Multi-Boot User Guide for Nexus Platform

Technical Note

FPGA-TN-02145-1.5

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

Tables

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.
EBR	Embedded Block RAM
FD-SOI (Fully Depleted Silicon On Insulator)	A process that uses an ultra-thin buried oxide layer.
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.
LRAM	Large RAM
Multi-Boot 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 loads this pattern in first. Only one Primary pattern is allowed.
Program	Writes into the selected Flash cells state a logical zero (0) (close fuse).
RAM	Random Access Memory
Refresh	The action loads the pattern from a non-volatile source to configure the FPGA device.
Sector (Block)	The smallest number of bytes of Flash fuses can be erased at the same time by the erase command.
SPI	Stands for the Serial Peripheral Interface defined originally by Motorola.
SRAM	Static Random Access Memory

1. Introduction

CrossLink™-NX, Certus™-NX, CertusPro™-NX, MachXO5™-NX families of low-power FPGAs can be used in a wide range of applications and are optimized for the bridging and processing needs in the Embedded Vision space. It is built on Lattice Nexus FPGA platform, using low-power 28-nm FD-SOI technology. CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX families support various booting options for loading the configuration SRAM from a non-volatile memory for configuration flexibility and fail-safe configuration. CrossLink-NX, Certus-NX and CertusPro-NX families use an external memory while MachXO5-NX families have an internal memory for storage of configuration bitstreams. The CrossLink-NX family includes LIFCL-17, LIFCL-33, and LIFCL-40 parts. The Certus family includes LFD2NX-17 and LFD2NX-40 parts. The CertusPro-NX family includes LFCPNX-100 parts. The MachXO5-NX family includes LFMXO5-25, LFMXO5-55T, and LFMXO5-100T parts.

The CrossLink-NX, Certus-NX, CertusPro-NX, and MachXO5-NX families support various configuration boot modes to mitigate risk during the field upgrade process and to allow flexibility of executing different patterns. Field upgrade disruptions may occur due to power disruption, communication interruption or bitstream pattern corruption. The CrossLink-NX, Certus-NX, CertusPro-NX, and MachXO5-NX families support the following boot modes:

- Dual Boot mode – Switches to load from the second known good (Golden) pattern when the first pattern becomes corrupted.
- Ping-Pong Boot mode – Switches between two bitstream patterns based on your choice. If the system fails to boot from one of the bitstreams, it automatically boots from the second bitstream.
- Multi-Boot mode – Allows the system to dynamically switch between two to five bitstream patterns while still being protected with a Golden (sixth) pattern.

The CrossLink-NX, Certus-NX, CertusPro-NX, MachXO5-NX families support these boot modes by combining all the bitstream patterns into a single boot image and storing it in a single external SPI Flash device (internal flash for MachXO5-NX family). This solution decreases cost, reduces board space and simplifies field grades.

2. Resources

The CrossLink-NX, Certus-NX, and CertusPro-NX families are SRAM-based FPGAs. The volatile SRAM configuration memory must be loaded from an external non-volatile memory that can store all of the configuration data. The size of the configuration data is based on the amount of logic available in the FPGA, number of pre-initialized Embedded Block RAM (EBR) components and number of pre-initialized Large RAM (LRAM) Block components. A CrossLink-NX /Certus-NX/CertusPro-NX/MachXO5-NX design using the largest device, with every EBR and LRAM pre-initialized with unique data values and generated without compression requires the largest amount of storage. The minimum SPI Flash densities or embedded flash required to support the different configuration boot modes are listed in Table 2.1, Table 2.2, and Table 2.3.

Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode

Device	Configuration	Uncompressed[1]	SPI Mode
		Single Bitstream Size (Mb)	Minimum SPI Flash Size (Mb)
LIFCL-17 LFD2NX-17	No LRAM, No EBR	2.817	4
	No LRAM, MAX EBR	3.273	4
	MAX LRAM, No EBR	5.517	8
	MAX LRAM, MAX EBR	5.873	8
LIFCL-40 LFD2NX-40	No LRAM, No EBR	6.232	8
	No LRAM, MAX EBR	7.758	8
	MAX LRAM, No EBR	7.281	8
	MAX LRAM, MAX EBR	8.807	16
LFCPNX-100	No LRAM, No EBR	14.310	16
	No LRAM, MAX EBR	17.966	32
	MAX LRAM, No EBR	17.810	32
	MAX LRAM, MAX EBR	21.466	32
LFMXO5-25 LFMXO5-55T LFMXO5-100T	No LRAM, No EBR	4.451	NA[2]
	No LRAM, MAX EBR	5.077	NA[2]
	MAX LRAM, No EBR	6.028	NA[2]
	MAX LRAM, MAX EBR	6.653	NA[2]
LIFCL-33	No LRAM, No EBR	4.453	8
	No LRAM, MAX EBR	5.967	8
	MAX LRAM, No EBR	7.150	8
	MAX LRAM, MAX EBR	8.667	16

Notes:

1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX family of devices support bitstream compression. Compression ratio depends on the bitstream. Therefore, Table 2.1 only provides uncompressed bitstream data.
2. MachXO5-NX family of devices boot from internal flash memory.

Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode/Ping-Pong Mode

Device	Configuration	Uncompressed[1]		SPI Mode
		Single Bitstream Size (Mb)	Two Bitstreams Size (Mb)	Minimum SPI Flash Size (Mb)
LIFCL-17 LFD2NX-17	No LRAM, No EBR	2.817	5.634	8
	No LRAM, MAX EBR	3.273	6.546	8
	MAX LRAM, No EBR	5.517	11.034	16
	MAX LRAM, MAX EBR	5.873	11.746	16
LIFCL-40 LFD2NX-40	No LRAM, No EBR	6.232	12.464	16
	No LRAM, MAX EBR	7.758	15.516	16
	MAX LRAM, No EBR	7.281	14.562	16
	MAX LRAM, MAX EBR	8.807	17.614	32
LFCPNX-100	No LRAM, No EBR	14.310	28.620	32
	No LRAM, MAX EBR	17.966	35.933	64
	MAX LRAM, No EBR	17.810	35.620	64
	MAX LRAM, MAX EBR	21.466	42.933	64
LFMXO5-25 LFMXO5-55T LFMXO5-100T	No LRAM, No EBR	4.451	8.902	NA[2]
	No LRAM, MAX EBR	5.077	10.154	NA[2]
	MAX LRAM, No EBR	6.028	12.056	NA[2]
	MAX LRAM, MAX EBR	6.653	13.306	NA[2]
LIFCL-33	No LRAM, No EBR	4.453	8.906	16
	No LRAM, MAX EBR	5.967	11.934	16
	MAX LRAM, No EBR	7.150	14.300	16
	MAX LRAM, MAX EBR	8.667	17.334	32

Notes:

1. CrossLink-NX/Certus-NX/CertusPro-NX/ MachXO5-NX family of devices support bitstream compression. Compression ratio depends on the bitstream. Therefore, Table 2.2 only provides uncompressed bitstream data.
2. MachXO5-NX family of devices boot from internal flash memory.

Table 2.3. Maximum Configuration Bitstream Size – Multi-Boot Mode

Device	Configuration	Uncompressed[1]	3 Bitstream Patterns[2]		4 Bitstream Patterns[2]		5 Bitstream Patterns[2]		6 Bitstream Patterns[2]	
		Single Bitstream Size (Mb)	Minimum SPI Flash Size (Mb)							
LIFCL-17 LFD2NX-17	No LRAM, No EBR	2.817	8.451	16	11.268	16	14.085	16	16.902	32
	No LRAM, MAX EBR	3.273	9.819	16	13.092	16	16.365	32	19.638	32
	MAX LRAM, No EBR	5.517	16.551	32	22.068	32	27.585	32	33.102	64
	MAX LRAM, MAX EBR	5.873	17.619	32	23.492	32	29.365	32	35.238	64
LIFCL-40 LFD2NX-40	No LRAM, No EBR	6.232	18.696	32	24.928	32	31.160	32	37.392	64
	No LRAM, MAX EBR	7.758	23.274	32	31.032	32	38.790	64	46.548	64
	MAX LRAM, No EBR	7.281	21.843	32	29.124	32	36.405	64	43.686	64
	MAX LRAM, MAX EBR	8.807	26.421	32	35.228	64	44.035	64	52.842	64
LFCPNX-100	No LRAM, No EBR	14.310	42.930	64	57.240	64	71.550	128	85.861	128
	No LRAM, MAX EBR	17.966	53.899	64	71.865	128	89.832	128	107.798	128
	MAX LRAM, No EBR	17.810	53.430	64	71.240	128	89.050	128	106.861	128
	MAX LRAM, MAX EBR	21.466	64.399	128	85.865	128	107.332	128	128.798	256
LFMXO5-25 LFMXO5-55T LFMXO5-100T	No LRAM, No EBR	4.451	13.353	NA[3]	NA[4]	NA	NA[4]	NA	NA[4]	NA
	No LRAM, MAX EBR	5.077	15.231	NA[3]	NA[4]	NA	NA[4]	NA	NA[4]	NA
	MAX LRAM, No EBR	6.028	18.084	NA[3]	NA[4]	NA	NA[4]	NA	NA[4]	NA
	MAX LRAM, MAX EBR	6.653	19.959	NA[3]	NA[4]	NA	NA[4]	NA	NA[4]	NA
LIFCL-33	No LRAM, No EBR	4.453	13.359	16	17.812	32	22.265	32	26.718	32
	No LRAM, MAX EBR	5.967	17.901	32	23.868	32	29.835	32	35.802	64
	MAX LRAM, No EBR	7.150	21.450	32	28.600	32	35.750	64	42.900	64
	MAX LRAM, MAX EBR	8.667	26.001	32	34.668	64	43.335	64	52.002	64

Notes:

1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX family of devices support bitstream compression. Compression ratio depends on the bitstream. Therefore, Table 2.3 only provides uncompressed bitstream data.
2. Includes Golden bitstream pattern.
3. MachXO5-NX family of devices boot from internal flash memory.
4. MachXO5-NX internal flash memory supports a maximum of three bitstreams.

3. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Mode

The CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot mode supports booting from two configuration patterns that reside in an external SPI Flash device (internal flash for MachXO5-NX family). One pattern is designated as the Primary pattern, and the second pattern is designated as the Golden pattern. When the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX part boots up, it attempts to boot from the Primary pattern. If loading of the Primary pattern fails, the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX part boots from the Golden pattern.

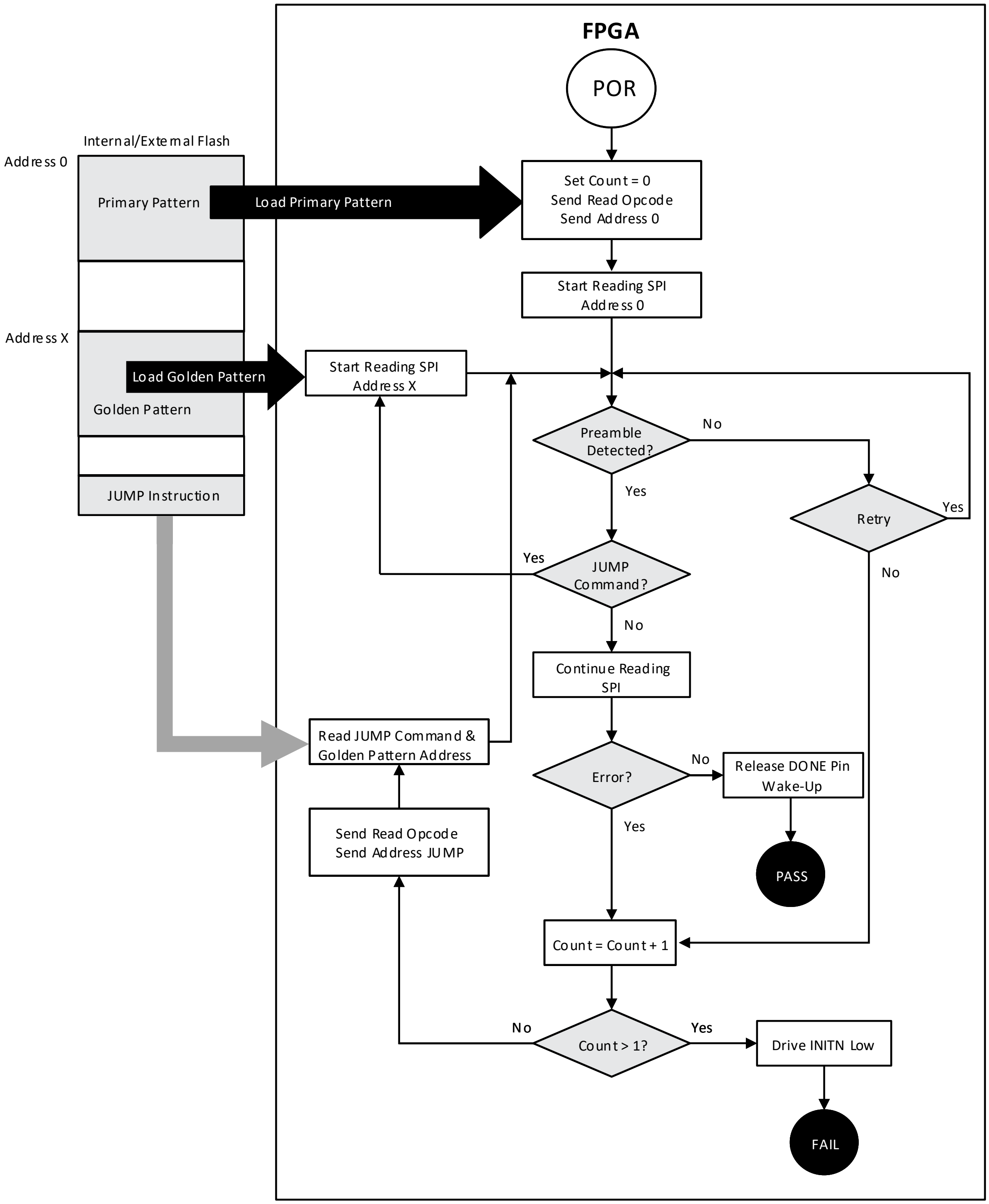


Figure 3.1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Flow Diagram

3.1. Description of the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Flow Diagram

This flow is triggered either by power cycle, the PROGRAMN pin being toggled, or by the REFRESH instruction being received.

When the Dual Boot mode is selected, in addition to the standard CRC check, a time-out check is performed while reading the Primary pattern, the Golden pattern, and the JUMP command.

- Time-Out Check – CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device searches for the preamble code 0xBDCD (0xBDB3 when Byte Wide Bit Mirror is enabled) from the Primary Pattern as part of the configuration protocol. The number of retries the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device attempts is defined in Control Register 1 [3:2] (Table 3.1).
- Data Corruption Check – After the detection of the preamble code, the CRC engine is turned on to detect whether or not the bitstream is corrupted. This determines whether the Flash device has a corrupted Primary pattern or Golden pattern due to Flash program disruption or data loss.

Table 3.1. Control Register 1 [3:2] – Master Preamble Timer Retry Value

	Bit 3	Bit 2
No retry	0	0
Retry 1 time	0	1
Retry 3 times	1	0
Reserved	1	1

If the Primary pattern fails one of the two checks above, the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device knows that the Primary pattern is not valid. It drives the INITN pin LOW briefly to indicate an error and resets the configuration engine. After clearing all the SRAM fuses, it drives the INITN pin HIGH, and reads the JUMP command that directs it to the location of the Golden pattern in the Flash.

If the JUMP command is corrupted, it also causes a configuration failure. It is important to note that a corrupted Golden pattern is not the only possible cause for Dual Boot configuration failure.

If the JUMP command is valid, the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device stops the SPI clock, drives the INITN pin LOW, resets the configuration engine, and performs a Clear All operation. The device then drives the INITN pin HIGH after the completion of the Clear All action, restarts the SPI clock, and reads the Golden pattern from the Flash address designated in the JUMP command.

CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device performs the same time-out check and CRC check when searching for the preamble code from the Golden pattern. If the Golden pattern is also corrupted, configuration fails. The CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device stops driving the SPI clock, and the INITN pin is driven LOW.

4. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Mode

The CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot mode supports booting from two configuration patterns that reside in an external SPI Flash device (internal flash for MachXO5-NX family). One pattern is designated as the Primary pattern and the second pattern is designated as the Secondary pattern. CrossLink-NX/Certus-NX/CertusPro-NX/ MachXO5-NX boots from the pattern assigned in the Jump table. The Jump table allows the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device to boot from either the Primary pattern or the Secondary pattern without changing the physical location of the patterns within the Flash. Only the Jump table needs to be updated to change the boot pattern. The other pattern, by default, becomes the Golden pattern.

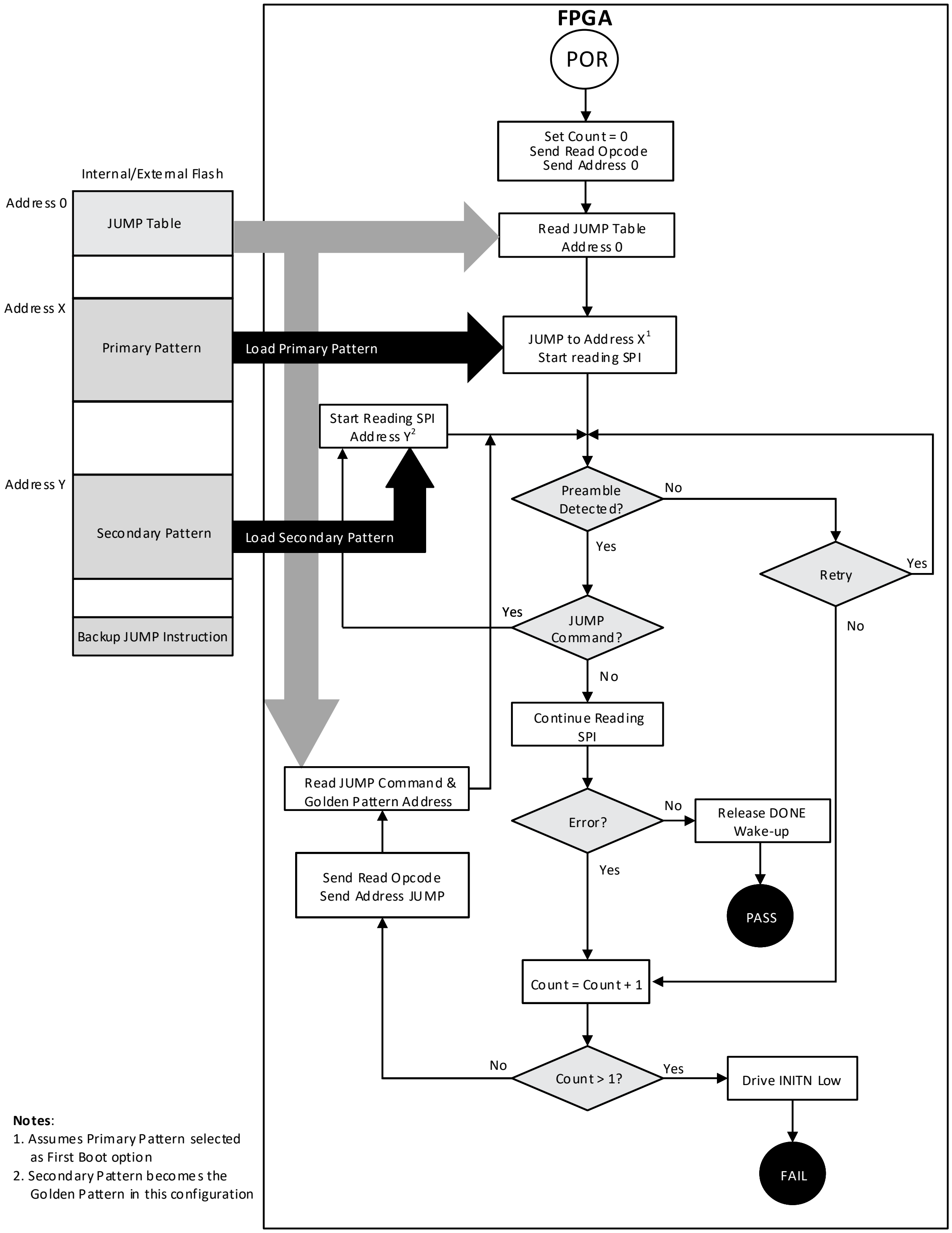


Figure 4.1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Flow Diagram

4.1. Description of the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Flow Diagram

This flow is triggered either by power cycle, the PROGRAMN pin being toggled, or by the REFRESH instruction being received.

When Ping-Pong Boot mode is selected, in addition to the standard CRC checking, a time-out check is performed while reading the Primary pattern, the Secondary pattern, and the JUMP command.

- Time-Out Check – CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device searches for the preamble code 0xBDCD (0xBDB3 when Byte Wide Bit Mirror is enabled) from the pattern designated as "First Boot" selection as part of the configuration protocol. The number of retries CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX attempts is defined in Control Register 1 [3:2] (Table 4.1).
- Data Corruption Check – After the detection of the preamble code, the CRC engine is turned on to detect whether the bitstream is corrupted. This will determine whether the Flash device has a corrupted Primary or Secondary Pattern due to Flash program disruption or data loss.

Table 4.1. Control Register 1 [3:2] – Master Preamble Timer Retry Value

	Bit 3	Bit 2
No retry	0	0
Retry 1 time	0	1
Retry 3 times	1	0
Reserved	1	1

If the "First Boot" pattern fails one of the two checks above, the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device knows that the pattern is not valid. It drives the INITN pin LOW briefly to indicate an error and resets the configuration engine. After clearing all the SRAM fuses, it drives the INITN pin HIGH, and reads the JUMP command that directs it to the location of the other pattern, acting as the Golden pattern, in the Flash.

If the JUMP command is corrupted, it also causes a configuration failure. It is important to note that a corrupted Golden pattern is not the only possible cause for Ping-ping mode configuration failure.

If the JUMP command is valid, the CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device stops the SPI clock, drives the INITN pin LOW, resets the configuration engine, and performs a Clear All operation. The device then drives the INITN pin HIGH after the completion of the Clear All action, restarts the SPI clock, and reads the Golden pattern from the Flash address designated in the JUMP command.

CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device performs the same time-out check and the CRC check when searching for the preamble code from the Golden Pattern. If the Golden Pattern is also corrupted, configuration fails, stops driving the SPI clock, and the INITN pin is driven LOW.

5. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Multi-Boot Mode

CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device Multi-Boot supports booting from up to six patterns that reside in an external SPI Flash device (up to three patterns for MachXO5-NX internal memory). The patterns include a Primary pattern, a Golden pattern, and up to four Alternate patterns, designated as Alternate pattern 1 to Alternate pattern 4.

CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device boots by loading the Primary pattern from the internal or external Flash, depending on the device family. If loading of the Primary pattern fails, CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX device attempts to load the Golden pattern. When a reprogramming of the bitstream is triggered through the toggling of the PROGRAMN pin or receiving a REFRESH command, Alternate pattern 1 is loaded. Subsequent PROGRAMN/REFRESH event loads the next pattern defined in the Multi-Boot configuration. The bitstream pattern sequence, target address of the Golden pattern, and target addresses of the Alternate patterns are defined during the Multi-Boot configuration process in the Lattice Radiant® Deployment Tool.

The Multi-Boot flow is similar to the Dual Boot flow (Figure 3.1). Each PROGRAMN/REFRESH event becomes a Dual Boot event with the addresses being different depending on the pattern being loaded.

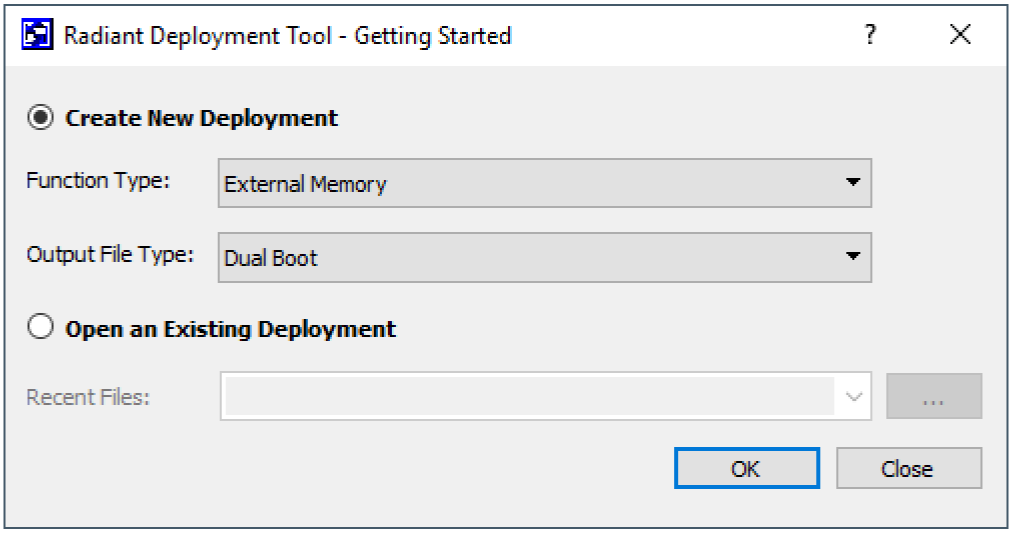
6. Creating a PROM File

The various boot features on the CrossLink-NX/Certus-NX/CertusPro-NX family of devices are simple, requiring only one external SPI Flash device, and flexible, due to the intelligent use of the JUMP command or table. The Lattice Radiant software provides a turn-key solution to implement this feature. The Lattice Deployment Tool, part of Lattice Radiant Software, merges the different patterns and the JUMP command and table into one PROM hex file. The PROM hex file can later be programmed into the internal or external Flash device using Radiant Programmer or a third-party programmer.

6.1. Using Radiant Deployment Tool to Create a Dual Boot PROM Hex File

The following steps provide the procedure for generating a Dual Boot PROM hex file using the Radiant Deployment Tool.

1. Generate the Golden and Primary bitstream files in Lattice Radiant Software.
   - Golden bitstream file MCCLK_FREQ (SPI Master Clock Frequency) setting should not exceed the external Flash device normal/standard read speed.
   - MCCLK_FREQ can be configured using the Global tab of the Device Constraint Editor in Lattice Radiant software.
2. Invoke **Lattice Radiant Deployment Tool** from **Start > Lattice Radiant Programmer > Deployment Tool**.
3. In the **Radiant Deployment Tool** window, select **External Memory** as the **Function Type** and select **Dual Boot** as the **Output File Type** (Figure 6.1).
4. Select **OK**.



Radiant Deployment Tool - Getting Started

● Create New Deployment

Function Type: External Memory

Output File Type: Dual Boot

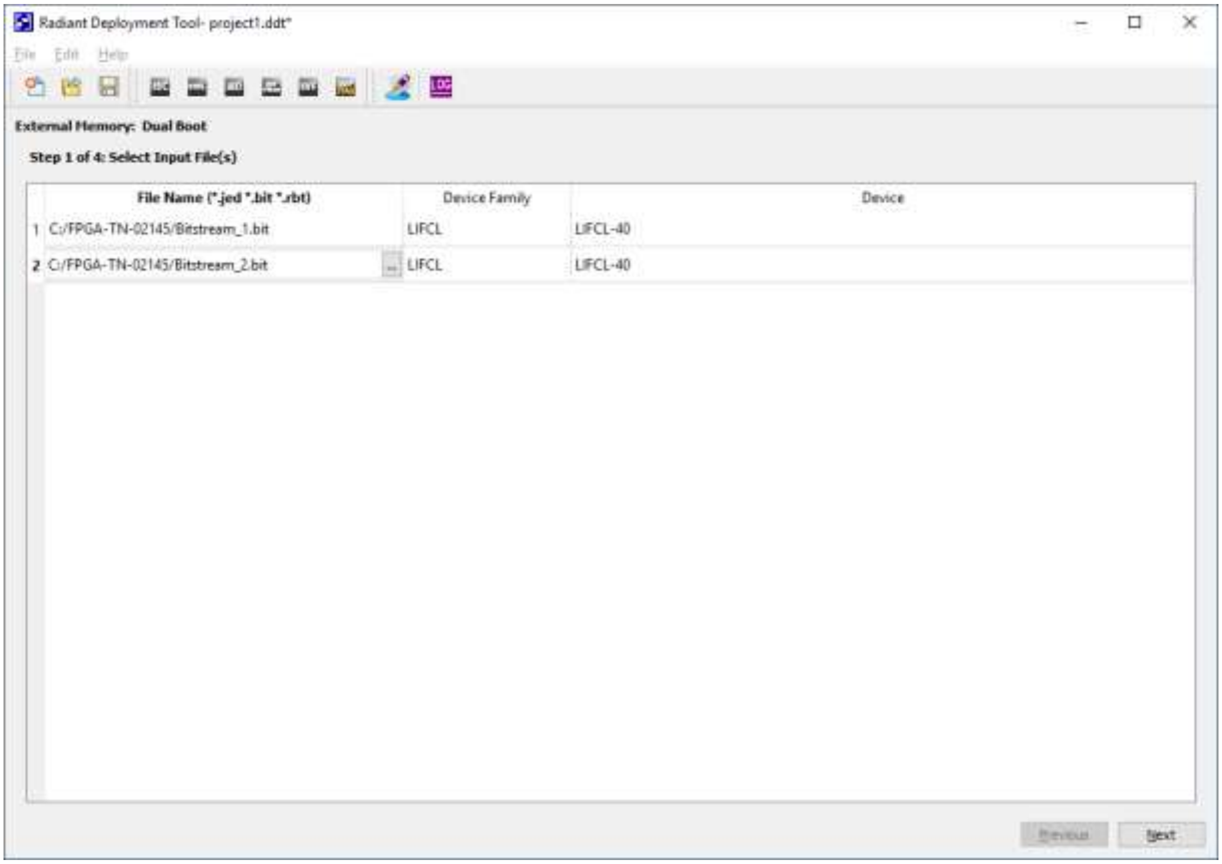
○ Open an Existing Deployment

Recent Files:

Figure 6.1. Creating New Deployment for Dual Boot PROM Hex File

Step 1 of 4: Select Input File(s) window (Figure 6.2)

- Click the **File Name** fields to browse and select the two bitstream files to be used to create the PROM hex file.
- The **Device Family** and **Device** fields auto populates based on the bitstream files selected.
- Select **Next**.



External Memory: Dual Boot

Step 1 of 4: Select Input File(s)

	File Name (*.jed *.bit *.rbt)	Device Family	Device
1	C:/FPGA-TN-02145/Bitstream_1.bit	LIFCL	LIFCL-40
2	C:/FPGA-TN-02145/Bitstream_2.bit	LIFCL	LIFCL-40

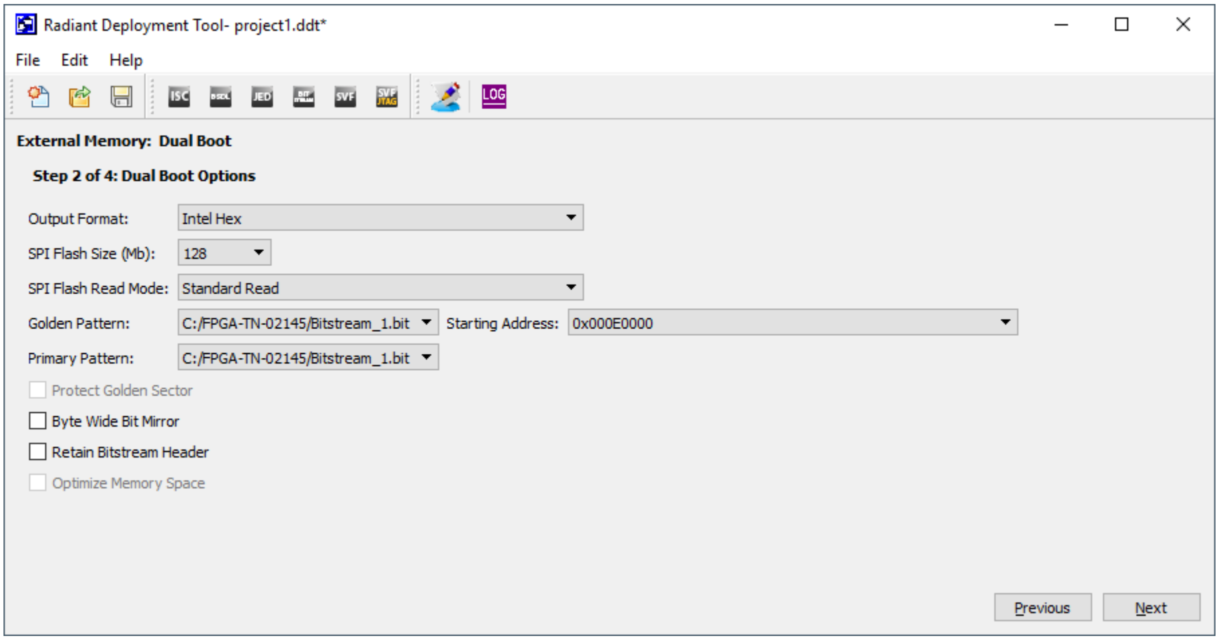
Figure 6.2. Select Input Files Window

Step 2 of 4: Dual Boot Options window (Figure 6.3)

- Select the **Output Format** (Intel Hex, Motorola Hex or Extended Tektronix Hex).
- Select the **SPI Flash Size** (4, 8, 16, 32, 64, 128, 256 or 512 Mb).
- Select **SPI Flash Read Mode** (Standard Read, Fast Read, Dual I/O SPI Flash Read or Quad I/O SPI Flash Read).
- The **Radiant Deployment Tool** automatically assigns the bitstream files selected in Step 1 to be used for the Golden pattern and Primary pattern.
  - Change the pattern options by clicking on the drop down menu of the respective fields.
  - The Starting Address of the Golden pattern is automatically assigned.
  - Change the Starting Address of the Golden pattern by clicking on the drop down menu.
- Select the following options as required.
  - **Byte Wide Bit Mirror** – Flips each byte in Intel, Extended Tektronix, or Motorola hexadecimal data files.

    For example, 0xCD (b1100 1101) can become 0xB3 (b1011 0011) when this is selected.
  - **Retain Bitstream Header** – By default, Radiant Deployment Tool replaces the bitstream header information (name, version number and date of the file) with 0xFF values.

    Selecting this option retains the header information that was generated as the header.
- Select **Next**.



External Memory: Dual Boot

Step 2 of 4: Dual Boot Options

Output Format: Intel Hex

SPI Flash Size (Mb): 128

SPI Flash Read Mode: Standard Read

Golden Pattern: C:/FPGA-TN-02145/Bitstream_1.bit Starting Address: 0x000E0000

Primary Pattern: C:/FPGA-TN-02145/Bitstream_1.bit

□ Protect Golden Sector

□ Byte Wide Bit Mirror

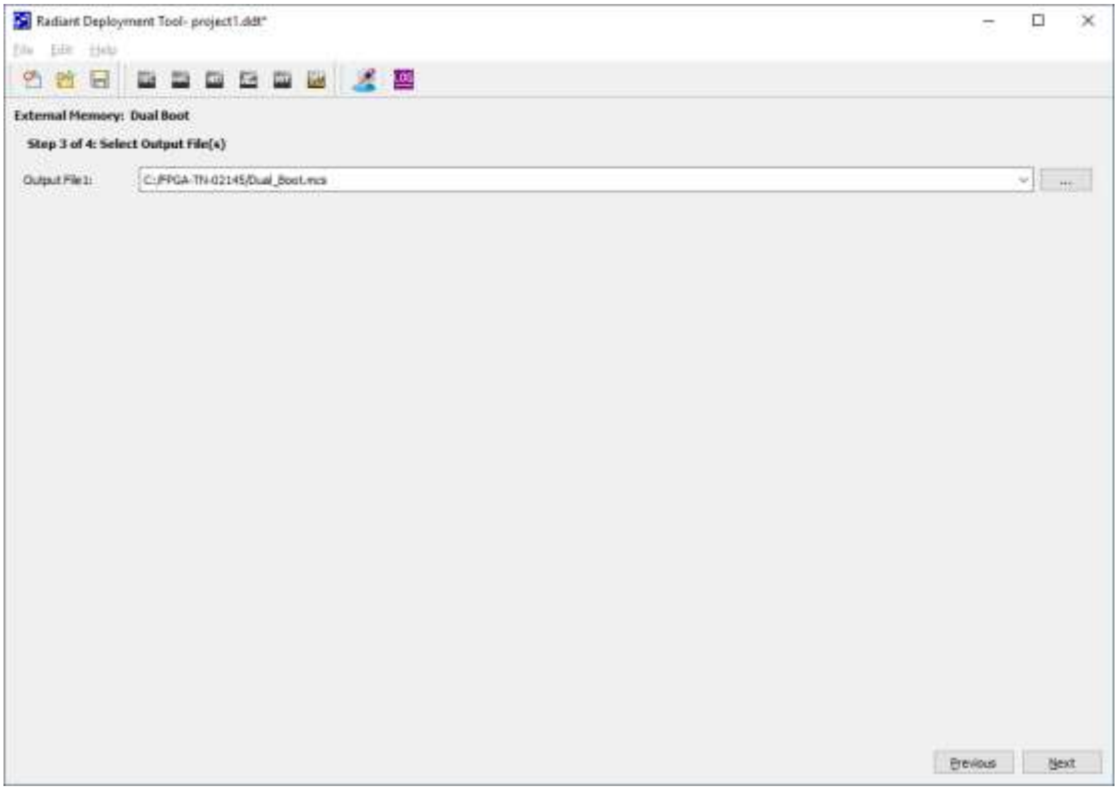
□ Retain Bitstream Header

□ Optimize Memory Space

Figure 6.3. Dual Boot Options Window

Step 3 of 4: Select Output File(s) window (Figure 6.4)

- Specify the name of the output PROM hex file in the **Output File 1** field.
- Select **Next**.



External Memory: Dual Boot

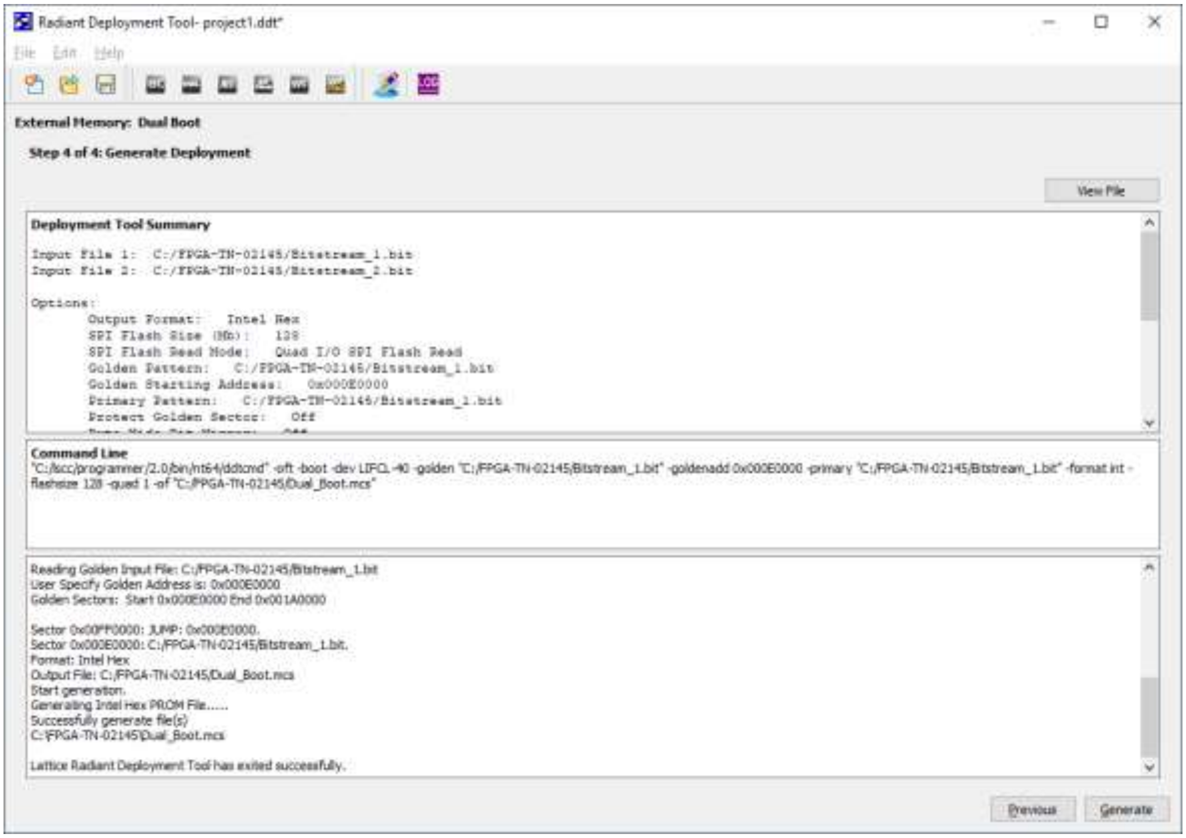
Step 3 of 4: Select Output File(s)

Output File 1: C:/FPGA-TN-02145/Dual_Boot.mcs

Figure 6.4. Select Output File Window

Step 4 of 4: Generate Deployment window (Figure 6.5)

- Review the summary information.
- If everything is correct, click the **Generate** button.
- The **Generate Deployment** pane should indicate that the PROM file was generated successfully.
- Save the deployment settings by selecting **File > Save**.
- To exit, select **File > Exit**.


Radiant Deployment Tool- project1.ddt*
File
Edit
Help
External Memory: Dual Boot
Step 4 of 4: Generate Deployment
View File

Deployment Tool Summary

```
Input File 1:  C:/FPGA-TN-02145/Bitstream_1.bit
Input File 2:  C:/FPGA-TN-02145/Bitstream_2.bit

Options:
        Output Format:   Intel Hex
        SPI Flash Size (Mb):   128
        SPI Flash Read Mode:   Quad I/O SPI Flash Read
        Golden Pattern:   C:/FPGA-TN-02145/Bitstream_1.bit
        Golden Starting Address:   0x000E0000
        Primary Pattern:   C:/FPGA-TN-02145/Bitstream_1.bit
        Protect Golden Sector:   Off
```

Command Line

```
"C:/lscc/programmer/2.0/bin/nt64/ddtcmd" -oft -boot -dev LIFCL-40 -golden "C:/FPGA-TN-02145/Bitstream_1.bit" -goldenadd 0x000E0000 -primary "C:/FPGA-TN-02145/Bitstream_1.bit" -format int -flashsize 128 -quad 1 -of "C:/FPGA-TN-02145/Dual_Boot.mcs"
```

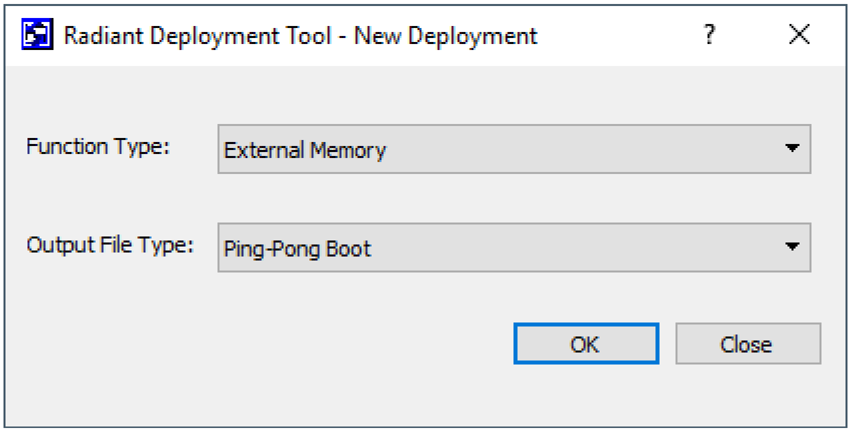
```
Reading Golden Input File: C:/FPGA-TN-02145/Bitstream_1.bit
User Specify Golden Address is: 0x000E0000
Golden Sectors:  Start 0x000E0000 End 0x001A0000

Sector 0x00FF0000: JUMP: 0x000E0000.
Sector 0x000E0000: C:/FPGA-TN-02145/Bitstream_1.bit.
Format: Intel Hex
Output File: C:/FPGA-TN-02145/Dual_Boot.mcs
Start generation.
Generating Intel Hex PROM File......
Successfully generate file(s)
C:\FPGA-TN-02145\Dual_Boot.mcs

Lattice Radiant Deployment Tool has exited successfully.
```

Figure 6.5. Generate Deployment Window

6.2. Using the Lattice Deployment Tool to Create a Ping-Pong Boot PROM Hex File

The following steps provide the procedure for generating a Ping-Pong boot PROM hex file using the Radiant Deployment Tool.

1. Generate the Primary and Secondary bitstream files in Lattice Radiant software.
   - When the Primary or Secondary bitstream is the second boot option, it by default becomes the Golden bitstream.
   - Golden bitstream file MCCLK_FREQ (SPI Master Clock Frequency) setting should not exceed the external Flash device normal/standard read speed.
   - MCCLK_FREQ can be configured using the Global tab of the Device Constraint Editor in Lattice Radiant software.
2. Invoke **Lattice Radiant Deployment Tool** from **Start > Lattice Radiant Programmer > Deployment Tool**.
3. In the **Radiant Deployment Tool** window, select **External Memory** as the **Function Type** and select **Ping-Pong Boot** as the **Output File Type** (Figure 6.6).
4. Select **OK**.



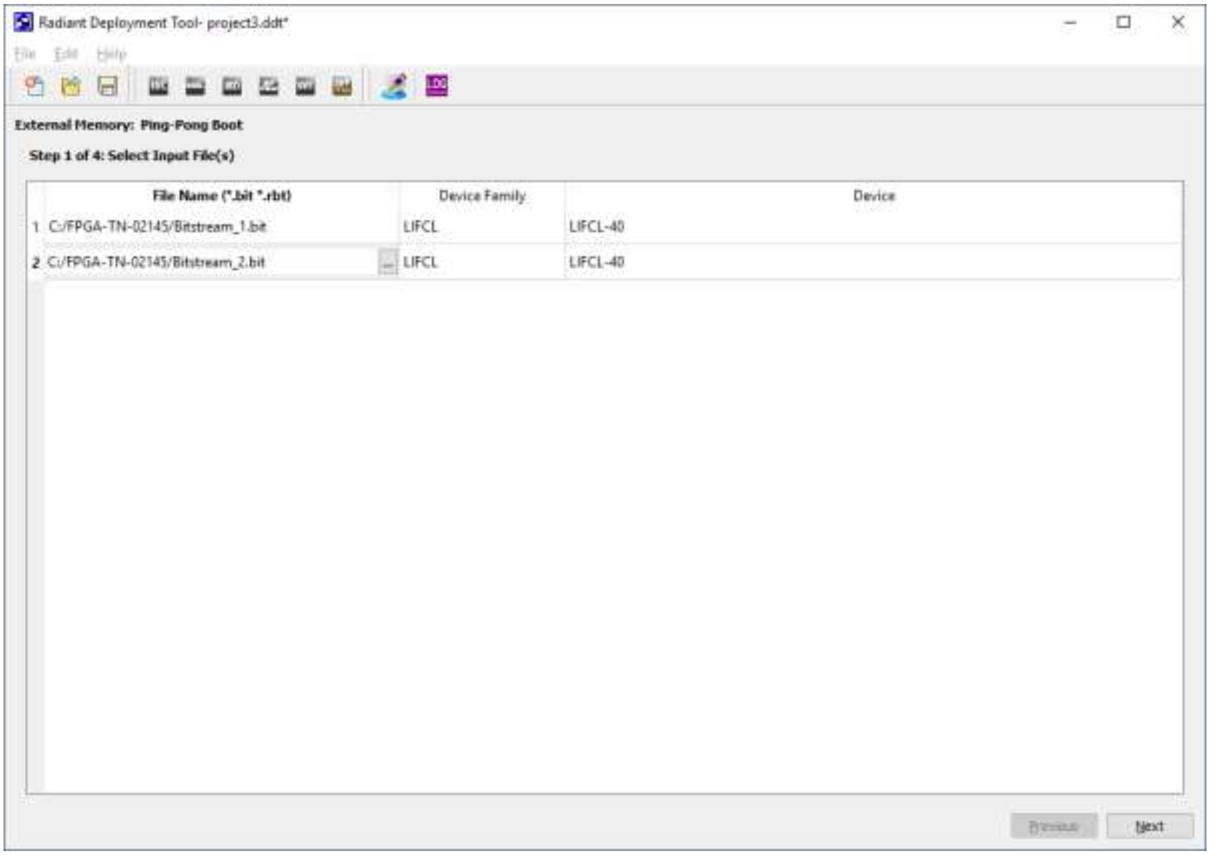
Function Type: External Memory

Output File Type: Ping-Pong Boot

Figure 6.6. Creating New Deployment for Ping-Pong Boot PROM Hex File

Step 1 of 4: Select Input File(s) window (Figure 6.7)

- Click the **File Name** fields to browse and select the two bitstream files to be used to create the PROM hex file.
- The **Device Family** and **Device** fields auto populates based on the bitstream files.
- Select **Next**.



Radiant Deployment Tool- project3.ddt*

External Memory: Ping-Pong Boot

Step 1 of 4: Select Input File(s)

	File Name (*.bit *.rbt)	Device Family	Device
1	C:/FPGA-TN-02145/Bitstream_1.bit	LIFCL	LIFCL-40
2	C:/FPGA-TN-02145/Bitstream_2.bit	LIFCL	LIFCL-40

Figure 6.7. Select Input Files Window

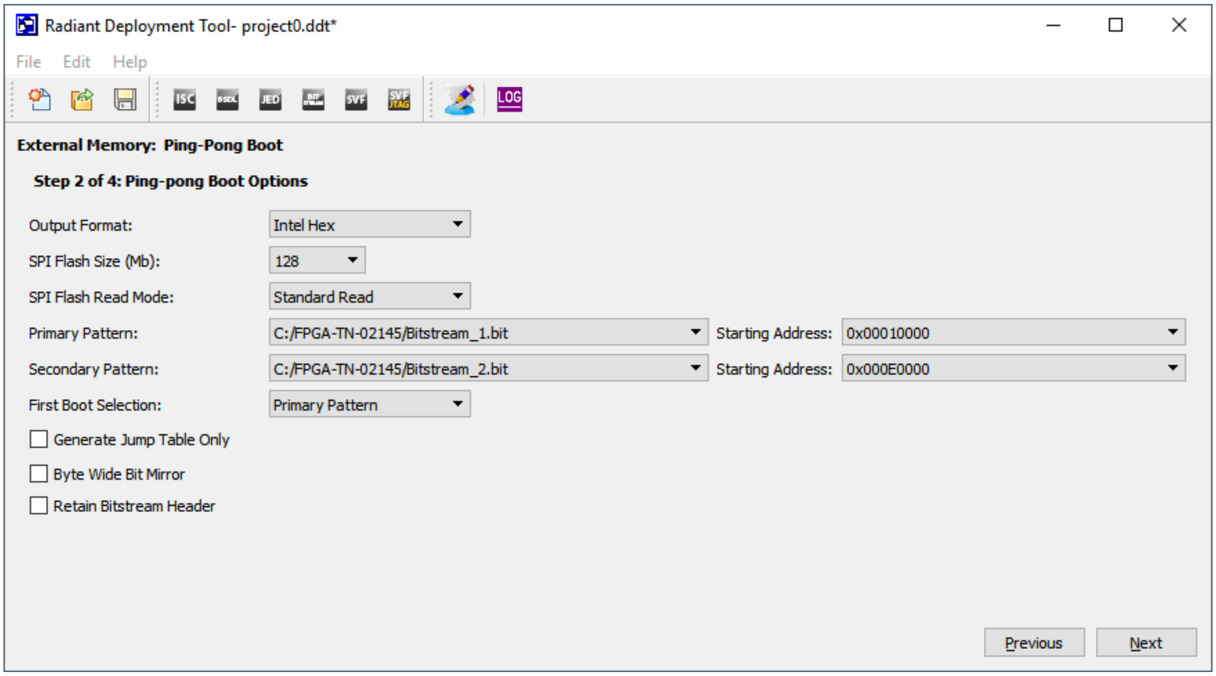
Step 2 of 4: Ping-Pong Boot Options window (Figure 6.8)

- Select the **Output Format** (Intel Hex, Motorola Hex, or Extended Tektronix Hex).
- Select the **SPI Flash Size** (4, 8, 16, 32, 64, 128, 256 or 512 Mb).
- Select **SPI Flash Read Mode** (Standard Read, Fast Read, Dual I/O SPI Flash Read or Quad I/O SPI Flash Read).
- The **Radiant Deployment Tool** automatically assigns the bitstream files selected in Step 1 to be used for Primary and Secondary Patterns.
  - Change the pattern options by clicking on the drop down menu of the respective field.
  - The Starting Address of the Primary Pattern is automatically assigned and can be modified by clicking on the drop down menu.
  - The Starting Address of the Secondary Pattern is automatically assigned and can be modified by clicking on the drop down menu.
- Select the following options as required.
  - **Generate Jump Table Only** – Generates a JUMP table to select an image for booting without changing the physical location of the images in the external SPI Flash.

    For example, a JUMP table file can be created to attempt to load Bitstream_2.bit file first. This new JUMP table file can be programmed to the external Flash to overwrite the previous JUMP table file.
  - **Byte Wide Bit Mirror** – Flips each byte in Intel, Extended Tektronix or Motorola hexadecimal data files

    For example, 0xCD (b1100 1101) becomes 0xB3 (b1011 0011) when this is selected.
  - **Retain Bitstream Header** – By default, Radiant Deployment Tool replaces the bitstream header information (name, version number and date of the file) with 0xFF values.

    Selecting this option retains the header information that was generated as the header.
- Select **Next**.


Radiant Deployment Tool- project0.ddt*
File Edit Help
ISC BSCL JED BIT SVF SVF JTAG LOG

External Memory: Ping-Pong Boot

Step 2 of 4: Ping-pong Boot Options

Output Format: Intel Hex

SPI Flash Size (Mb): 128

SPI Flash Read Mode: Standard Read

Primary Pattern: C:/FPGA-TN-02145/Bitstream_1.bit Starting Address: 0x00010000

Secondary Pattern: C:/FPGA-TN-02145/Bitstream_2.bit Starting Address: 0x000E0000

First Boot Selection: Primary Pattern

□ Generate Jump Table Only

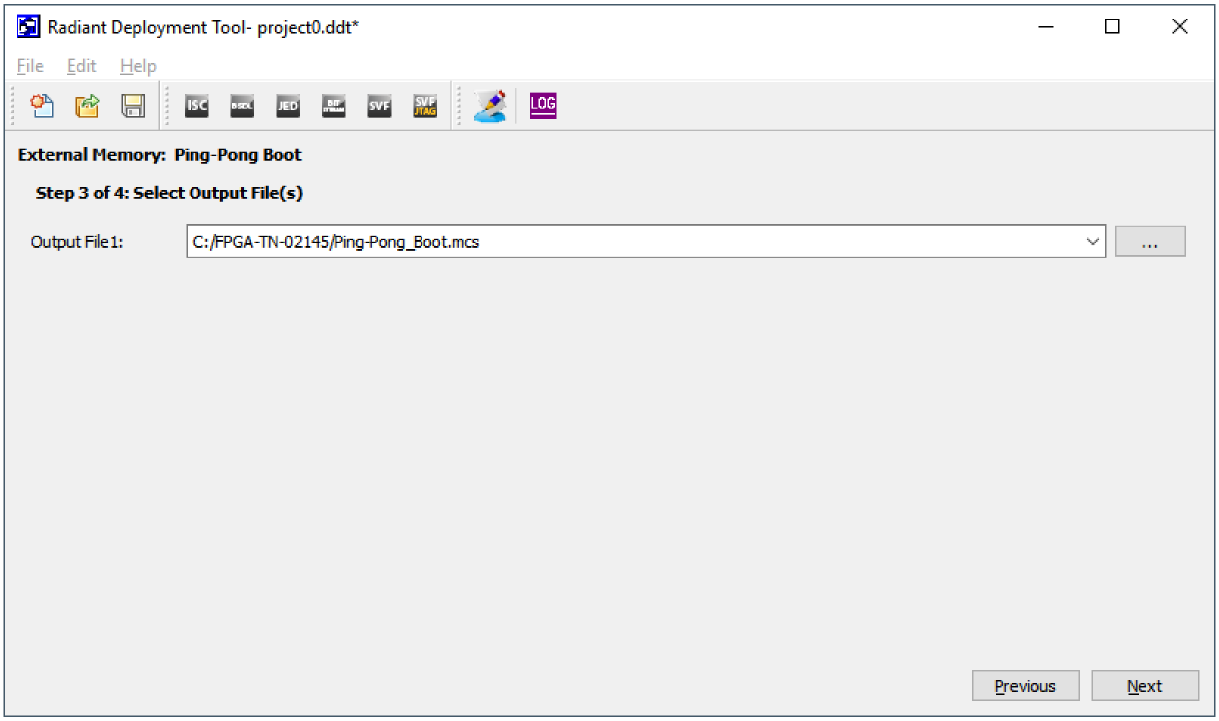
□ Byte Wide Bit Mirror

□ Retain Bitstream Header

Figure 6.8. Ping-Pong Boot Options Window

Step 3 of 4: Select Output File(s) window (Figure 6.9)

- Specify the name of the output PROM hex file in the **Output File 1** field.
- Select **Next**.



External Memory: Ping-Pong Boot

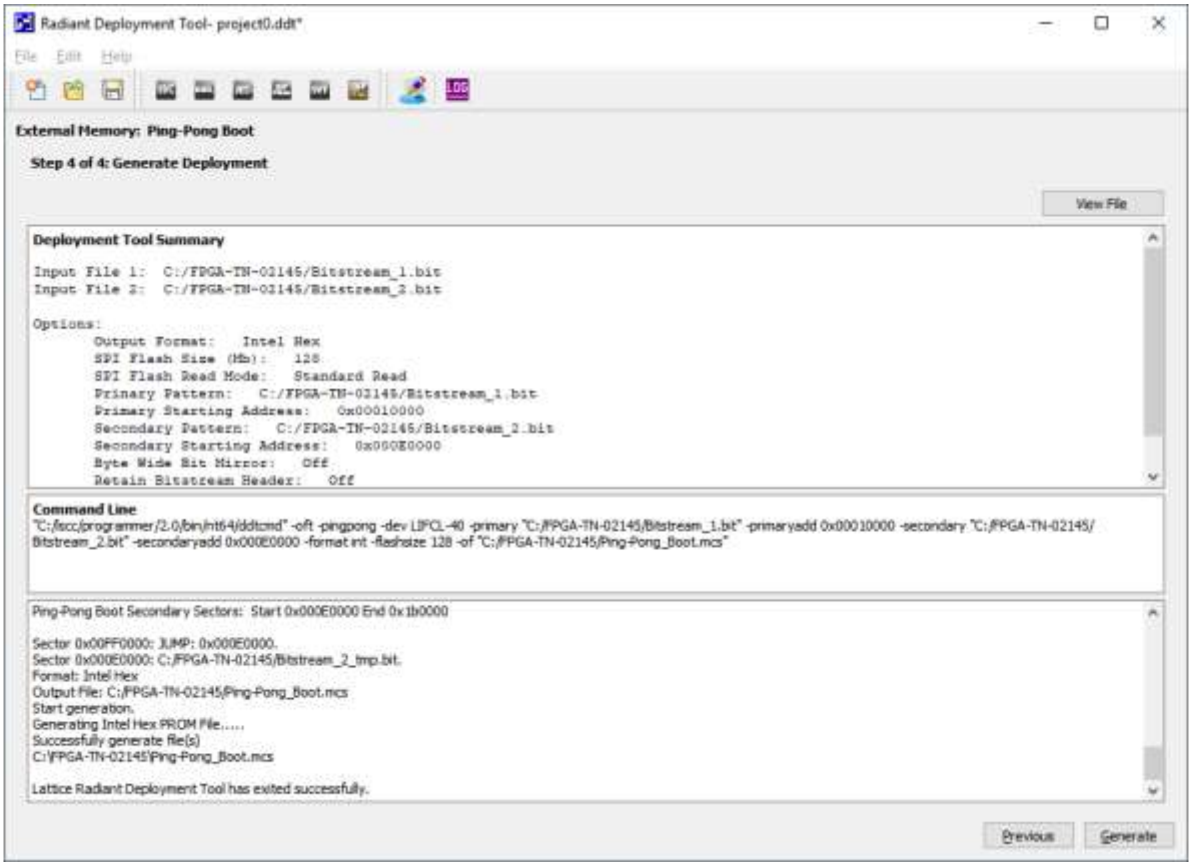
Step 3 of 4: Select Output File(s)

Output File 1: C:/FPGA-TN-02145/Ping-Pong_Boot.mcs

Figure 6.9. Select Output File Window

Step 4 of 4: Generate Deployment window (Figure 6.10)

- Review the summary information.
- If everything is correct, click the **Generate** button.
- The **Generate Deployment** pane should indicate the PROM file was generated successfully.
- Save the deployment settings by selecting **File > Save**.
- To exit, select **File > Exit**.



External Memory: Ping-Pong Boot

Step 4 of 4: Generate Deployment

Deployment Tool Summary

```
Input File 1:  C:/FPGA-TN-02145/Bitstream_1.bit
Input File 2:  C:/FPGA-TN-02145/Bitstream_2.bit

Options:
        Output Format:   Intel Hex
        SPI Flash Size (Mb):   128
        SPI Flash Read Mode:   Standard Read
        Primary Pattern:   C:/FPGA-TN-02145/Bitstream_1.bit
        Primary Starting Address:   0x00010000
        Secondary Pattern:   C:/FPGA-TN-02145/Bitstream_2.bit
        Secondary Starting Address:   0x000E0000
        Byte Wide Bit Mirror:   Off
        Retain Bitstream Header:   Off
```

Command Line

```
"C:/lscc/programmer/2.0/bin/nt64/ddtcmd" -oft -pingpong -dev LIFCL-40 -primary "C:/FPGA-TN-02145/Bitstream_1.bit" -primaryadd 0x00010000 -secondary "C:/FPGA-TN-02145/
Bitstream_2.bit" -secondaryadd 0x000E0000 -format int -flashsize 128 -of "C:/FPGA-TN-02145/Ping-Pong_Boot.mcs"
```

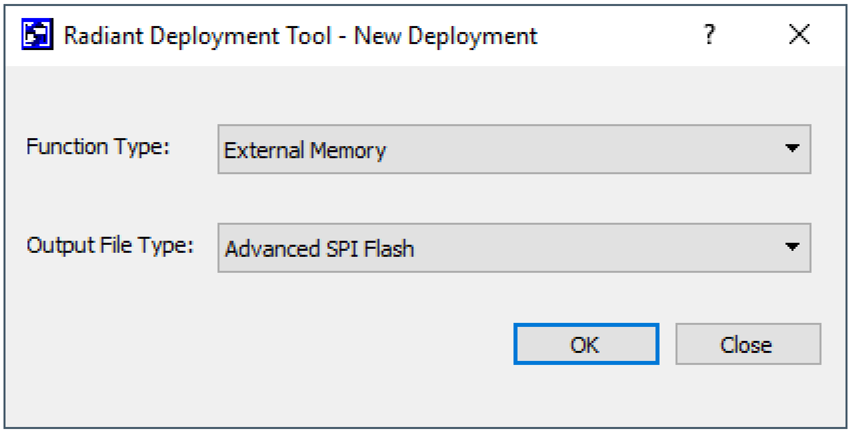
```
Ping-Pong Boot Secondary Sectors:  Start 0x000E0000 End 0x1b0000

Sector 0x00FF0000: JUMP: 0x000E0000.
Sector 0x000E0000: C:/FPGA-TN-02145/Bitstream_2_tmp.bit.
Format: Intel Hex
Output File: C:/FPGA-TN-02145/Ping-Pong_Boot.mcs
Start generation.
Generating Intel Hex PROM File.....
Successfully generate file(s)
C:\FPGA-TN-02145\Ping-Pong_Boot.mcs

Lattice Radiant Deployment Tool has exited successfully.
```

Figure 6.10. Generate Deployment Window

6.3. Using Radiant Deployment Tool to Create a Multi-Boot PROM Hex File

The following steps provide the procedure for generating a Multi-Boot PROM hex file using the Radiant Deployment Tool. This procedure is an example for four total bitstreams, Primary Pattern, Golden Pattern, Alternate Pattern 1 and Alternate Pattern 2.

1. Generate all the bitstream files needed in Lattice Radiant Software.
    - Golden bitstream file MCCLK_FREQ (SPI Master Clock Frequency) setting should not exceed the external Flash device normal/standard read speed.
    - MCCLK_FREQ can be configured under the Global tab of the Device Constraint Editor in Lattice Radiant software.
2. Invoke **Lattice Radiant Deployment Tool** from **Start > Lattice Radiant Programmer > Deployment Tool**.
3. In the Radiant Deployment Tool window, select **External Memory** as the **Function Type** and select **Advanced SPI Flash** as the **Output File Type** (Figure 6.11).
4. Select **OK**.


Radiant Deployment Tool - New Deployment

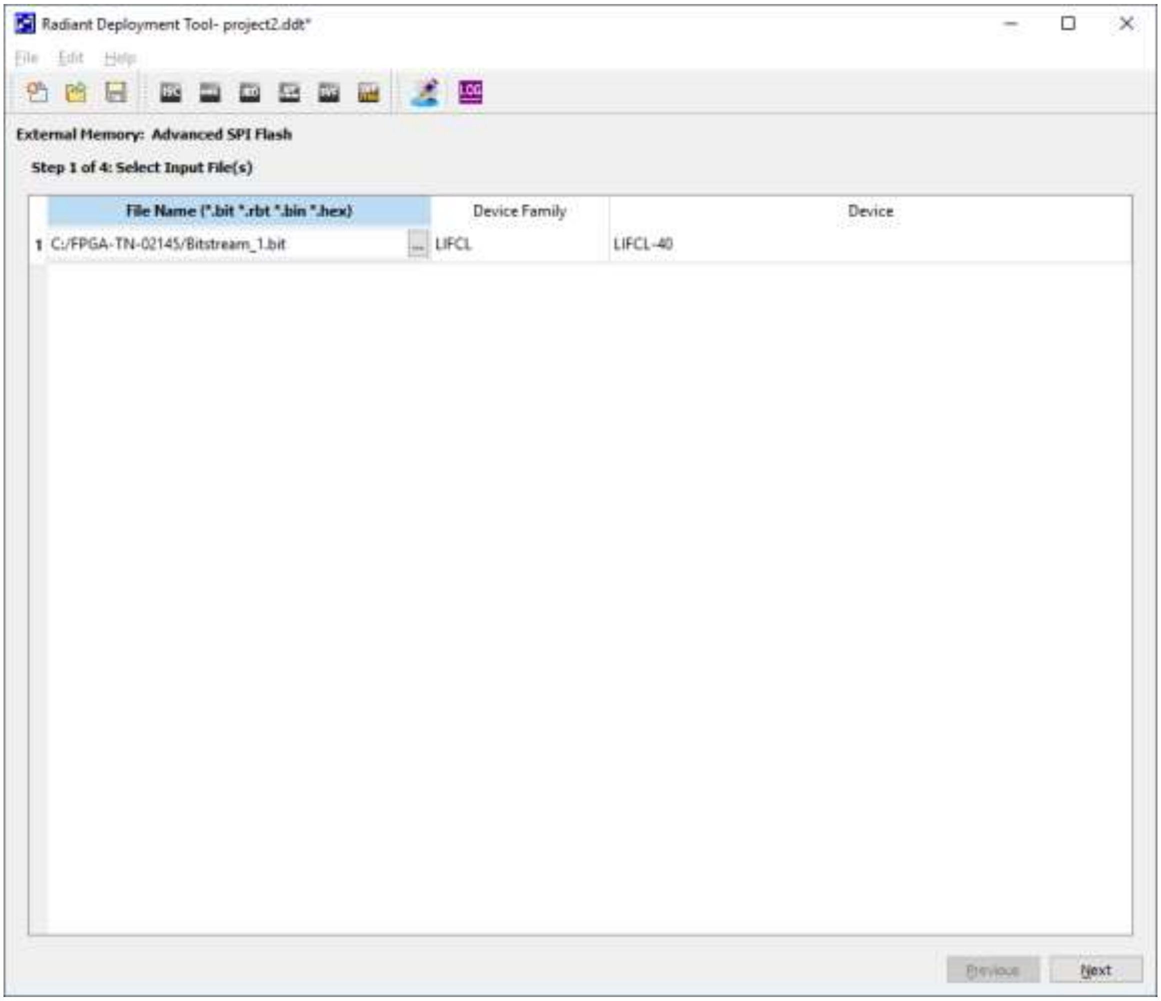
Function Type: External Memory

Output File Type: Advanced SPI Flash

Figure 6.11. Creating New Deployment for Multi-Boot

Step 1 of 4: Select Input File(s) window (Figure 6.12)

- Click the **File Name** field to browse and select the primary bitstream file to be used to create the PROM hex file.
- The **Device Family** and **Device** fields auto populates based on the bitstream files selected.
- Select **Next**.



External Memory: Advanced SPI Flash

Step 1 of 4: Select Input File(s)

	File Name (*.bit *.rbt *.bin *.hex)	Device Family	Device
1	C:/FPGA-TN-02145/Bitstream_1.bit	LIFCL	LIFCL-40

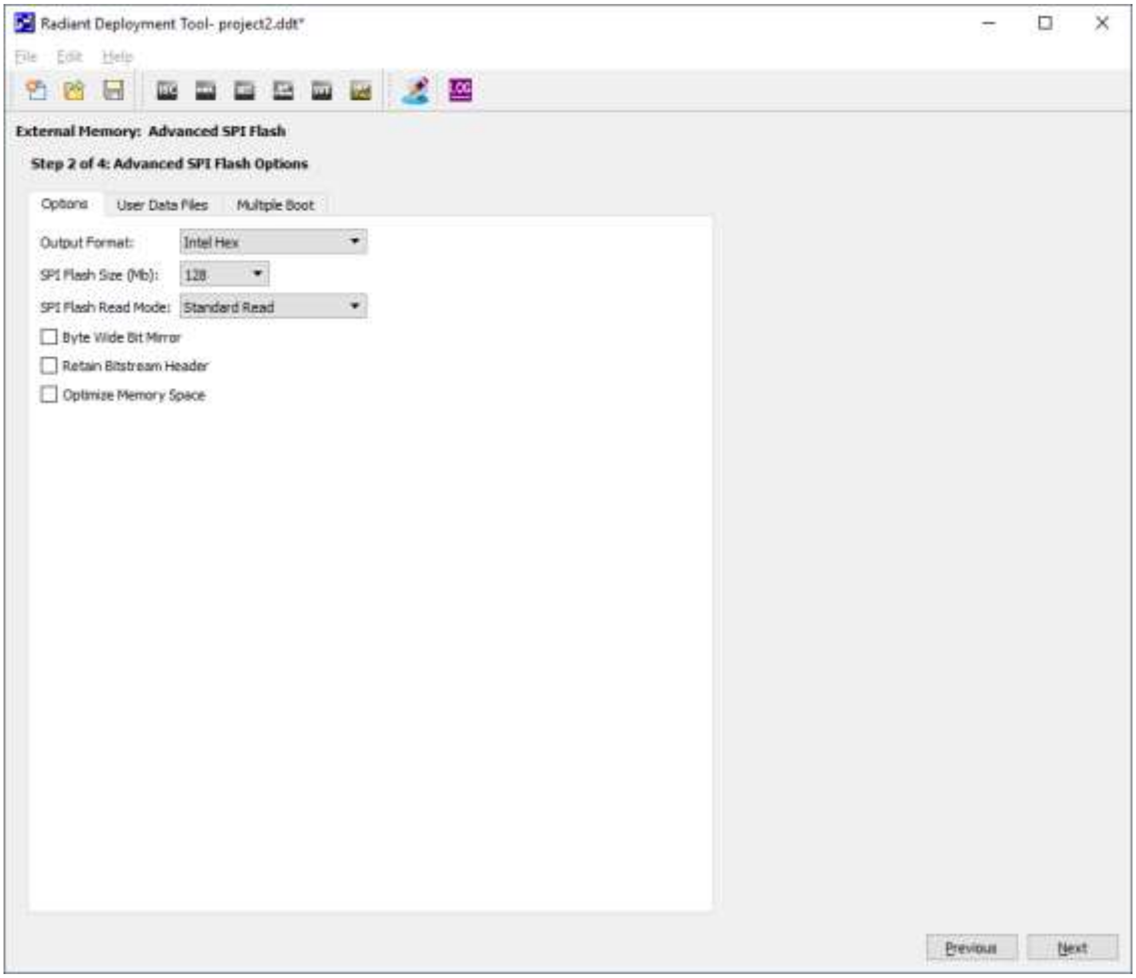
Figure 6.12. Select Input File Window

Step 2 of 4: Advanced SPI Flash Options window (Figure 6.13)

- Go to the **Options** tab.
- Select the **Output Format** (Intel Hex, Motorola Hex or Extended Tektronix Hex).
- Select the **SPI Flash Size** (4, 8, 16, 32, 64, 128, 256 or 512 Mb).
- Select **SPI Flash Read Mode** (Standard Read, Fast Read, Dual I/O SPI Flash Read or Quad I/O SPI Flash Read).
- Select the following options as required:
  - **Byte Wide Bit Mirror** – Flips each byte in Intel, Extended Tektronix, or Motorola hexadecimal data files
    For example, 0xCD (b1100 1101) becomes 0xB3 (b1011 0011) when this is selected.
  - **Retain Bitstream Header** – By default, Radiant Deployment Tool replaces the bitstream header information (name, version number and date of the file) with 0xFF values. Selecting this option retains the header information that was generated as the header.
  - **Optimize Memory Space** – By default, the Radiant Deployment Tool uses the worst case file size for SPI Flash memory space allocation.

    a. Worst case size is an uncompressed bitstream with maximum EBR and PCS. This allows maximum flexibility for field upgrades. If a new Primary Pattern file size grows significantly due to less compression or adding EBR blocks, it is guaranteed to fit in the sectors already allocated for Primary Pattern.

    b. When this option is selected, the Radiant Deployment Tool uses the actual file size for the address allocation. This reduces wasted SPI Flash space and may allow for a smaller Flash device. If one or more of the new patterns have smaller compression ratio or more EBR/PCS, the new pattern(s) can encroach into another pattern bitstream memory space. If this occurs, the entire SPI Flash needs to be erased/re-programmed with a new Hex file.
- Go to the **Multiple Boot** tab (Figure 6.14).
- Select the **Multiple Boot** option.
- Click on the **Golden Pattern** browse button to select the Golden Pattern bitstream.
  - The Starting Address of the Golden Pattern is automatically assigned. Change the Starting Address of the Golden Pattern by clicking on the drop down menu.
- Select the following option as required:
  Protect Golden Sector – By default, the golden sector, where the Golden Pattern is stored, is located immediately after the primary sector to save SPI Flash space. When this option is selected, the Golden Pattern location is moved to the first sector in the upper half of the SPI Flash. The new location is reflected in the Golden Patten Starting Address field. This protects the Golden Pattern from accidental erase/reprogram by protecting the upper half of the SPI Flash when it is programmed.
- In the **Number of Alternate Patterns** field, select the number of alternate patterns to include through the drop down menu.
- In the **Alternate Pattern 1** field, click on the browse button to select the first alternate pattern.
  - The Starting Address of Alternate Pattern 1 is automatically populated. You can change the Starting Address of Alternate Pattern 1 by clicking on the drop down menu.
- The **Next Alternate Pattern to Configure** field is automatically populated.
  - This is the pattern that is loaded during the next PROGRAMN/REFRESH event. You can change the pattern by clicking on the drop down menu.
- In the **Alternate Pattern 2** field, click on the browse button to select the second alternate pattern.
  - The Starting Address of Alternate Pattern 2 is automatically populated. You can change the Starting Address of Alternate Pattern 2 by clicking on the drop down menu.
- The **Next Alternate Pattern to Configure** field is automatically populated.
  - This is the pattern that is loaded during the next PROGRAMN/REFRESH event. You can change the pattern by clicking on the drop down menu.
- Select **Next**.

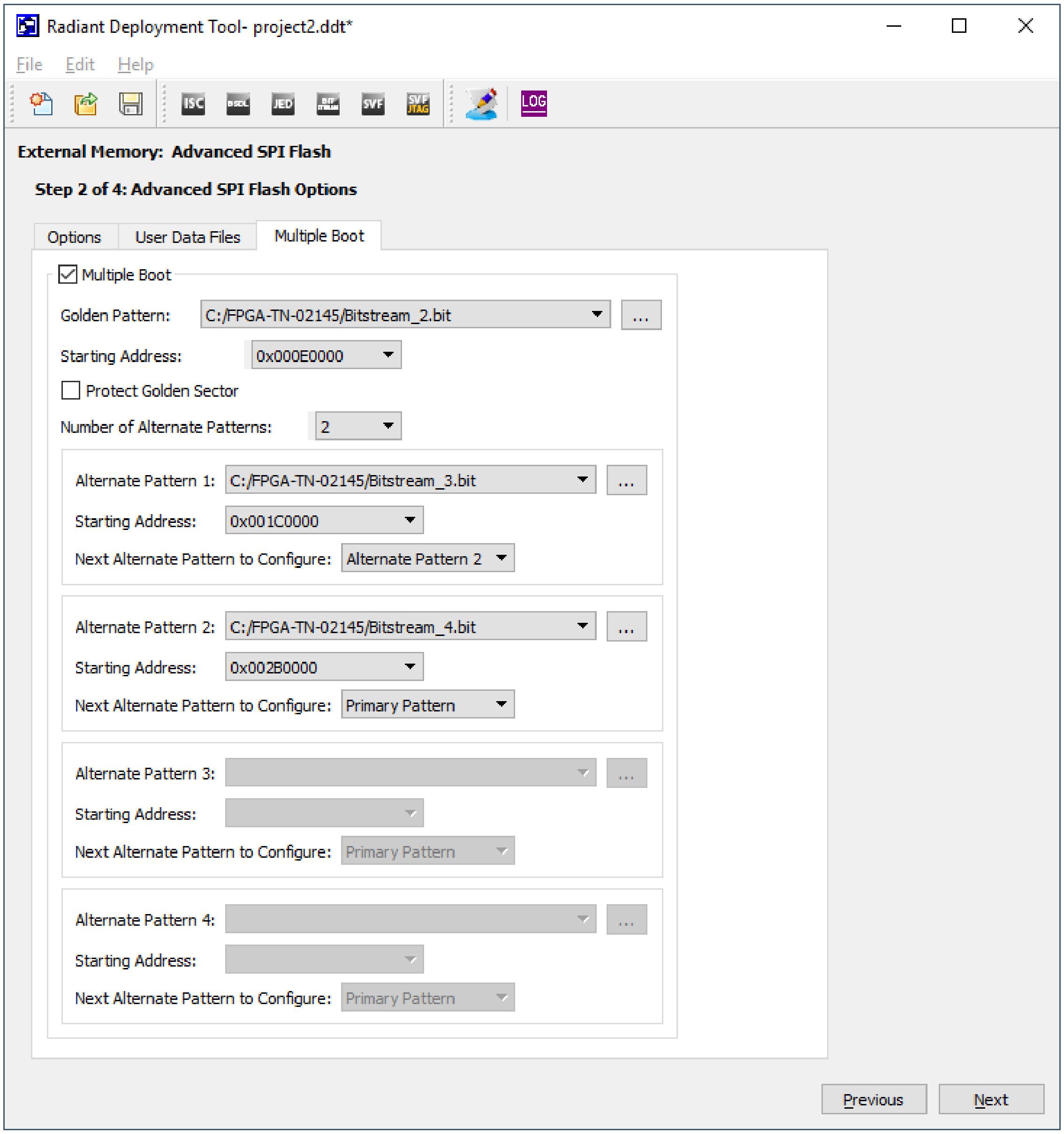


External Memory: Advanced SPI Flash

Step 2 of 4: Advanced SPI Flash Options

Output Format: Intel Hex
SPI Flash Size (Mb): 128
SPI Flash Read Mode: Standard Read
☐ Byte Wide Bit Mirror
☐ Retain Bitstream Header
☐ Optimize Memory Space

Figure 6.13. Advanced SPI Flash Options – Options Tab Window



Radiant Deployment Tool- project2.ddt*

External Memory: Advanced SPI Flash

Step 2 of 4: Advanced SPI Flash Options

☑ Multiple Boot

Golden Pattern: C:/FPGA-TN-02145/Bitstream_2.bit

Starting Address: 0x000E0000

□ Protect Golden Sector

Number of Alternate Patterns: 2

Alternate Pattern 1: C:/FPGA-TN-02145/Bitstream_3.bit

Starting Address: 0x001C0000

Next Alternate Pattern to Configure: Alternate Pattern 2

Alternate Pattern 2: C:/FPGA-TN-02145/Bitstream_4.bit

Starting Address: 0x002B0000

Next Alternate Pattern to Configure: Primary Pattern

Alternate Pattern 3:

Starting Address:

Next Alternate Pattern to Configure: Primary Pattern

Alternate Pattern 4:

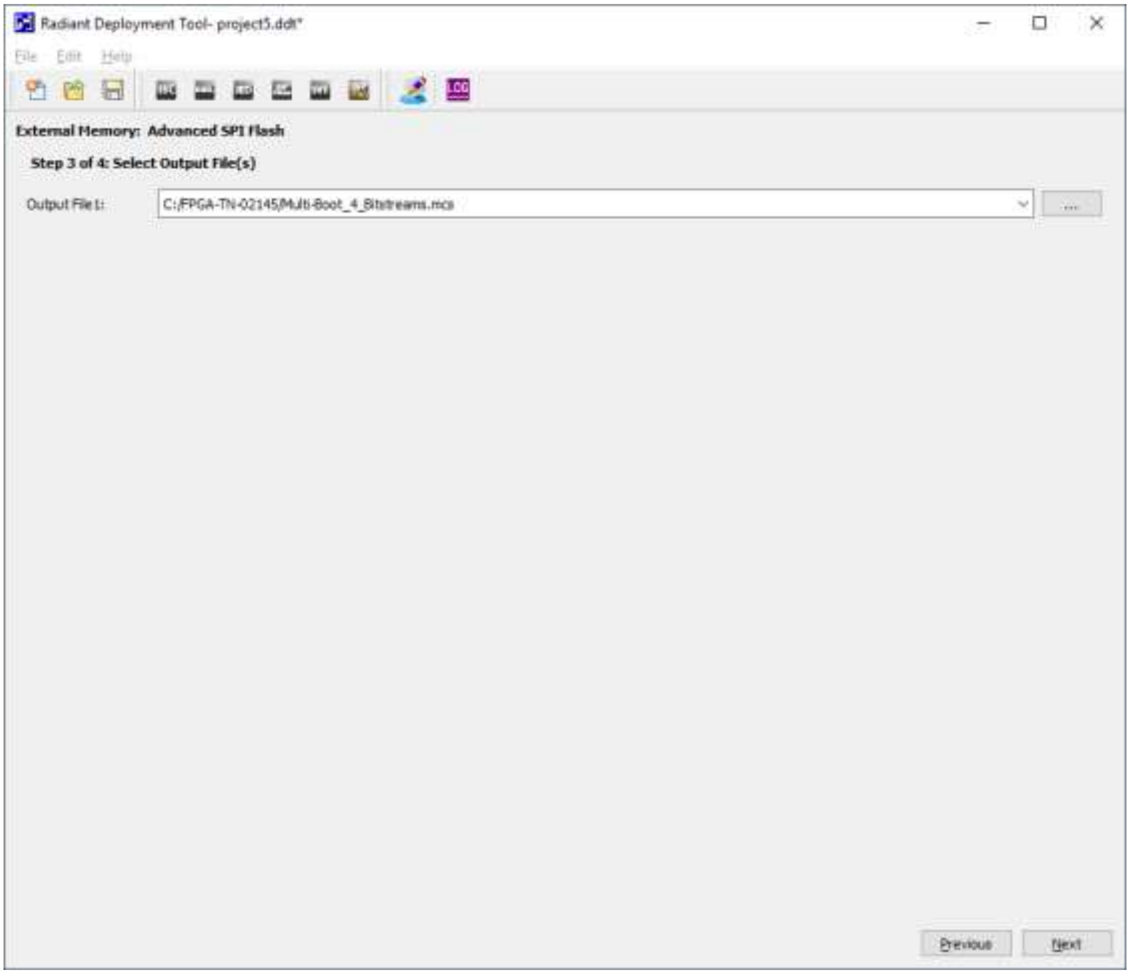
Starting Address:

Next Alternate Pattern to Configure: Primary Pattern

Figure 6.14. Advanced SPI Flash Options – Multiple Boot Tab Window

Step 3 of 4: Select Output File(s) window (Figure 6.15)

- Specify the name of the output PROM hex file in the **Output File 1** field.
- Select **Next**.



External Memory: Advanced SPI Flash

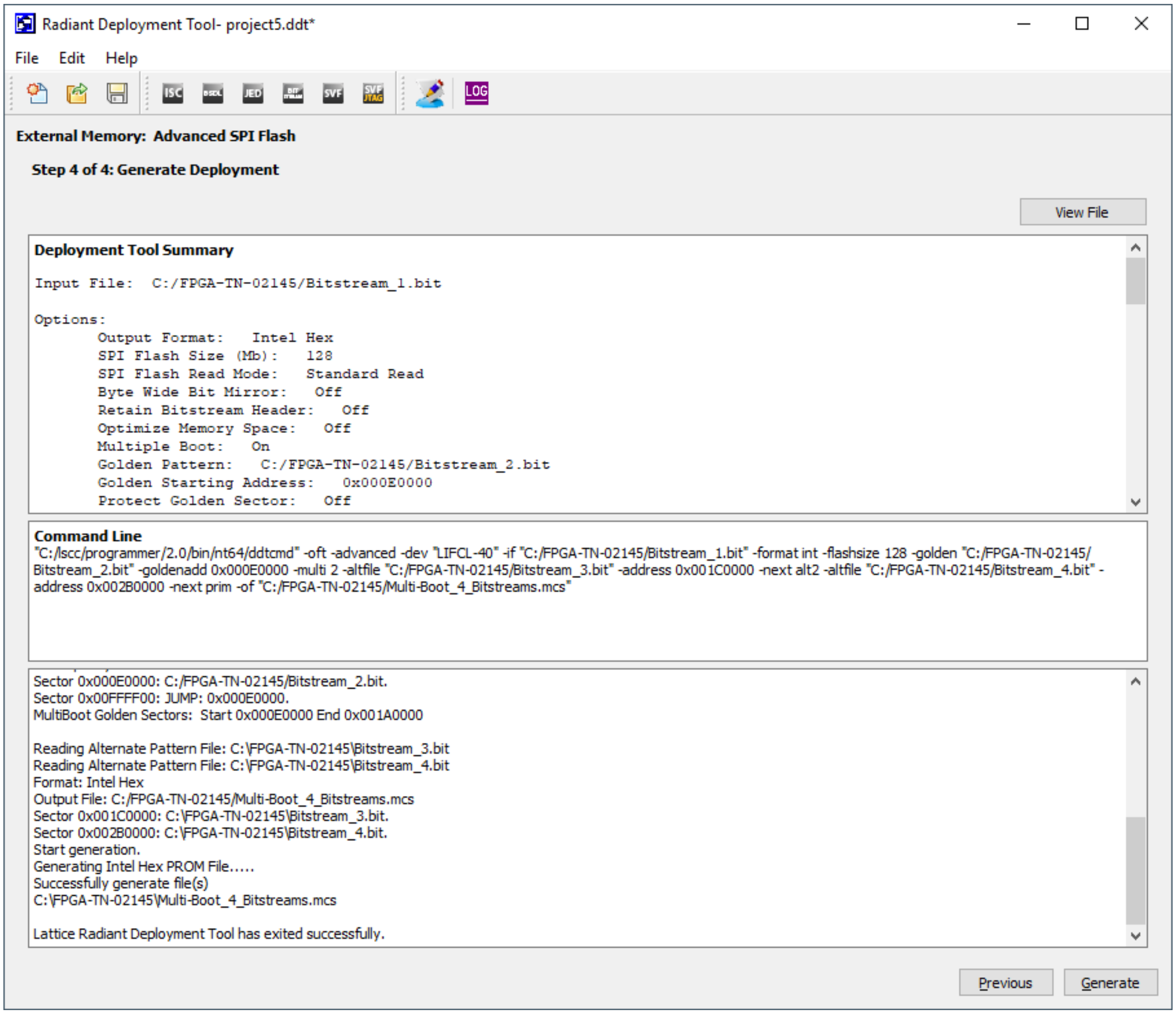
Step 3 of 4: Select Output File(s)

Output File 1: C:/FPGA-TN-02145/Multi-Boot_4_Bitstreams.mcs

Figure 6.15. Select Output File Window

Step 4 of 4: Generate Deployment window (Figure 6.16)

- Review the summary information.
- If everything is correct, click the **Generate** button.
- The **Generate Deployment** pane should indicate the PROM file was generated successfully.
- Save the deployment settings by selecting **File > Save**.
- To exit, select **File > Exit**.



External Memory: Advanced SPI Flash

Step 4 of 4: Generate Deployment

Deployment Tool Summary

```
Input File:  C:/FPGA-TN-02145/Bitstream_1.bit

Options:
        Output Format:   Intel Hex
        SPI Flash Size (Mb):   128
        SPI Flash Read Mode:   Standard Read
        Byte Wide Bit Mirror:   Off
        Retain Bitstream Header:   Off
        Optimize Memory Space:   Off
        Multiple Boot:   On
        Golden Pattern:   C:/FPGA-TN-02145/Bitstream_2.bit
        Golden Starting Address:   0x000E0000
        Protect Golden Sector:   Off
```

Command Line

```
"C:/lscc/programmer/2.0/bin/nt64/ddtcmd" -oft -advanced -dev "LIFCL-40" -if "C:/FPGA-TN-02145/Bitstream_1.bit" -format int -flashsize 128 -golden "C:/FPGA-TN-02145/Bitstream_2.bit" -goldenadd 0x000E0000 -multi 2 -altfile "C:/FPGA-TN-02145/Bitstream_3.bit" -address 0x001C0000 -next alt2 -altfile "C:/FPGA-TN-02145/Bitstream_4.bit" -address 0x002B0000 -next prim -of "C:/FPGA-TN-02145/Multi-Boot_4_Bitstreams.mcs"
```

```
Sector 0x000E0000: C:/FPGA-TN-02145/Bitstream_2.bit.
Sector 0x00FFFF00: JUMP: 0x000E0000.
MultiBoot Golden Sectors:  Start 0x000E0000 End 0x001A0000

Reading Alternate Pattern File: C:\FPGA-TN-02145\Bitstream_3.bit
Reading Alternate Pattern File: C:\FPGA-TN-02145\Bitstream_4.bit
Format: Intel Hex
Output File: C:/FPGA-TN-02145/Multi-Boot_4_Bitstreams.mcs
Sector 0x001C0000: C:\FPGA-TN-02145\Bitstream_3.bit.
Sector 0x002B0000: C:\FPGA-TN-02145\Bitstream_4.bit.
Start generation.
Generating Intel Hex PROM File.....
Successfully generate file(s)
C:\FPGA-TN-02145\Multi-Boot_4_Bitstreams.mcs

Lattice Radiant Deployment Tool has exited successfully.
```

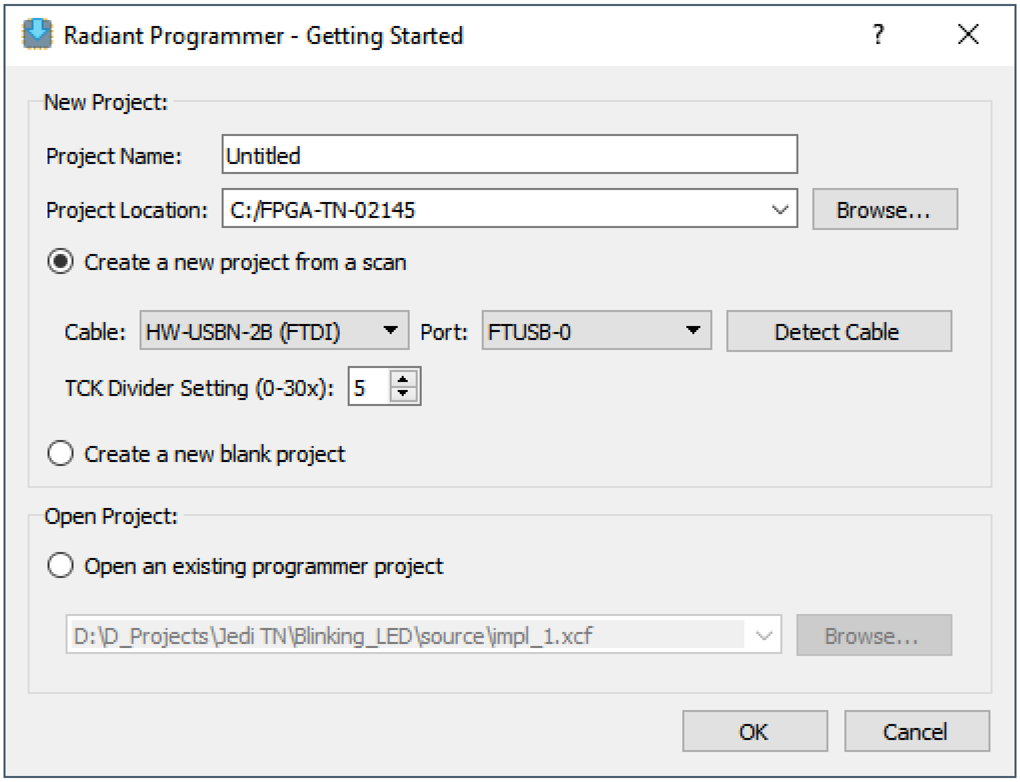
Figure 6.16. Generate Deployment Window

7. Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the External SPI Flash Device

The following procedure is for programming a Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the SPI Flash Device using Radiant Programmer:

1. Connect power to the board and connect a download cable from the board to the PC.
2. Invoke Radiant Programmer using one of the following methods:
   - In Radiant Software window, select **Tools > Programmer**;
   - In Radiant Software window, select the **Programmer** icon () in the Radiant toolbar;
   - In the Windows Start menu, select **Start > Lattice Radiant Programmer > Radiant Programmer**;
   - In the Windows Start menu, select **Start > Lattice Radiant Software > Radiant Programmer**.
3. Radiant Programmer – Getting Started window opens (Figure 7.1).
   - Select **Create a New Project from a Scan**, or **Create a new blank project**, or Select **Open an existing programmer project**.
   - Select **Detect Cable** to scan the PC to determine what cable is connected. Or, manually select the type of Cable and Port.
   - Select **OK**.
4. Select the Operation field by moving the cursor over it and double clicking the left mouse button.
5. The **Device Properties** window opens (Figure 7.2).
   - For **Target Memory**, select **External SPI Flash Memory (SPI Flash)**.
   - For **Port Interface**, select **JTAG2SPI**.

     CrossLink-NX/Certus-NX/CertusPro-NX and Radiant Programmer automatically takes care of the details to connect the JTAG port pins to the SPI interface pins and to program the external SPI Flash device via the JTAG port.
   - For **Access Mode**, select **Direct Programming**.
   - For **Operation**, select **Erase, Program, Verify**.
   - For Programming File, browse to select the .mcs file.
   - In the **SPI Flash Options** field, specify the **Family**, **Vendor**, **Device**, and **Package** of the Flash device used on the board.
   - For **Data File Size (Bytes)**, click on the **Load from File** button.
   - Click the **OK** button.
6. Program the external Flash device with one of the following methods:
   - In the **Radiant Programmer** window, select **Run > Program Device**.
   - In the **Radiant Programmer** window, click on the **Program** Device icon () in the toolbar.



Radiant Programmer - Getting Started

New Project:

Project Name: Untitled

Project Location: C:/FPGA-TN-02145

● Create a new project from a scan

Cable: HW-USBN-2B (FTDI) Port: FTUSB-0

TCK Divider Setting (0-30x): 5

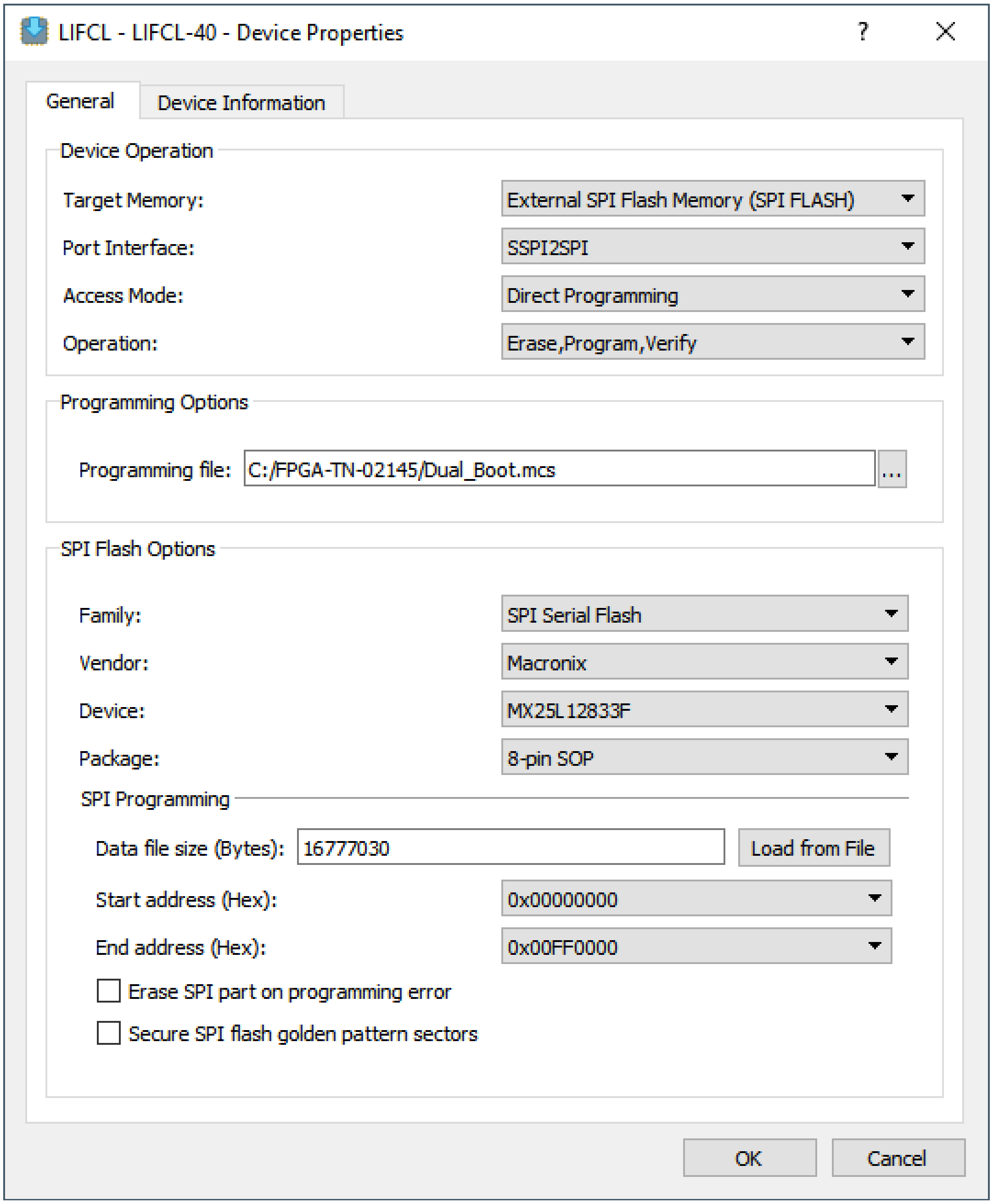
○ Create a new blank project

Open Project:

○ Open an existing programmer project

D:\D_Projects\Jedi TN\Blinking_LED\source\impl_1.xcf

Figure 7.1. Radiant Programmer – Getting Started Window



LIFCL - LIFCL-40 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: SSPI2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: C:/FPGA-TN-02145/Dual_Boot.mcs

SPI Flash Options

Family: SPI Serial Flash

Vendor: Macronix

Device: MX25L12833F

Package: 8-pin SOP

SPI Programming

Data file size (Bytes): 16777030

Start address (Hex): 0x00000000

End address (Hex): 0x00FF0000

□ Erase SPI part on programming error

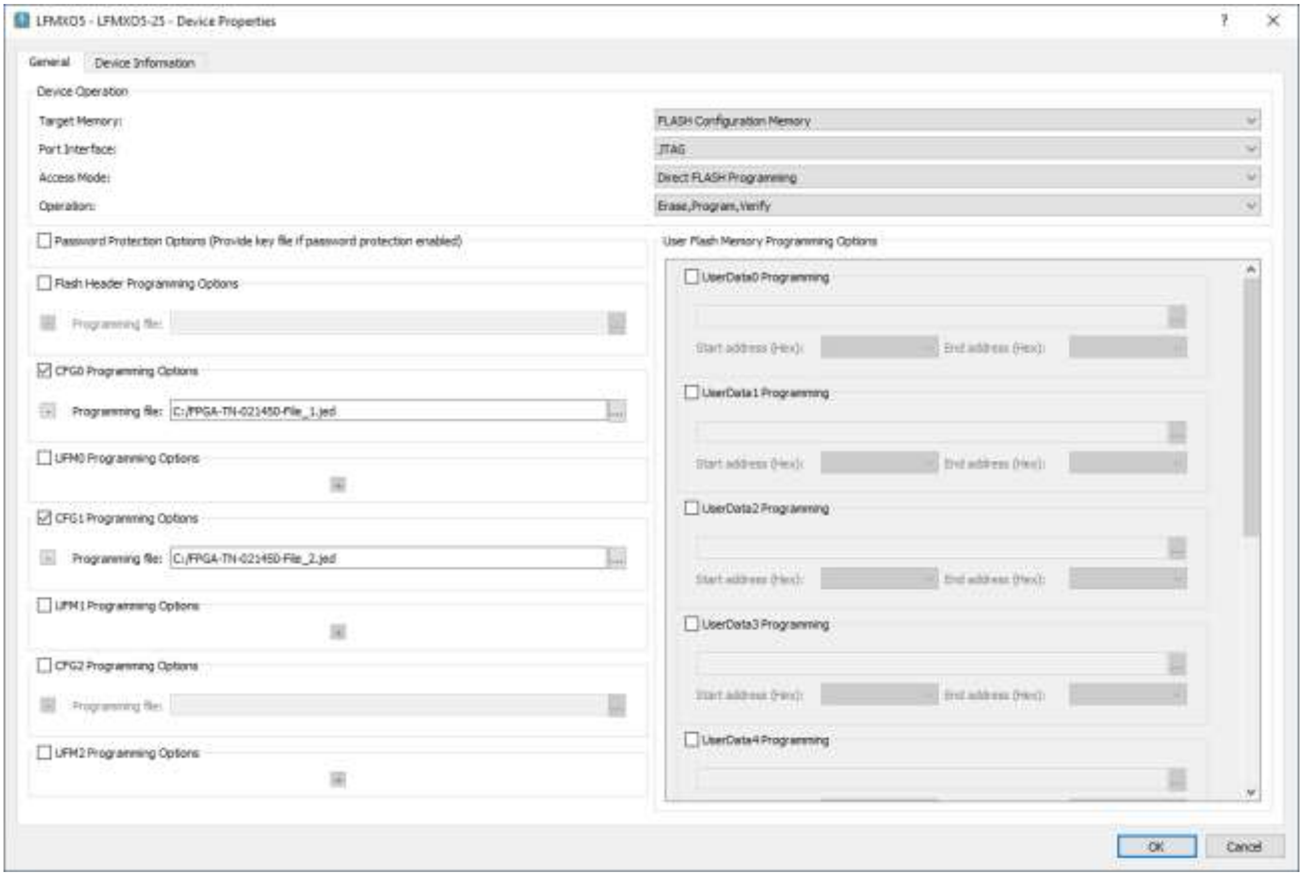
□ Secure SPI flash golden pattern sectors

Figure 7.2. Radiant Programmer – Device Properties Window

8. Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the Embedded Flash

The following procedure is for programming a Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the embedded Flash, such as MachXO5 device, using Radiant Programmer:

1. Connect power to the board and connect a download cable from the board to the PC.
2. Invoke Radiant Programmer using one of the following methods:
   - In Radiant Software window, select **Tools > Programmer**;
   - In Radiant Software window, select the **Programmer** icon () in the Radiant toolbar;
   - In the Windows Start menu, select **Start > Lattice Radiant Programmer > Radiant Programmer**;
   - In the Windows Start menu, select **Start > Lattice Radiant Software > Radiant Programmer**.
3. **Radiant Programmer – Getting Started** window opens (Figure 7.1).
   - Select **Create a New Project from a Scan**, or **Create a new blank project**, or **Select Open an existing programmer project**.
   - Select **Detect Cable** to scan the PC to determine what cable is connected. Or, manually select the type of Cable and Port.
   - Click **OK**.
4. Select the Operation field by moving the cursor over it and double clicking the left mouse button.
5. The **Device Properties** window opens (Figure 8.1).
   - For **Target Memory**, select **Flash Configuration Memory**.
   - For **Port Interface**, select **JTAG**.
   - For **Access Mode**, select **Direct FLASH Programming**.
   - For **Operation**, select **Erase, Program, Verify**.
   - For **CFGx Programming Files**, browse to select the .jed files.
   - Click the **OK** button.
   - Refer to MachXO5-NX Programming and Configuration User Guide (FPGA-TN-02271) for designation of primary boot and secondary boot.
6. Program the embedded Flash device with one of the following methods:
   - In the Radiant Programmer window, select **Run > Program** Device.
   - In the Radiant Programmer window, click on the **Program** Device icon () in the toolbar.



LFMXO5 - LFMXO5-25 - Device Properties

General | Device Information

Device Operation

Target Memory: FLASH Configuration Memory

Port Interface: JTAG

Access Mode: Direct FLASH Programming

Operation: Erase,Program,Verify

□ Password Protection Options (Provide key file if password protection enabled)

□ Flash Header Programming Options

Programming file:

☑ CFG0 Programming Options

Programming file: C:/FPGA-TN-021450-File_1.jed

□ UFM0 Programming Options

☑ CFG1 Programming Options

Programming file: C:/FPGA-TN-021450-File_2.jed

□ UFM1 Programming Options

□ CFG2 Programming Options

Programming file:

□ UFM2 Programming Options

User Flash Memory Programming Options

□ UserData0 Programming

Start address (Hex): End address (Hex):

□ UserData1 Programming

Start address (Hex): End address (Hex):

□ UserData2 Programming

Start address (Hex): End address (Hex):

□ UserData3 Programming

Start address (Hex): End address (Hex):

□ UserData4 Programming

Figure 8.1. Radiant Programmer – Device Properties Window

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 1.5, March 2023

Section	Change Summary
Introduction	Added support to the LFMXO5-55T and LFMXO5-100T parts in the description.
Resources	• Removed MachXO5-NX from the SRAM support. • Added embedded flash support for different configuration boot modes. • Added LFMXO5-55T and LFMXO5-100T device support in Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode, Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode/Ping-Pong Mode, and Table 2.3. Maximum Configuration Bitstream Size – Multi-Boot Mode.
Creating a PROM File	Removed MachXO5-NX family from the various boot features support.
Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the External SPI Flash Device	• Updated the section title adding external. • Removed the MachXO5-NX family support from this section.
Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the Embedded Flash	Newly added section.
Technical Support Assistance	Added the frequently asked questions website link.

Revision 1.4, June 2022

Section	Change Summary
Introduction	Added CrossLink-NX-33 (LIFCL-33) device support.
Resources	• Added CrossLink-NX-33 device support. • Added CrossLink-NX-33 (LIFCL-33) device and its related data to Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode, Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode/Ping-Pong Mode, and Table 2.3. Maximum Configuration Bitstream Size – Multi-Boot Mode.

Revision 1.3, March 2022

Section	Change Summary
All	Changed the document title to *Multi-Boot User Guide for Nexus Platform*.
Introduction	Added MachXO5-NX family support.
Resources	• Added MachXO5-NX family support. • Added MachXO5-NX device and its related data to Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode, Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode/Ping-Pong Mode, and Table 2.3. Maximum Configuration Bitstream Size – Multi-Boot Mode.
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Mode	• Added MachXO5-NX device family support. • Updated Figure 3.1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Flow Diagram changing to Internal/External Flash.
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Mode	• Added MachXO5-NX device family support. • Updated Figure 4.1. CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Flow Diagram changing to Internal/External Flash.
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Multi-Boot Mode	Added MachXO5-NX family support.
Creating a PROM File	

Section	Change Summary
Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the External SPI Flash Device	

Revision 1.2, May 2021

Section	Change Summary
Introduction	Added support for CertusPro-NX device family.
Resources	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Mode	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Mode	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Multi-Boot Mode	
Creating a PROM File	
Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the External SPI Flash Device	
Resources	Added resources details for CertusPro-NX device family to Table 2.1, Table 2.2, and Table 2.3.

Revision 1.1, May 2020

Section	Change Summary
All	Changed the document title to “Multi-Boot Usage Guide for Nexus Platform”.
Introduction	Added support for the Nexus platform including Certus-NX and CrossLink-NX device families.
Resources	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Dual Boot Mode	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Ping-Pong Boot Mode	
CrossLink-NX/Certus-NX/CertusPro-NX/MachXO5-NX Multi-Boot Mode	
Creating a PROM File	
Programming the Dual Boot, Ping-Pong Boot, or Multi-Boot Pattern into the External SPI Flash Device	
Resources	Added resources details for Certus-NX device family in Table 2.1, Table 2.2, and Table 2.3.

Revision 1.0, January 2020

Section	Change Summary
All	Initial release.

LATTICE
SEMICONDUCTOR®

www.latticesemi.com