Lattice Avant Multi-Boot User Guide

Preliminary

Technical Note

FPGA-TN-02314-0.82

November 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 all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. 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. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. 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.

Inclusive Language

This document was created consistent with Lattice Semiconductor's inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice's inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Glossary

A list 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.
CRC	Cyclic Redundancy Check
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.
GUI	Graphic User Interface.
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.
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).
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

Lattice Avant™ is a low-power mid-range density FPGA platform optimized for a wide range of applications across multiple markets — optimized for edge computing workloads requiring large memories and DSP resources and delivering a variety of high bandwidth interfaces ideal for video processing, communications, and machine learning inferencing. With Lattice low-power FPGA architecture, Avant devices deliver best-in-class power efficiency while meeting performance requirements for a wide range of applications.

The Avant platform supports various booting options for loading the configuration SRAM from a non-volatile memory for configuration flexibility and fail-safe configuration. Avant devices require an external memory for storage of the configuration bitstreams.

Multiple configuration boot modes mitigate risk during the field upgrade process and allow flexibility of executing different patterns. Field upgrade disruptions may occur due to power disruption, communication interruption or bitstream pattern corruption. The Avant platform supports the following boot modes:

- Single Bitstream Boot mode.
- 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 — Enables the device to dynamically switch from more than two bitstream patterns, inclusive of the Primary and Golden patterns.

The Avant platform combines multiple bitstream patterns into a single boot image stored in a single external SPI Flash device. This solution decreases cost, reduces board space, and simplifies field upgrades.

2. Resources

The Avant family is an SRAM-based FPGA. The volatile SRAM configuration memory must be loaded from an external non-volatile memory that can store all the configuration data. The size of the configuration data is based on the amount of logic available in the FPGA and number of pre-initialized Embedded Block RAM (EBR) components. An Avant design using the largest device, with every EBR pre-initialized with unique data values and generated without compression requires the largest amount of storage. The minimum SPI Flash densities required to support the different configuration boot modes are listed in Table 2.1 and Table 2.2.

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

Device	Configuration	Uncompressed[1]	SPI Mode
		Unencrypted/Encrypted Single Bitstream Size (Mb)	Minimum SPI Flash Size (Mb)
LAV-AT-E/G/X30	No EBR	44	64
	MAX EBR	58.4	64
LAV-AT-E/G/X50	No EBR	67	128
	MAX EBR	89.7	128
LAV-AT-E/G/X70	No EBR	107	128
	MAX EBR	142.6	256

Notes:

1. The Avant family of devices support bitstream compression. Compression ratio depends on the bitstream. Therefore, Table 2.1 only provides uncompressed bitstream data.
2. Both unencrypted and encrypted bitstream are the same size.

Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode

Device	Configuration	Uncompressed[1]		SPI Mode
		Unencrypted/Encrypted Single Bitstream Size (Mb)	Unencrypted/Encrypted Two Bitstreams Size (Mb)	Minimum SPI Flash Size (Mb)
LAV-AT-E/G/X30	No EBR	44	88	128
	MAX EBR	58.4	116.8	128
LAV-AT-E/G/X50	No EBR	67	134	256
	MAX EBR	89.7	179.4	256
LAV-AT-E/G/X70	No EBR	107	214	256
	MAX EBR	142.6	285.2	512

Notes:

1. The Avant family of devices support bitstream compression. Compression ratio depends on the bitstream. Therefore, Table 2.2 only provides uncompressed bitstream data.
2. Both unencrypted and encrypted bitstream are the same size.

3. Avant Dual Boot Mode

Avant Dual Boot mode supports booting from two configuration patterns that reside in an external SPI Flash device. One pattern is designated as the Primary pattern, and the second pattern is designated as the Golden pattern. When Avant device boots up, it attempts to boot from the Primary pattern. If loading of the Primary pattern fails, the Avant device boots from the Golden pattern.

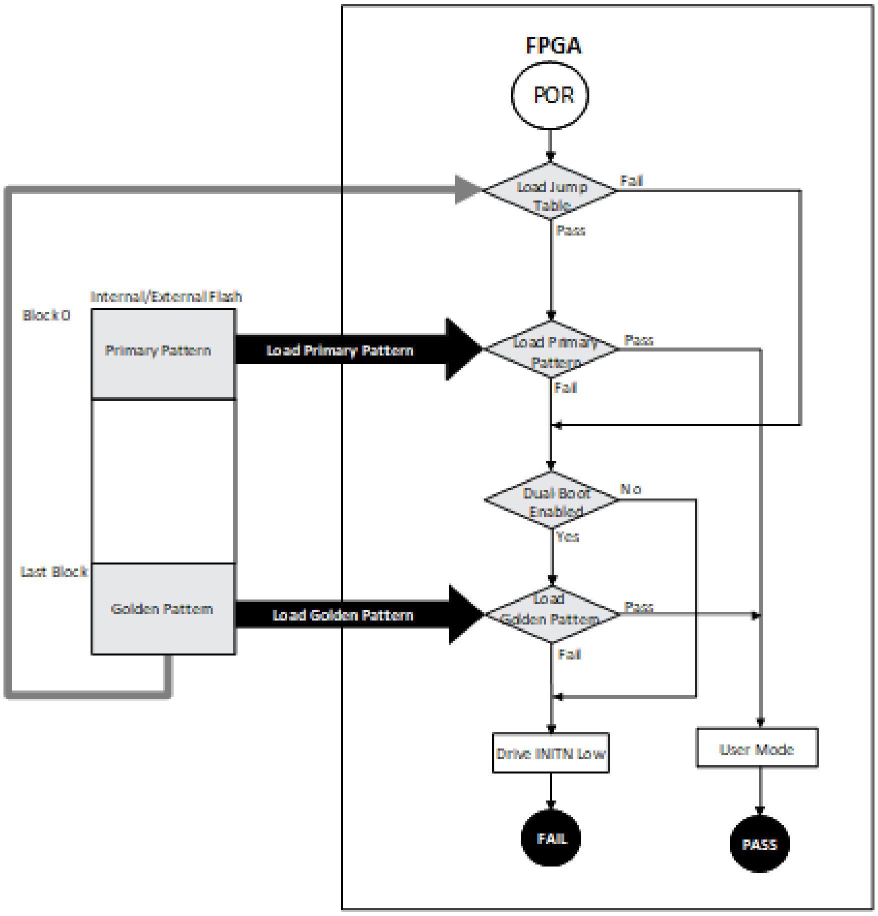


Figure 3.1. Avant Family Dual Boot Flow Diagram

3.1. Description of the Avant 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, the Primary pattern may fail to load for one of the following reasons:

- Time-Out Check — Avant device searches for the preamble code (Table 3.1) from the Primary Pattern as part of the configuration protocol failed.

Table 3.1. Bitstream Preamble Codes

Preamble Code	Note
0xFFFFBDB3	Bitstream file is unencrypted and unauthenticated.
0xFFFFBFB3	Bitstream file is encrypted and authenticated.
0xFFFFBEB3	Bitstream file is authenticated only.
0xFFFFBCB3	Bitstream file is used for authenticated control segment.
0xFFFFB0B3	Segment contains plain text BITSTREAM_BURST_DONE command.

- Device ID checked at the beginning of the bitstream failed.
- 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.
- An illegal command is encountered.
- Time-Out (Table 3.2) when loading the Primary pattern stored in the external Flash device.

Table 3.2. Control Register 1 [2:0] – Master Preamble Timer Count Value

CR1 [2:0]	Timer Value
0	200 ms
1	100 ms
2	50 ms
3	40 ms
4	20 ms
5	1 ms
6	500 µs
7	100 µs

If the Primary pattern fails one of the checks above, the Avant 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 loads the Golden pattern from the last block of the Flash device.

An Avant 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 Avant device stops driving the SPI clock, and the INITN pin is driven LOW.

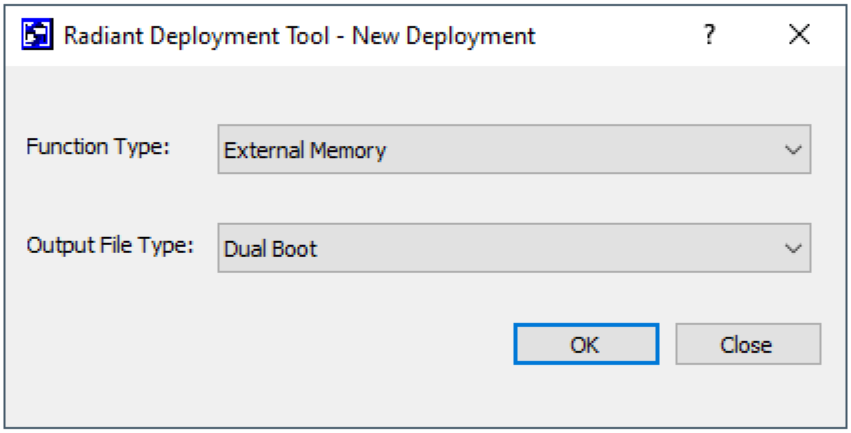
4. Creating a PROM File

Lattice Radiant Software provides a turnkey solution to manage the low-level implementation details of the dual-boot feature on the Avant family of devices. Lattice Deployment Tool, part of Lattice Radiant Software, merges the different patterns and the JUMP command into one PROM hex file. The PROM hex file can later be programmed into an external Flash device using Radiant Programmer or a third-party programmer.

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

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

- 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.
- Invoke Lattice Radiant Deployment Tool by choosing **Tools > Deployment Tool** from Lattice Radiant Programmer.
- In the Radiant Deployment Tool window, select **External Memory** as the **Function Type** and select **Dual Boot** as the **s** (Figure 4.1).
- Select **OK**.



Radiant Deployment Tool - New Deployment

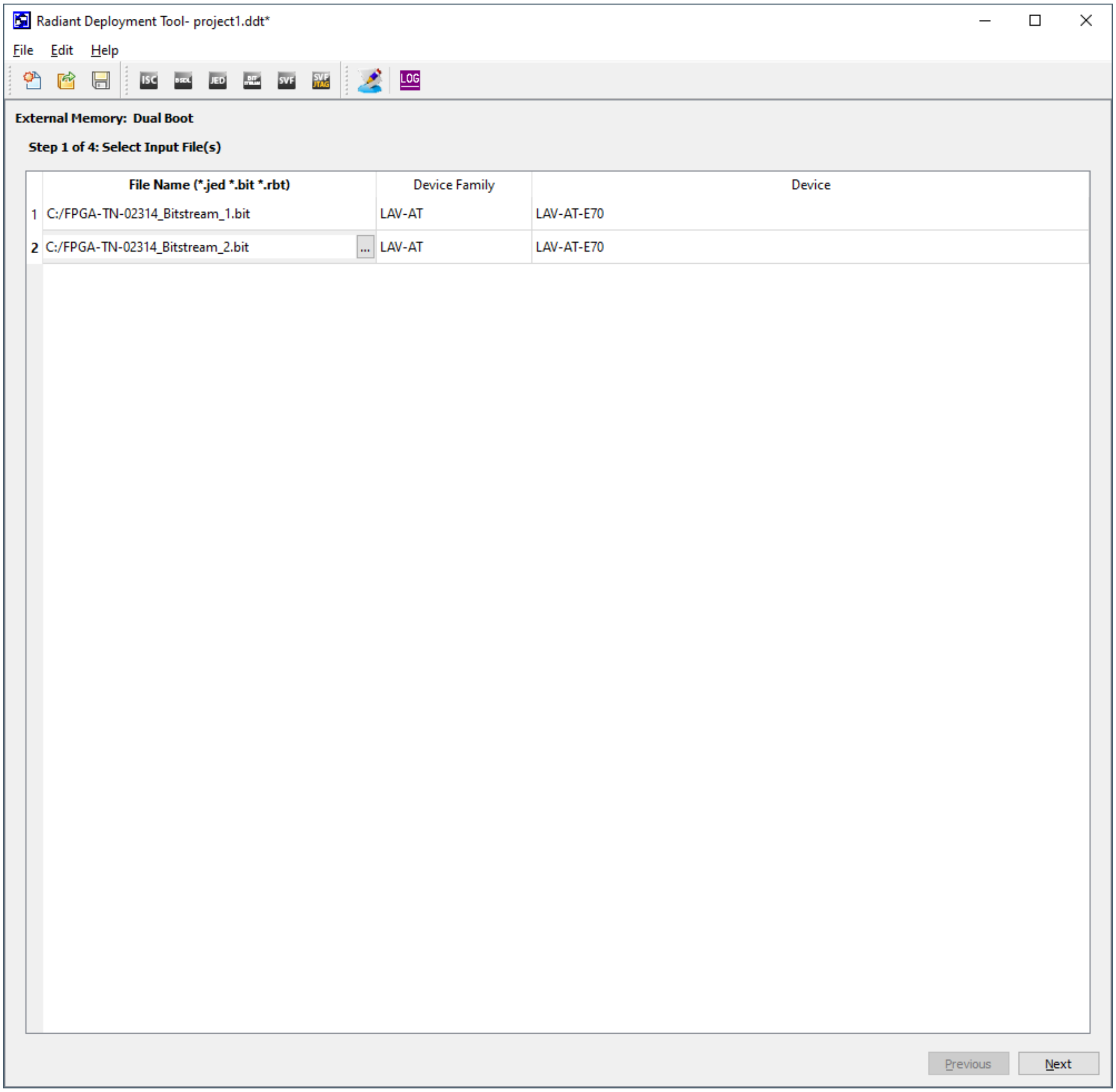
Function Type: External Memory

Output File Type: Dual Boot

Figure 4.1. Creating New Deployment for Dual Boot PROM Hex File

Step 1 of 4: Select Input File(s) window (Figure 4.2)

- 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-02314_Bitstream_1.bit	LAV-AT	LAV-AT-E70
2	C:/FPGA-TN-02314_Bitstream_2.bit	LAV-AT	LAV-AT-E70

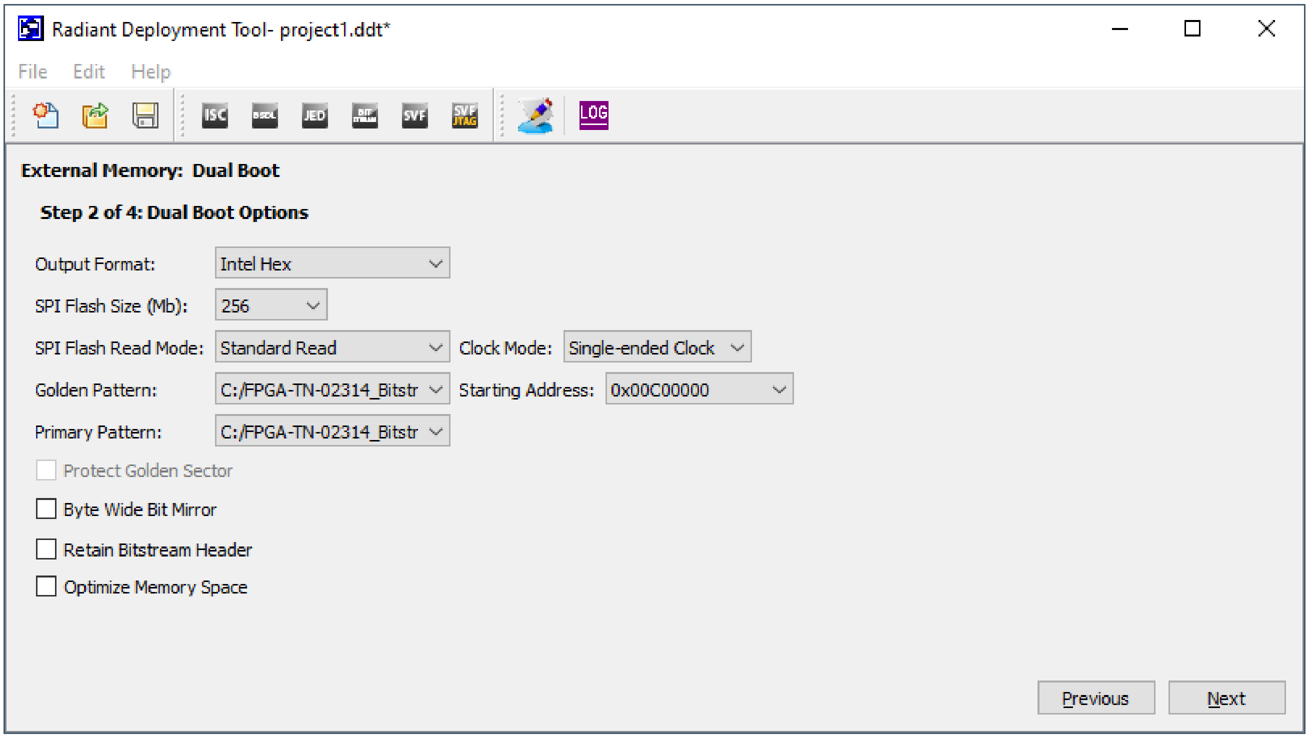
Figure 4.2. Select Input Files Window

Step 2 of 4: Dual Boot Options window (Figure 4.3)

- Select the **Output Format** (Intel Hex, Motorola Hex or Extended Tektronix Hex).
- Select the **SPI Flash Size (Mb)** (4, 8, 16, 32, 64, 128, 256, 512, or 1024 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) 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** – When checked, it uses the bitstream file size instead of the worst case bitstream size.
- Select **Next**.



External Memory: Dual Boot

Step 2 of 4: Dual Boot Options

Output Format: Intel Hex

SPI Flash Size (Mb): 256

SPI Flash Read Mode: Standard Read   Clock Mode: Single-ended Clock

Golden Pattern: C:/FPGA-TN-02314_Bitstr   Starting Address: 0x00C00000

Primary Pattern: C:/FPGA-TN-02314_Bitstr

□ Protect Golden Sector

□ Byte Wide Bit Mirror

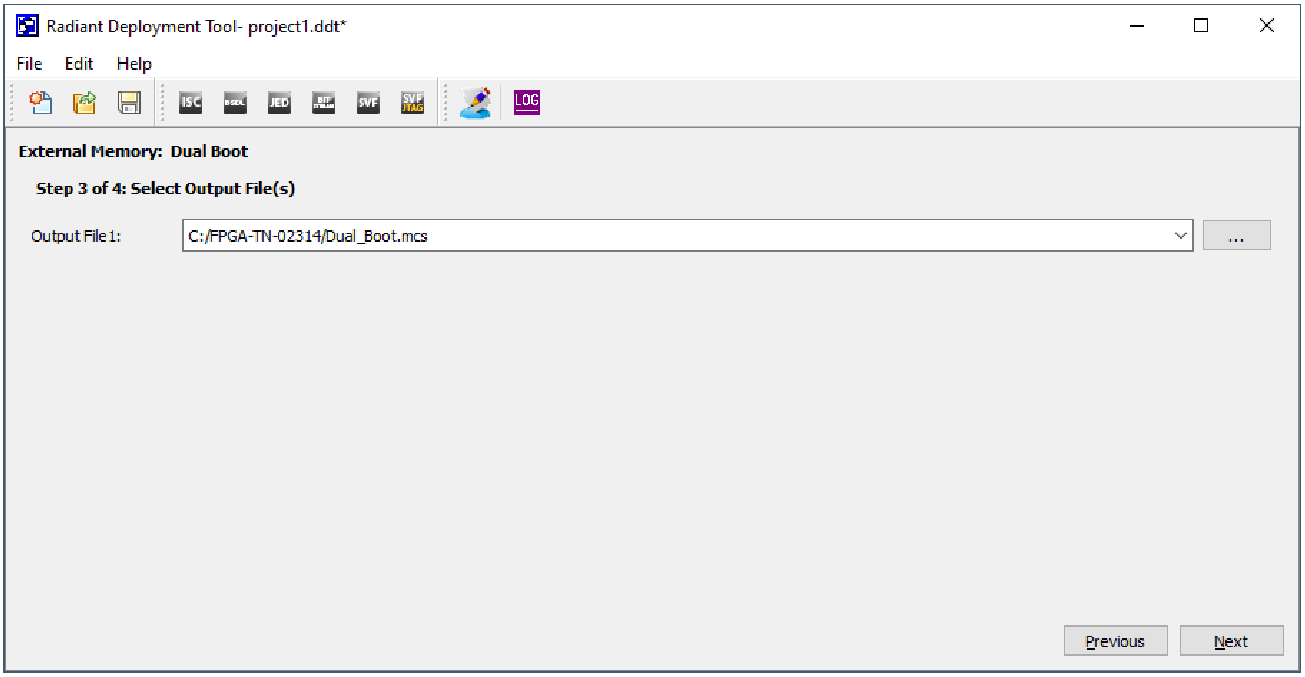
□ Retain Bitstream Header

□ Optimize Memory Space

Figure 4.3. Dual Boot Options Window

Step 3 of 4: Select Output File(s) window (Figure 4.4)

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



External Memory: Dual Boot

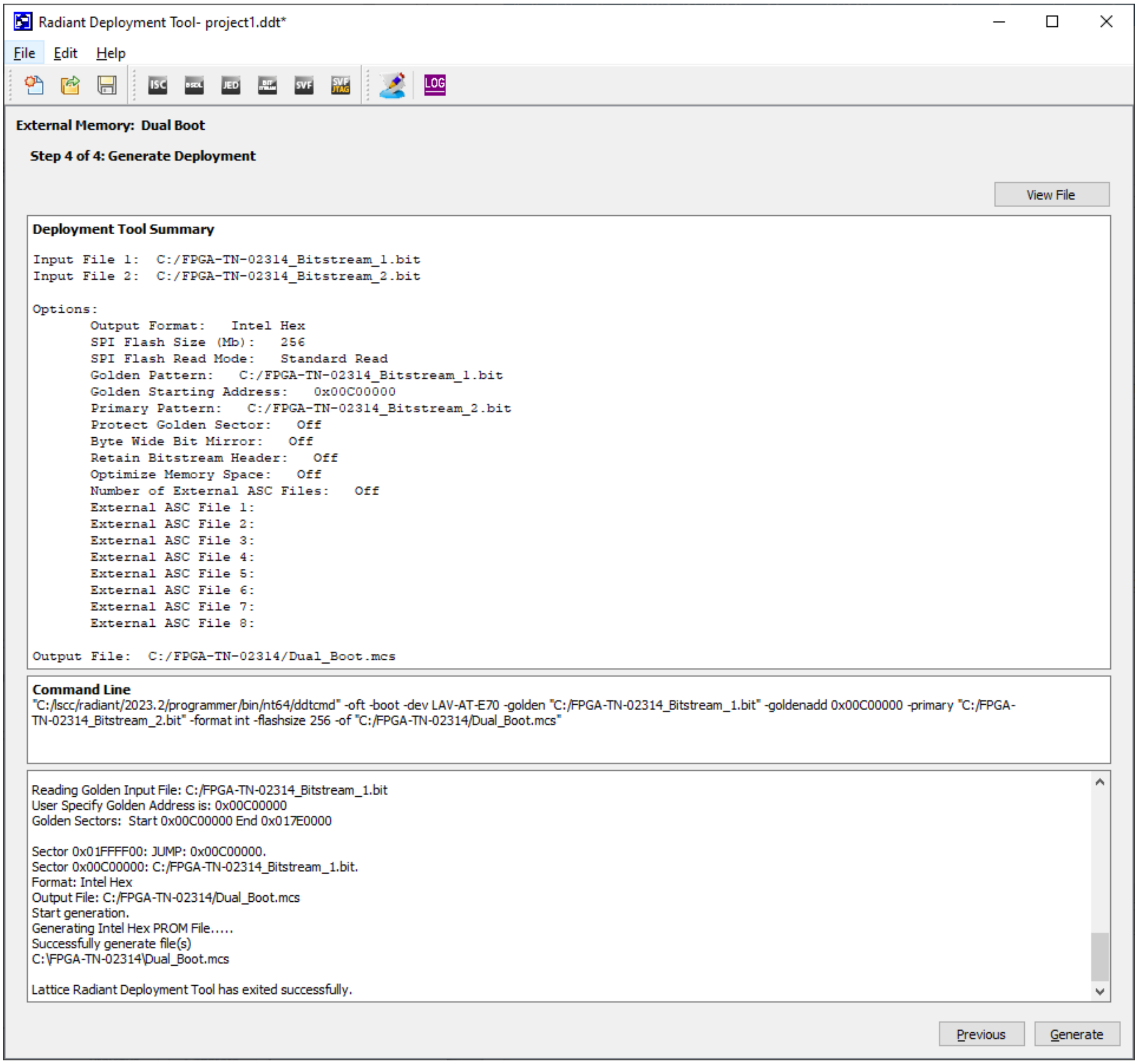
Step 3 of 4: Select Output File(s)

Output File 1: C:/FPGA-TN-02314/Dual_Boot.mcs

Figure 4.4. Select Output File Window

Step 4 of 4: Generate Deployment window (Figure 4.5)

- 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**.



External Memory: Dual Boot

Step 4 of 4: Generate Deployment

Deployment Tool Summary

```
Input File 1:  C:/FPGA-TN-02314_Bitstream_1.bit
Input File 2:  C:/FPGA-TN-02314_Bitstream_2.bit

Options:
        Output Format:   Intel Hex
        SPI Flash Size (Mb):   256
        SPI Flash Read Mode:   Standard Read
        Golden Pattern:   C:/FPGA-TN-02314_Bitstream_1.bit
        Golden Starting Address:   0x00C00000
        Primary Pattern:   C:/FPGA-TN-02314_Bitstream_2.bit
        Protect Golden Sector:   Off
        Byte Wide Bit Mirror:   Off
        Retain Bitstream Header:   Off
        Optimize Memory Space:   Off
        Number of External ASC Files:   Off
        External ASC File 1:
        External ASC File 2:
        External ASC File 3:
        External ASC File 4:
        External ASC File 5:
        External ASC File 6:
        External ASC File 7:
        External ASC File 8:

Output File:  C:/FPGA-TN-02314/Dual_Boot.mcs
```

Command Line

```
"C:/lscc/radiant/2023.2/programmer/bin/nt64/ddtcmd" -oft -boot -dev LAV-AT-E70 -golden "C:/FPGA-TN-02314_Bitstream_1.bit" -goldenadd 0x00C00000 -primary "C:/FPGA-TN-02314_Bitstream_2.bit" -format int -flashsize 256 -of "C:/FPGA-TN-02314/Dual_Boot.mcs"
```

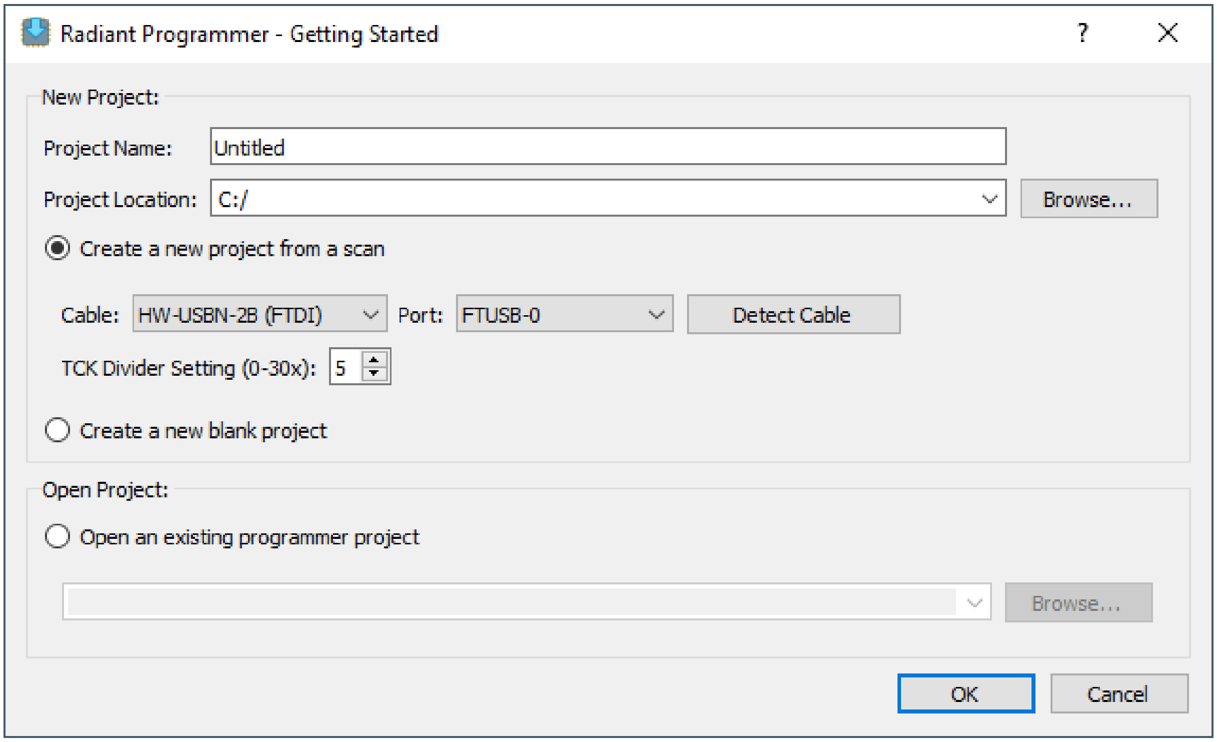
```
Reading Golden Input File: C:/FPGA-TN-02314_Bitstream_1.bit
User Specify Golden Address is: 0x00C00000
Golden Sectors:  Start 0x00C00000 End 0x017E0000

Sector 0x01FFFF00: JUMP: 0x00C00000.
Sector 0x00C00000: C:/FPGA-TN-02314_Bitstream_1.bit.
Format: Intel Hex
Output File: C:/FPGA-TN-02314/Dual_Boot.mcs
Start generation.
Generating Intel Hex PROM File.....
Successfully generate file(s)
C:\FPGA-TN-02314\Dual_Boot.mcs

Lattice Radiant Deployment Tool has exited successfully.
```

Figure 4.5. Generate Deployment Window

5. Programming the Dual Boot Pattern into the SPI Flash Device

The following procedure is for Programming a Dual 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 2022.1 or later 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 Software 2023.2 > Radiant Programmer**;
   - In the Windows Start menu, select **Start > Lattice Radiant Software 2023.2 > Radiant Software > Radiant Programmer**.
3. **Radiant Programmer – Getting Started** window opens (Figure 5.1).
   - 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. The **Radiant Programmer** main window prompts. In the **Operation** field, move the cursor over the highlighted operation such as Bypass and double-click the left mouse button.
5. The **Device Properties** window opens (Figure 5.2).
   - For **Target Memory**, select **External SPI Flash Memory (SPI Flash)**.
   - For **Port Interface**, select **JTAG2SPI**.
   - 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.
   - Select 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, select the **Program Device** icon ( ) in the toolbar.



Radiant Programmer - Getting Started

New Project:

Project Name: Untitled

Project Location: C:/

● Create a new project from a scan

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

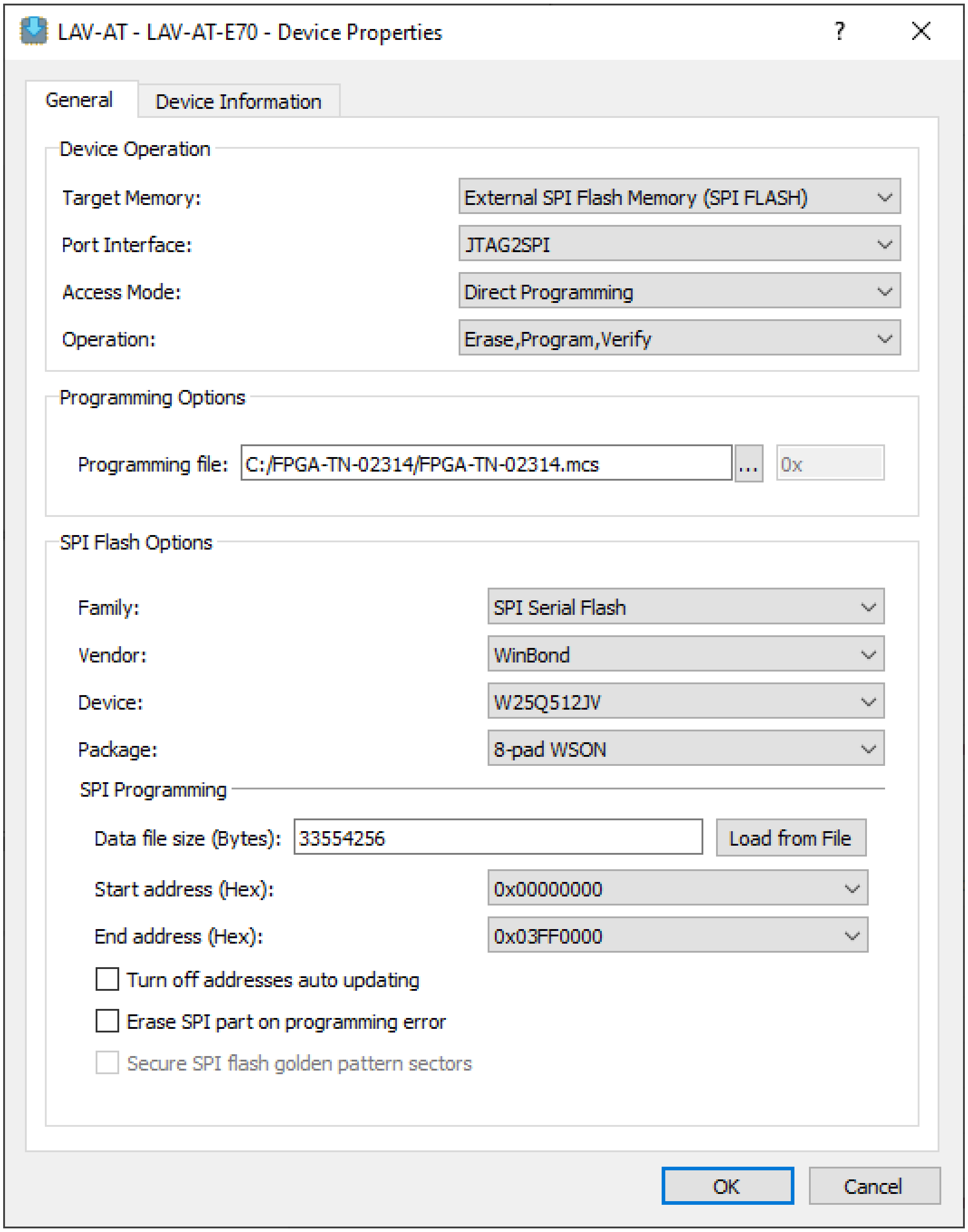
TCK Divider Setting (0-30x): 5

○ Create a new blank project

Open Project:

○ Open an existing programmer project

Figure 5.1. Radiant Programmer – Getting Started Window



LAV-AT - LAV-AT-E70 - Device Properties

Device Operation

Target Memory: External SPI Flash Memory (SPI FLASH)

Port Interface: JTAG2SPI

Access Mode: Direct Programming

Operation: Erase,Program,Verify

Programming Options

Programming file: C:/FPGA-TN-02314/FPGA-TN-02314.mcs ... 0x

SPI Flash Options

Family: SPI Serial Flash

Vendor: WinBond

Device: W25Q512JV

Package: 8-pad WSON

SPI Programming

Data file size (Bytes): 33554256 Load from File

Start address (Hex): 0x00000000

End address (Hex): 0x03FF0000

□ Turn off addresses auto updating

□ Erase SPI part on programming error

□ Secure SPI flash golden pattern sectors

Figure 5.2. Radiant Programmer – Device Properties

References

- Avant-E Web Page
- Avant-G Web Page
- Avant-X Web Page

A variety of technical notes for the Lattice Avant platform are available.

- High-Speed PCB Design Considerations (FPGA-TN-02178)
- Lattice Avant Embedded Memory User Guide (FPGA-TN-02289)
- Lattice Avant Hardware Checklist (FPGA-TN-02317)
- Lattice Avant High-Speed I/O and External Memory Interface (FPGA-TN-02300)
- Lattice Avant Power User Guide (FPGA-TN-02291)
- Lattice Avant sysCLOCK PLL Design and User Guide (FPGA-TN-02298)
- Lattice Avant sysCONFIG User Guide (FPGA-TN-02299)
- Lattice Avant sysDSP User Guide (FPGA-TN-02293)
- Lattice Avant sysI/O User Guide (FPGA-TN-02297)
- Lattice Memory Mapped Interface and Lattice Interrupt Interface User Guide (FPGA-UG-02039)
- sub-LVDS Signaling Using Lattice Devices (FPGA-TN-02028)
- Thermal Management (FPGA-TN-02044)
- Using TraceID (FPGA-TN-02084)

For more information on Lattice Avant-related IP, reference designs, and board documents, refer to the following pages:

- SGMII and Gb Ethernet PCS IP Core – Lattice Radiant Software (FPGA-IPUG-02077)
- Avant-E Evaluation Board – User Guide (FPGA-EB-02057)
- IP and Reference Designs for Avant
- Development Kits and Boards for Avant

For further information on interface standards refer to the following websites:

- JEDEC Standards (LVTTL, LVCMOS, SSTL) – www.jedec.org
- PCI – www.pcisig.com

Other references:

- Lattice Insights for Lattice Semiconductor training courses and learning plans
- Lattice Radiant FPGA design software

Technical Support Assistance

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

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

Revision History

Revision 0.82, November 2023

Section	Change Summary
Disclaimers	Updated this section.
Resources	Updated Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode and Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode reflecting the Avant Product name change to the current.
Creating a PROM File	Updated Figure 4.2. Select Input Files Window and Figure 4.5. Generate Deployment Window reflecting the Avant product name change in the software GUI.
Programming the Dual Boot Pattern into the SPI Flash Device	• Updated the software version to *2023.2*. • Updated Figure 5.2. Radiant Programmer – Device Properties reflecting the Avant product name change in the software GUI.
References	Newly added section.

Revision 0.81.1, May 2023

Section	Change Summary
Resources	Updated Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode and Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode adding the data for Lattice Avant G/X devices.
Avant Dual Boot Mode	Updated the whole section including the newly added Table 3.1. Bitstream Preamble Codes and Table 3.2. Control Register 1 [2:0] – Master Preamble Timer Count Value reflecting the recent Avant device behavior.
Creating a PROM File	Updated the general description removing the mentioning of external SPI Flash.
Programming the Dual Boot Pattern into the SPI Flash Device	Updated procedure upon the most recent Lattice Radiant Programmer software.

Revision 0.81, April 2023

Section	Change Summary
Inclusive Language	Newly added section.
Resources	Updated Table 2.1. Maximum Configuration Bitstream Size – Single Bitstream Boot Mode and Table 2.2. Maximum Configuration Bitstream Size – Dual Boot Mode adding the data for Lattice Avant 200E, 300E and the G/X devices.
Avant Dual Boot Mode	Updated the whole section including the newly added Table 3.1. Bitstream Preamble Codes and Table 3.2. Control Register 1 [2:0] – Master Preamble Timer Count Value reflecting the recent Avant device behavior.
Creating a PROM File	Updated the general description removing the mentioning of external SPI Flash.
Programming the Dual Boot Pattern into the SPI Flash Device	Updated procedure upon the most recent Lattice Radiant Programmer software.

Revision 0.80, November 2022

Section	Change Summary
All	Initial Preliminary release.

www.latticesemi.com