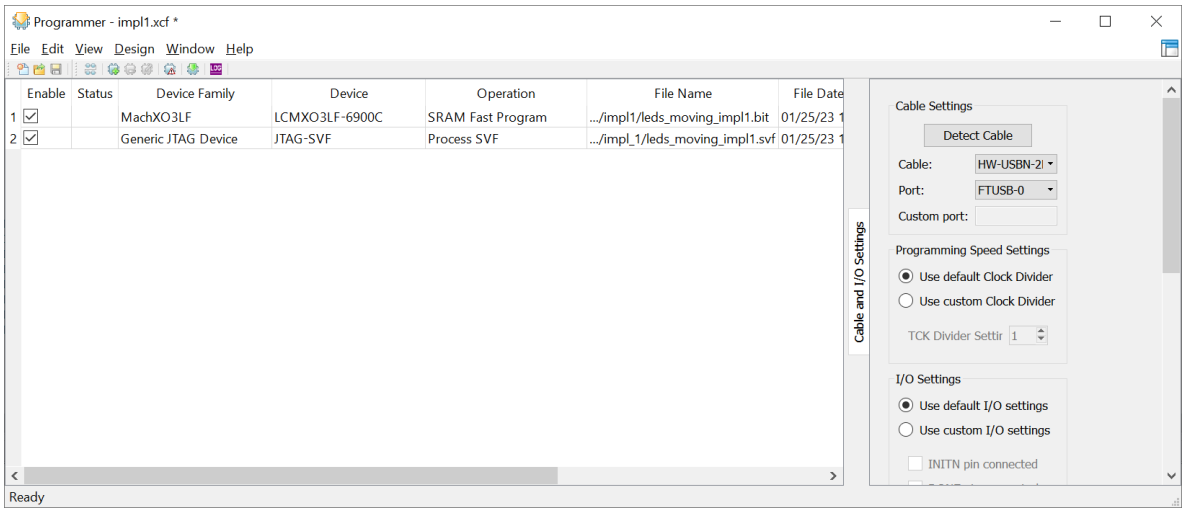

Figure 2.9. JTAG Chain Configuration in Diamond Programmer

6. Click the **Program** icon from the toolbar to program each device in the JTAG chain.
   a. If programming fails, try to raise the TCK divider setting to slow down the JTAG clock frequency.

3. Debugging Mixed Platform Devices in a JTAG Chain

To debug a device within a mixed platform JTAG chain, a new programming file must be generated for the target device to be debugged. The main thing to note for debugging devices in a mixed JTAG chain, is that only a single device can be debugged at a time. In addition, another important thing to note is that each respective device can only be debugged using the version of Reveal native to that device's software tool. For example, the MachXO3D™ device from Lattice Diamond can only be used with Diamond's version of Reveal, and cannot be used within Radiant, although the general debugging steps are the same for both software tools.

Although the screenshots from the example below are targeting Radiant's version of Reveal and Programmer, the steps for debugging a device native to Diamond in a mixed platform JTAG chain are the same. For more information on generating and configuring debug cores using Reveal Analyzer and Controller, refer to the Lattice semiconductor website.

1. Create, configure, and insert Reveal debug logic to a project using Reveal Inserter.
   - For more information about this flow, refer to Radiant and Diamond's web help documentation.
2. Generate a new bitstream containing the Reveal debug core logic.
   - Each time Reveal Inserter is used to modify a debug core, a new bitstream must be generated.
   - Generate a new bitstream only to debug the target device.
3. Load Programmer and select **Create a new blank project** from the popup window that appears, then click **OK**.
   - It is recommended that a new .XCF chain configuration project is generated for debugging. This prevents user from reconfigure the devices in the JTAG chain each time they want to debug a different device.
4. Configure the **Device**, **Device Family**, and **Operation** fields according to the position of each respective device in the JTAG chain.
   - The target device to be debugged should have the correct family and part number selected.
     - The programming file for this device should be the updated Reveal bitstream.
   - All other devices in the JTAG chain should be set to Generic JTAG Device and JTAG-NOP, with their operation set to Bypass.

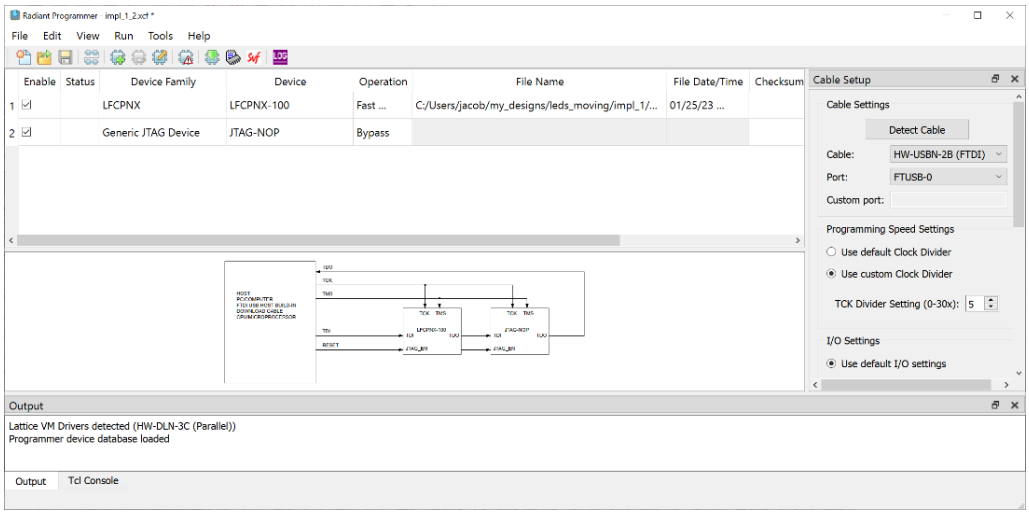

Figure 3.1. Programmer Project Configuration to Debug a Device in a Mixed Platform JTAG Chain

5. Click the **Program** icon from the toolbar to program each device in the JTAG chain. If programming fails, try to raise the TCK divider setting to slow down the JTAG clock frequency.
6. Load **Reveal Analyzer/Controller** and create a new .RVA Reveal project.
   a. Click the **Detect** button to detect the USB port the HW-USBN-2B cable is connected to.
   b. Select the **Browse…** button to select the Programmer project used to program this JTAG chain.
   c. Click **Scan** to detect the devices in the JTAG chain, then select the correct device containing the Reveal debug logic from the drop-down menu.

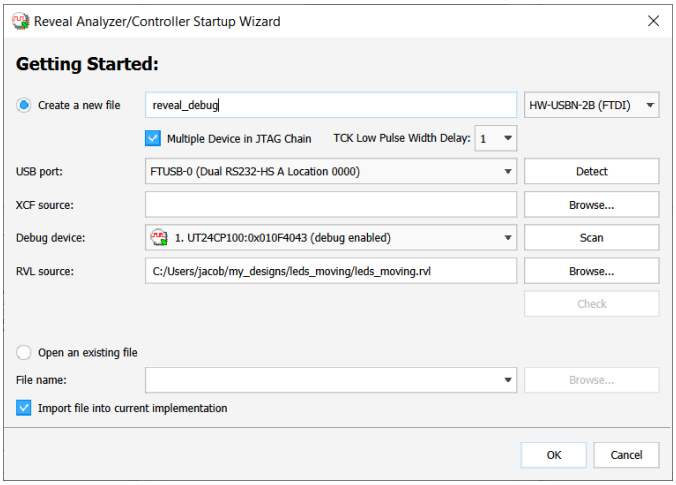

Figure 3.2. Reveal Analyzer/Controller Project Configuration Window Setup

7. Click **OK** to finish creating the new Reveal debug project**.**
8. Begin debugging like any other Reveal project**.**
   - This process can vary depending on the debug logic inserted.
   - For Reveal Analyzer debug cores, click the icon to begin scanning for trigger conditions.

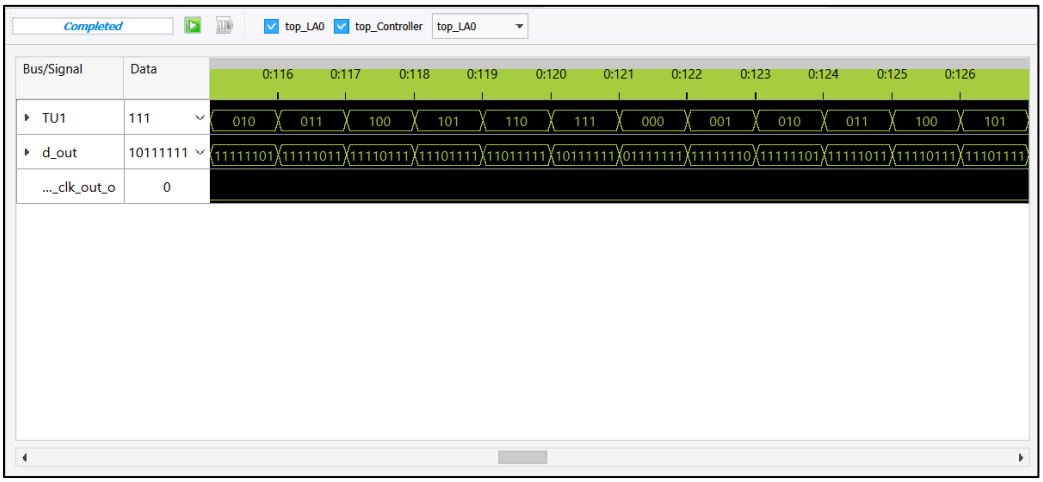

Figure 3.3. Captured Signals from the Device in a JTAG Chain with Mixed Platform Devices

Technical Support Assistance

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

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, March 2023

Section	Change Summary
All	Initial release.

www.latticesemi.com