Mach-NX Dual Boot Feature Usage Guide

Technical Note

FPGA-TN-02229-1.0

August 2021

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 and with all faults, and all risk associated with such information is entirely with 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

Glossary

A glossary of terms used in this document.

Acronym	Definition
Alternative Boot	After the FPGA device has been configured, this pattern is loaded when the PROGRAMN pin is toggled or the Refresh instruction is issued. Up to four Alternative Boot patterns are possible.
Binary Hex Data File (.bin File)	The data image of the Hex data file in binary format. All Hex data files are converted into this format prior to consumption. This type of file is not printable.
Bitstream Data File (.bit File)	The configuration data file, for a single FPGA device, in the format that can be loaded directly into the FPGA device to configure the SRAM cells. The file is expressed in binary Hex format. The file is not printable.
Configure	Write the pattern into the SRAM fuses of the FPGA device and wake up.
Dual Boot	The device has two patterns, a Primary pattern and a Golden pattern, to choose to load.
Flash Lock	The feature provides protection to the Flash fuses against accidental erase or corruption. Most of the SPI Flash devices support Soft Lock. Lock choices include: • Whole device • Bottom half • Bottom quarter • Last sector Details can be found in the SPI Flash device data sheet.
Golden Boot	The guaranteed good pattern loaded into the FPGA device when booting failure occurs. It is also known as the root boot. Only one Golden Boot pattern is allowed.
Hex Data File (.exo, .mcs, .xtek Files)	The data record files that are in the format commonly known as Intel Hex, Motorola Hex or Extended Tektronix Hex. They are also known as addressed record files. The advantages include its small size and it is printable, and thus good for record keeping. This type of file is not directly consumable by the utilities supporting it.
Multiple Boot	The device has more than two patterns, a Primary pattern, a Golden pattern and some Alternative patterns, to choose to load.
Primary Boot	Upon power cycling, the FPGA device will load this pattern in first. Only one Primary pattern is allowed.
Program	Writes into the selected Flash cells state a logical zero (0) (close fuse).
Refresh	The action loads the pattern from a non-volatile source to configure the FPGA device.
SPI	Stands for the Serial Peripheral Interface defined originally by Motorola.
Sector (Block)	The smallest number of bytes of Flash fuses can be erased at the same time by the erase command.

1. Introduction

In computing, booting is starting up a computer or computer appliance until it can be used. The process involves a chain of stages. At each stage, a smaller, simpler program loads and then executes the larger and more complicated program of the next stage. When a computer is turned off, its software including operating systems, application code, and data remains stored in non-volatile memory (NVM) from which the operating system programs and the data can be loaded into the RAM in subsequent booting. Those applications stored in NVM need to be read or updated from time to time.

One of the biggest risks in the field upgrade applications is disruption during the field upgrade process, for example:

- Power disruption
- Data file corruption
- Communications disruption

Even if the system does not require a field upgrade, the pattern corruption can still occur due to the following problems caused by data retention issue of flash devices:

- Read fatigue
- Charge loss

To eliminate the risk completely, Lattice Semiconductor provides a Dual Boot feature that switches to load from the second known good pattern, Golden pattern, when the first applied pattern, Primary pattern, is corrupted. This feature enhances the reliability of a field upgradeable system, as the Golden pattern is less affected by these problems since most of the time it is in a dormant state.

The Lattice Semiconductor Mach-NX devices provide a unique on-chip Dual Boot feature to optionally boot from two sectors of the internal flash, CFG0 and CFG1.

2. Mach-NX Dual Boot Mode

The Mach-NX family supports two types of on-chip Dual Boot configuration modes, golden image dual configuration and version-based dual configuration. In golden image dual configuration mode, if the Primary bitstream becomes corrupted while being loaded into the SRAM, the device can be automatically booted from the Golden bitstream. In version-based dual configuration, it can be booted with the LATEST or FORMER bitstream first and designating the other bitstream as the Golden pattern.

3. Configuring and Programming for Dual Boot

3.1. Configuring for Dual Boot

1. In Lattice Diamond Software, the **Spreadsheet View** allows you to select the image for PRIMARY_BOOT and the image for SECONDARY_BOOT. Open **Spreadsheet View** by clicking on the **Spreadsheet View** icon ( ) or go to **Tools** > **Spreadsheet View**, as shown in Figure 3.1.
2. Go to the **Global Preferences** tab and the options are listed under sysConfig in Figure 3.2. Four possible dual boot configurations are available for Mach-NX devices:
   - IMAGE0 – IMAGE1
   - IMAGE1 – IMAGE0
   - LATEST – FORMER
   - FORMER – LATEST

   The boot mode configuration is assigned in Diamond Software through **Spreadsheet View** – **Global Preferences** tab. Under **sysConfig,** there are options to select the source for the PRIMARY_BOOT and SECONDARY_BOOT. Four possible options are available from the PRIMARY_BOOT preference:
   - IMAGE_0 – Designates the configuration image in CFG0 for PRIMARY_BOOT.
   - IMAGE_1 – Designates the configuration image in CFG1 for PRIMARY_BOOT.
   - LATEST – Designates either the CFG0 or CFG1 image, depending on which one has the later configuration image or the larger version number as that of the PRIMARY_BOOT.
   - FORMER – Designates either the CFG0 or CF1 image, depending on which one has the former configuration image or the smaller version number as that of the PRIMARY_BOOT.

   Five possible options are available from the SECONDARY_BOOT preference:
   - IMAGE_0 – Designates the configuration image in CFG0 for SECONDARY_BOOT.
   - IMAGE_1 – Designates the configuration image in CFG1 for SECONDARY_BOOT.
   - LATEST – Designates either the CFG0 or CFG1 image, depending on which one has the later configuration image or the larger version number as that of the SECONDARY_BOOT.
   - FORMER – Designates either the CFG0 or CF1 image, depending on which one has the former configuration image or the smaller version number as that of the SECONDARY_BOOT.
   - NONE – This option is selected when dual boot is not used.
3. Select **JEDEC File** from the Process Pane to generate the .jed programming file with the design settings.

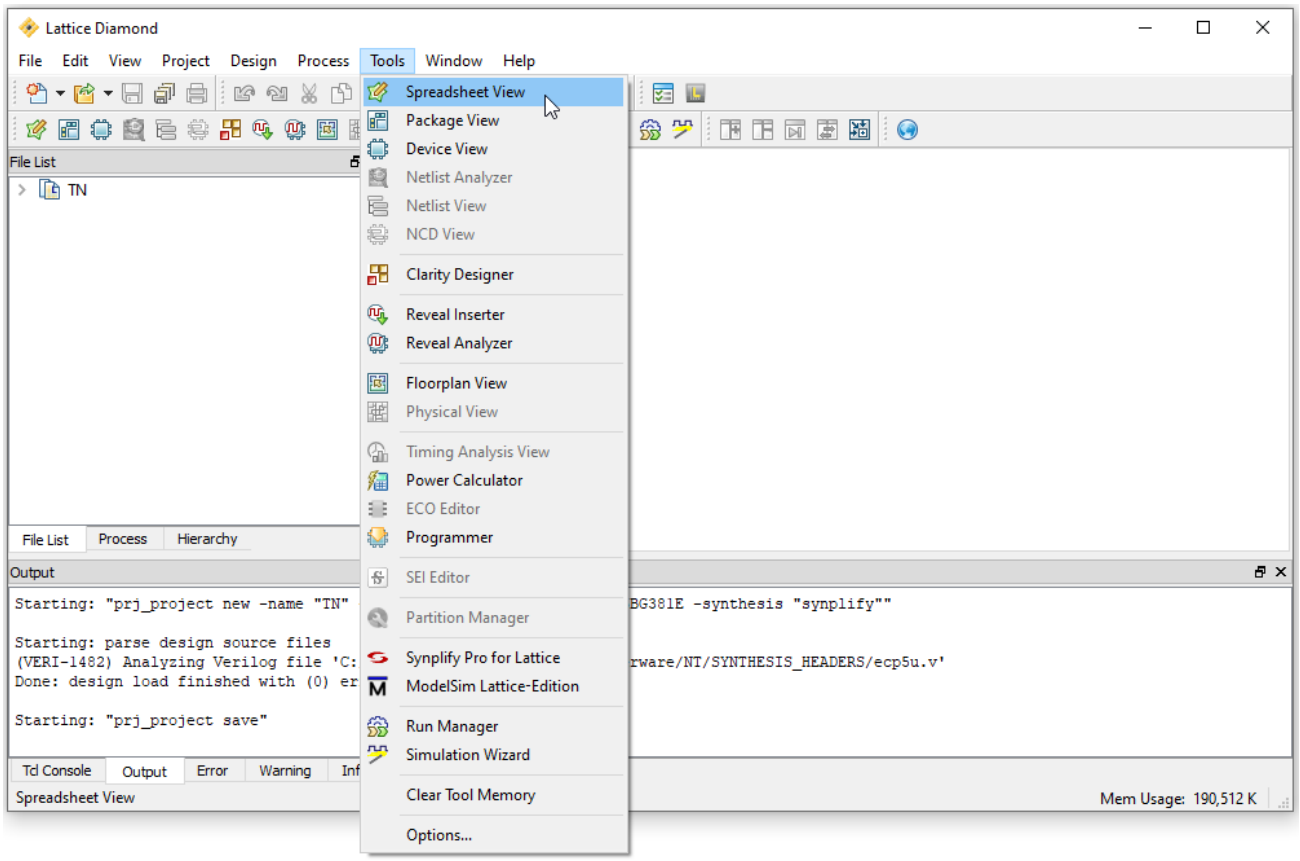

Figure 3.1. Lattice Diamond Software – Spreadsheet View

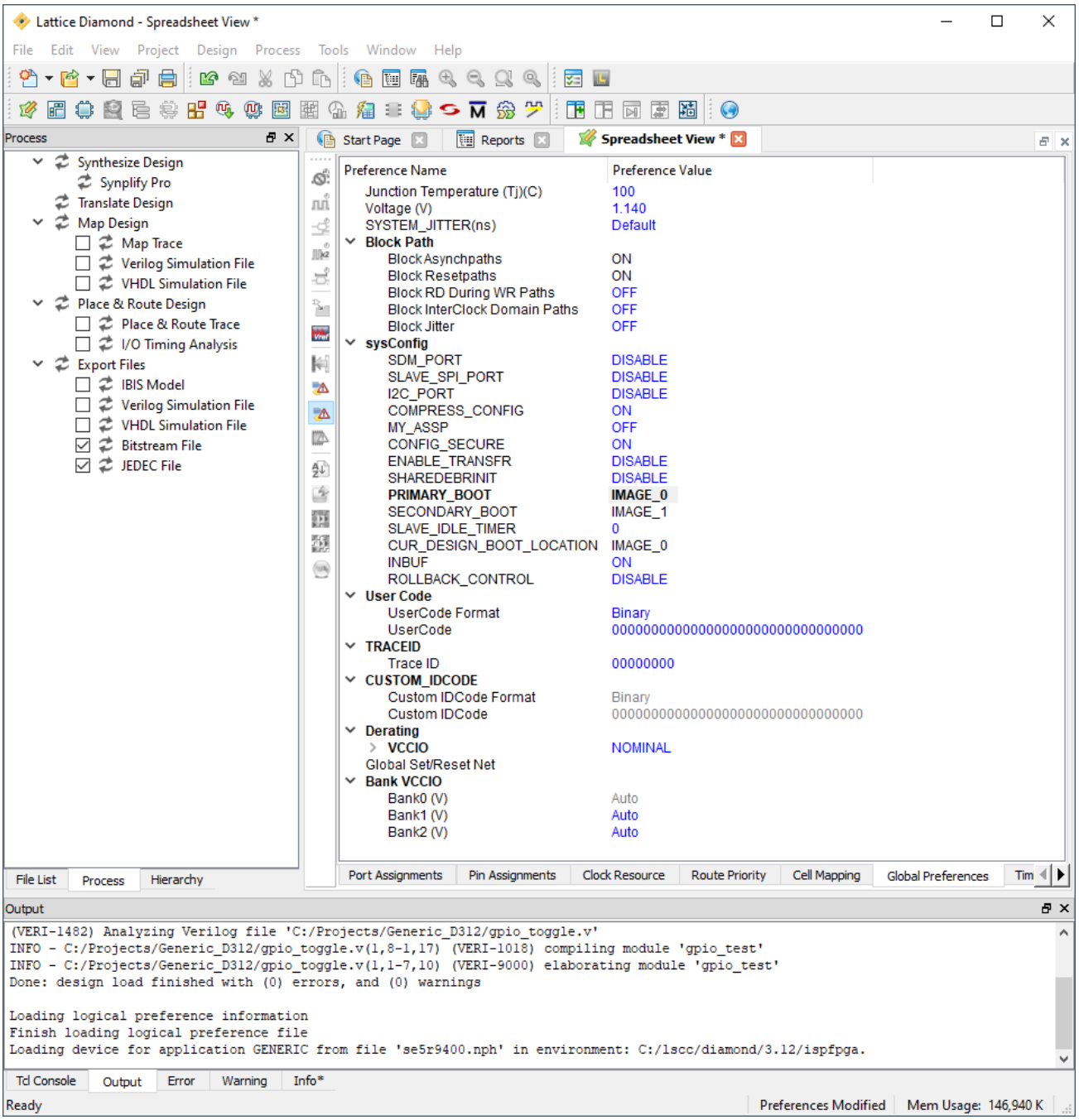

Figure 3.2. Lattice Diamond Software – Spreadsheet View – Global Preferences Tab

3.2. Programming for Dual Boot

1. Invoke Diamond Programmer from the Diamond Software window by going to **Tools** > **Programmer**, as shown in Figure 3.3, or clicking the **Programmer** icon ( ) from the Diamond Software toolbar. Diamond Programmer can also be selected outside of Diamond Software by going to Windows Start Menu. Select it from the **Lattice Diamond 3.12** folder.
2. In the Diamond Programmer graphic user interface (GUI), double-click in the **Operation** field (Figure 3.4).

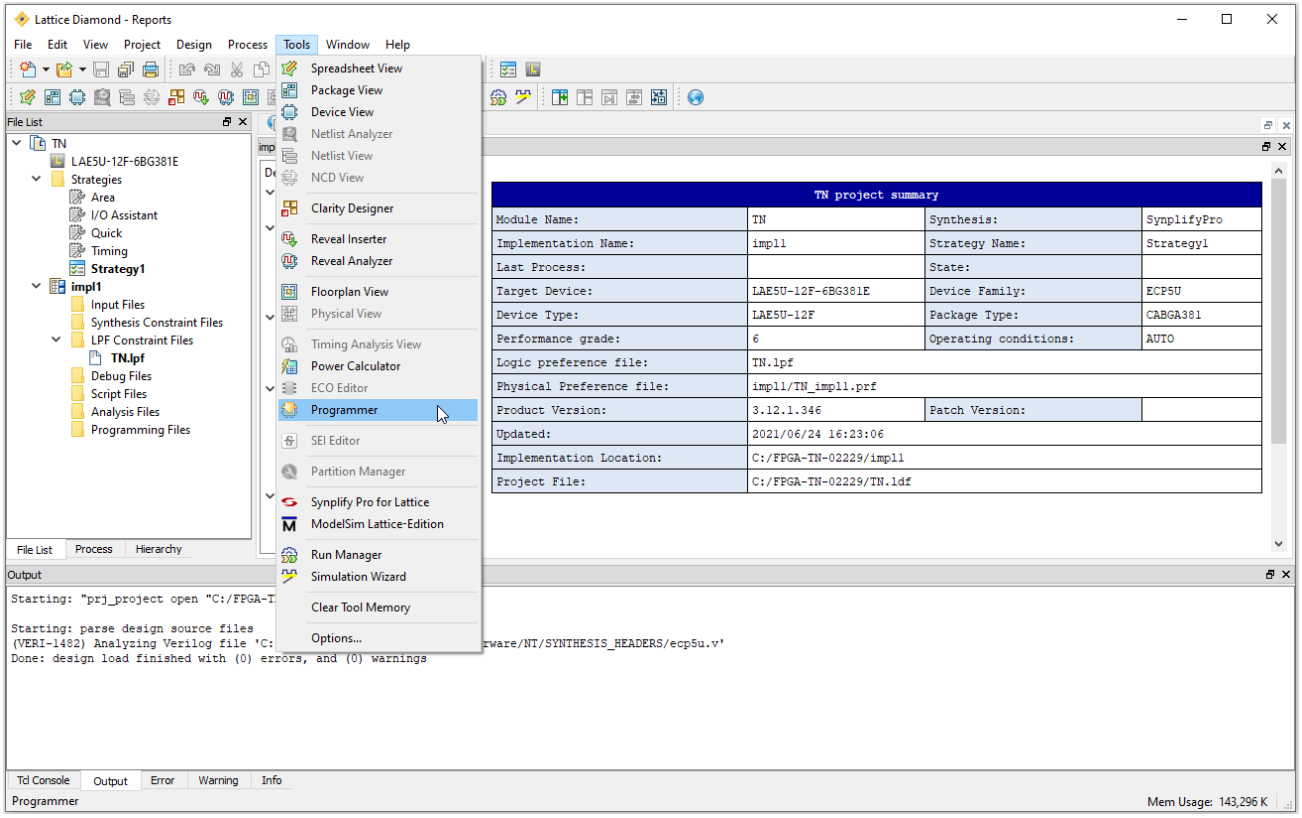

Figure 3.3. Lattice Diamond Software – Programmer

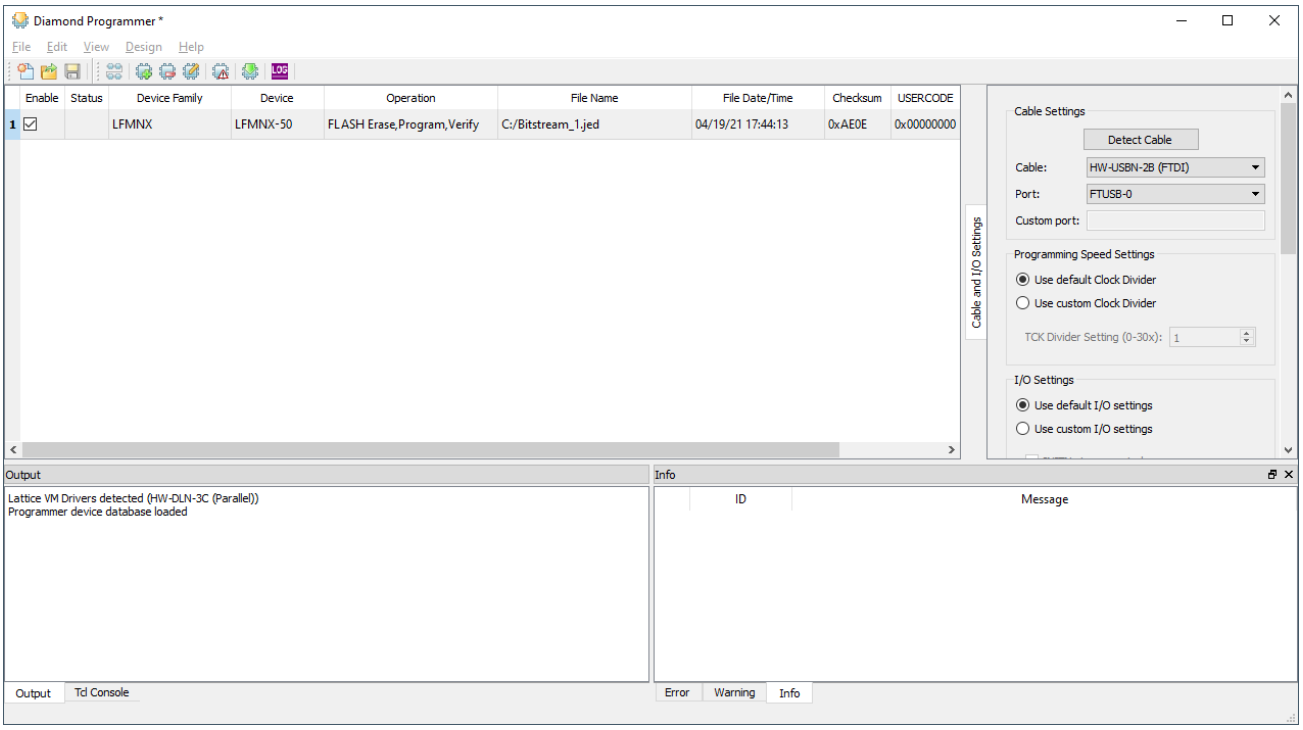

Figure 3.4. Diamond Programmer Window

3. The **Device Properties** window opens (Figure 3.5).
    a. Select **Flash Programming Mode** for **Access mode**.
    b. Select **JTAG Interface** for **Port Interface**.
    c. Select **FLASH Erase,Program,Verify** for **Operation**.
    d. Check the **CFG0 Programming Options** box. Select the configuration file in the **Programming file** field by clicking on the **...** button. This loads the configuration file into the CFG0 portion of the in-chip Flash memory space.
    e. Check the **CFG1 Programming Options** box. Select the configuration file in the **Programming file** field by clicking on the **...** button. This loads the configuration file into the CFG1 portion of the in-chip Flash memory space.
    f. Click **OK**.

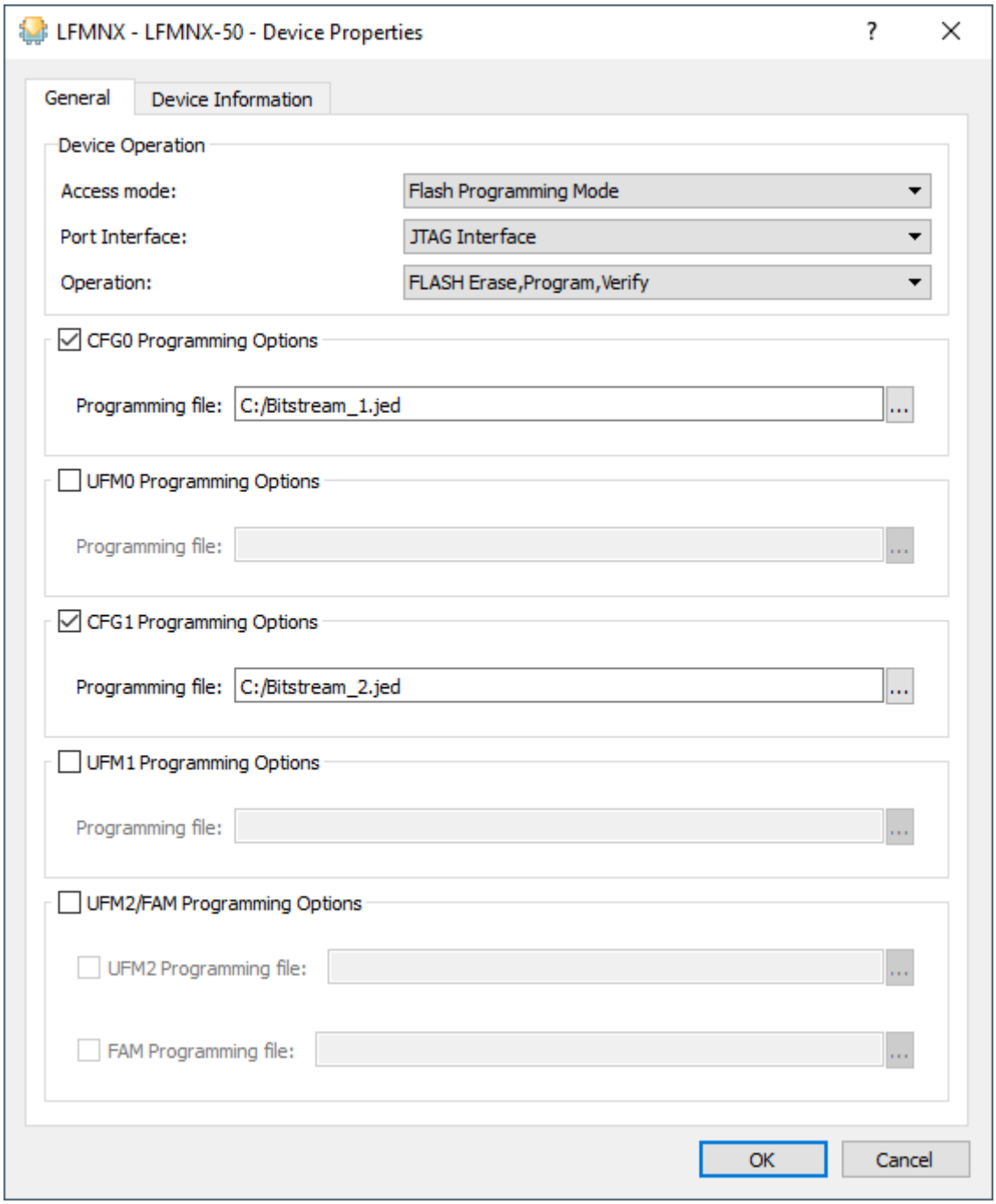

Figure 3.5. Device Properties Window

4. Click the **Program** icon ( ) from the toolbar, or select **Design** > **Program** (Figure 3.6) to program the Mach-NX device.

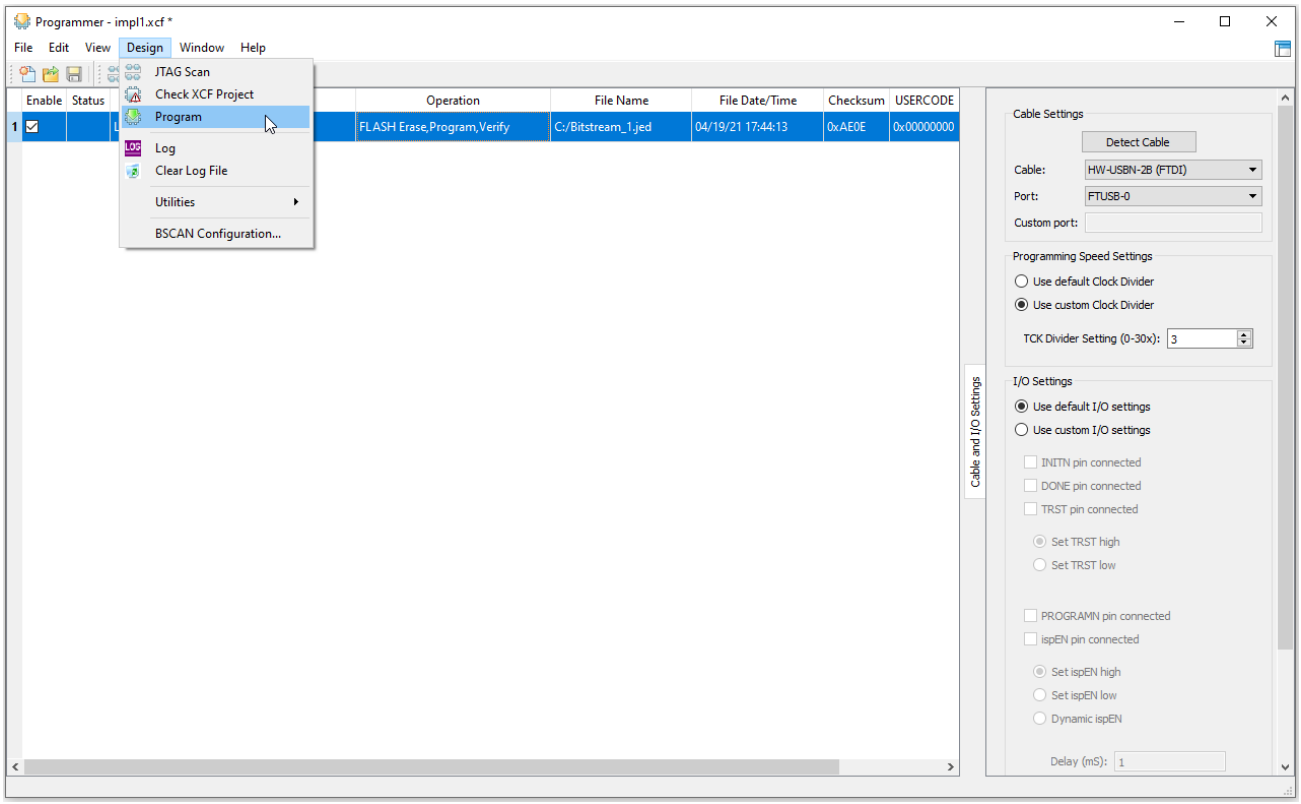

Figure 3.6. Diamond Programmer Window

Technical Support Assistance

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

Revision History

Revision 1.0, August 2021

Section	Change Summary
All	Production release.

www.latticesemi.com