Lattice Nexus 2 Multi-Boot User Guide

Preliminary Technical Note

FPGA-TN-02385-0.80

December 2024

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 REDUNDANCIES, 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

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
CRC	Cyclic Redundancy Check
EBR	Embedded Block RAM
MSPI	Controller Serial Peripheral Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
xSPI	Expanded Serial Peripheral Interface

1. Introduction

The Lattice Nexus™ 2 platform supports various booting options for loading configuration SRAM from non-volatile memory. These booting options provide configuration flexibility and facilitate fail-safe configuration. Certus™-N2 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 might occur due to power disruption, communication interruption, or bitstream pattern corruption. The Nexus 2 platform supports the following boot modes:

- Single bitstream boot mode
- Dual-boot mode
- Ping-pong boot mode
- Multi-boot mode

The Nexus 2 platform combines multiple bitstream patterns into a single boot image stored in a single external SPI storage device. This solution decreases cost, reduces board space, and simplifies field upgrades.

2. Definition of Terms

Table 2.1 lists the terms used in this document to describe common functions, features, or concepts.

Table 2.1. Definition of Terms

Term	Definition
Alternative Boot Pattern	In multi-boot mode, after the FPGA device has been configured, this pattern is loaded after PROGRAMN pin pulsing or REFRESH command execution. Up to 16 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.
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 format.
Configuration	A change in the state of the SRAM memory cells.
Dual-Boot	This feature allows the FPGA device to support two configuration images that reside in an SPI storage device. Whenever loading failure occurs with the primary image, the FPGA device searches for and loads the secondary image. Both images come from non-volatile SPI memory.
Golden Pattern	The guaranteed good pattern loaded into the FPGA device when booting failure occurs. Only one Golden pattern is allowed.
Hex Data File, Hex File (.exo, .mcs, .xtek Files)	A data record file commonly in a format such as Intel Hex, Motorola Hex, or Extended Tektronix Hex. This file is also known as an addressed record file. It is used for programming configuration memory or flash devices.
Multi-Boot	The FPGA device determines and triggers the loading of the next pattern after a prior successful configuration. Multiple patterns (that is, two patterns or more) are available for the FPGA device to choose to load on demand. All patterns are stored in non-volatile SPI memory.
Ping-Pong Boot	This feature allows the FPGA device to utilize the jump table to select a pattern for booting without changing the location of the pattern in the SPI storage device.
Primary Pattern	At power up, the FPGA device loads this pattern first. Only one Primary pattern is allowed.
Programming	The process used to alter the contents of the external configuration memory.
Refresh	The process of triggering a configuration data load operation. It is activated by PROGRAMN pin pulsing or REFRESH command execution (which emulates PROGRAMN pin pulsing).

3. Resources

Nexus 2 devices are SRAM-based FPGAs. The 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 varies. It is dependent on the amount of logic available in the FPGA and the number of pre-initialized Embedded Block RAM (EBR) components. A design using the largest Nexus 2 device, with every EBR pre-initialized with unique data values and generated without compression enabled, requires the largest amount of storage.

Table 3.1. Bitstream Size versus Recommended SPI Storage Size (Single Bitstream Boot Mode)

Device	Scenario	Bitstream Size (Mb)[1]	Recommended SPI Storage Device Size (Mb)
LN2-CT-20	No EBR	44	64
	Maximum EBR	58.4	64

Note:

1. Both unencrypted and encrypted bitstreams are the same size. Bitstream compression ratio varies depending on the bitstream so only uncompressed bitstream sizes are shown.

Table 3.2. Bitstream Size versus Recommended SPI Storage Size (Dual-Boot Mode, Ping-Pong Boot Mode)

Device	Scenario	Bitstream Size (Mb)[1]		Recommended SPI Storage Device Size (Mb)
		1 Bitstream Pattern	2 Bitstream Patterns	
LN2-CT-20	No EBR	44	88	128
	Maximum EBR	58.4	116.8	128

Note:

1. Both unencrypted and encrypted bitstreams are the same size. Bitstream compression ratio varies depending on the bitstream so only uncompressed bitstream sizes are shown.

Table 3.3. Bitstream Size versus Recommended SPI Storage Size (Multi-Boot Mode)

Device	Scenario	Bitstream Size (Mb)[1]					Recommended SPI Storage Device Size (Mb)			
		1 Bitstream Pattern	3 Bitstream Patterns	4 Bitstream Patterns	5 Bitstream Patterns	6 Bitstream Patterns	3 Bitstream Patterns	4 Bitstream Patterns	5 Bitstream Patterns	6 Bitstream Patterns
LN2-CT-20	No EBR	44	132	176	220	264	256	256	256	512
	Maximum EBR	58.4	175.2	233.6	292	350.4	256	256	512	512

Note:

1. Both unencrypted and encrypted bitstreams are the same size. Bitstream compression ratio varies depending on the bitstream so only uncompressed bitstream sizes are shown.

4. Dual-Boot Mode

The Nexus 2 device dual-boot mode supports booting from two configuration bitstream patterns that reside in an external SPI storage device. These patterns are designated the primary pattern and golden pattern. When a Nexus 2 device boots up, it attempts to boot from the primary pattern. If primary pattern fails to load, the device attempts to boot from the golden pattern. Figure 4.1 shows the dual-boot flow diagram.

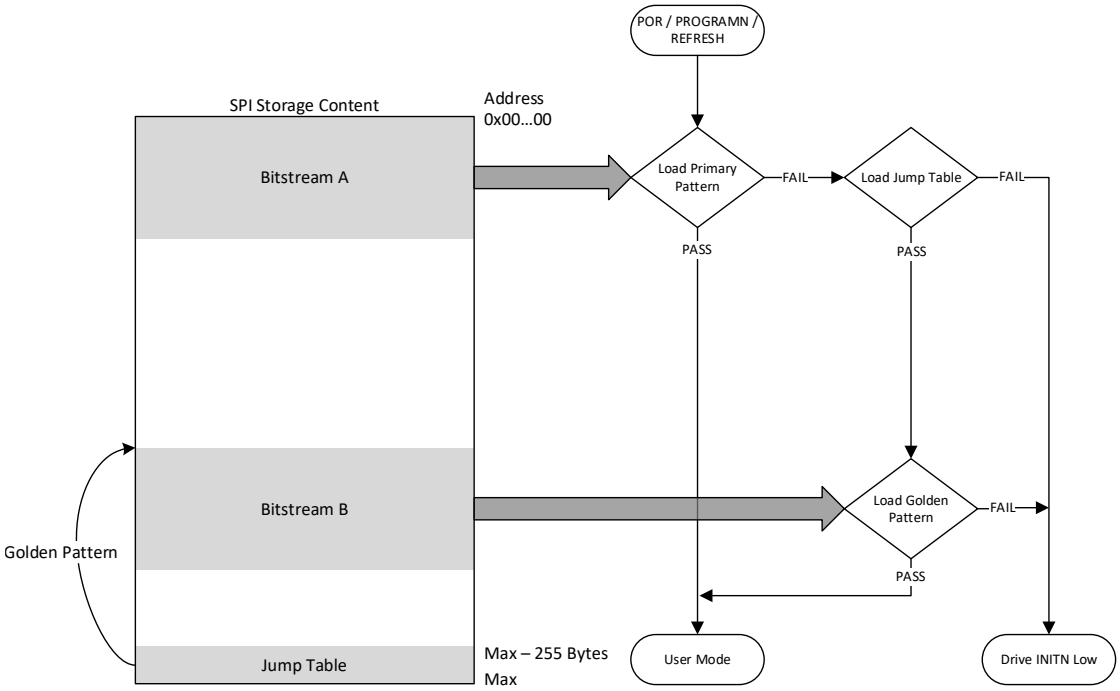


Figure 4.1. Dual-Boot Flow Diagram

4.1. Dual-Boot Flow

The dual-boot mode flow is triggered either at power up, PROGRAMN pin pulsing, or REFRESH command execution.

In dual-boot mode, the primary pattern might fail to load for one of the following reasons:

- Time-out check — Search for a 32-bit preamble code (0xFFFFBxB3) in the primary pattern, as part of the configuration process, fails.
- Device ID check — ID check at the beginning of the bitstream fails.
- Data corruption check — CRC checks detect data corruption in the bitstream.
- An illegal command is encountered.

Table 4.1 shows the time-out period when loading the primary pattern stored in the external SPI storage device.

Table 4.1. Control Register 1 [2:0] – MSPI Preamble Timer Count

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

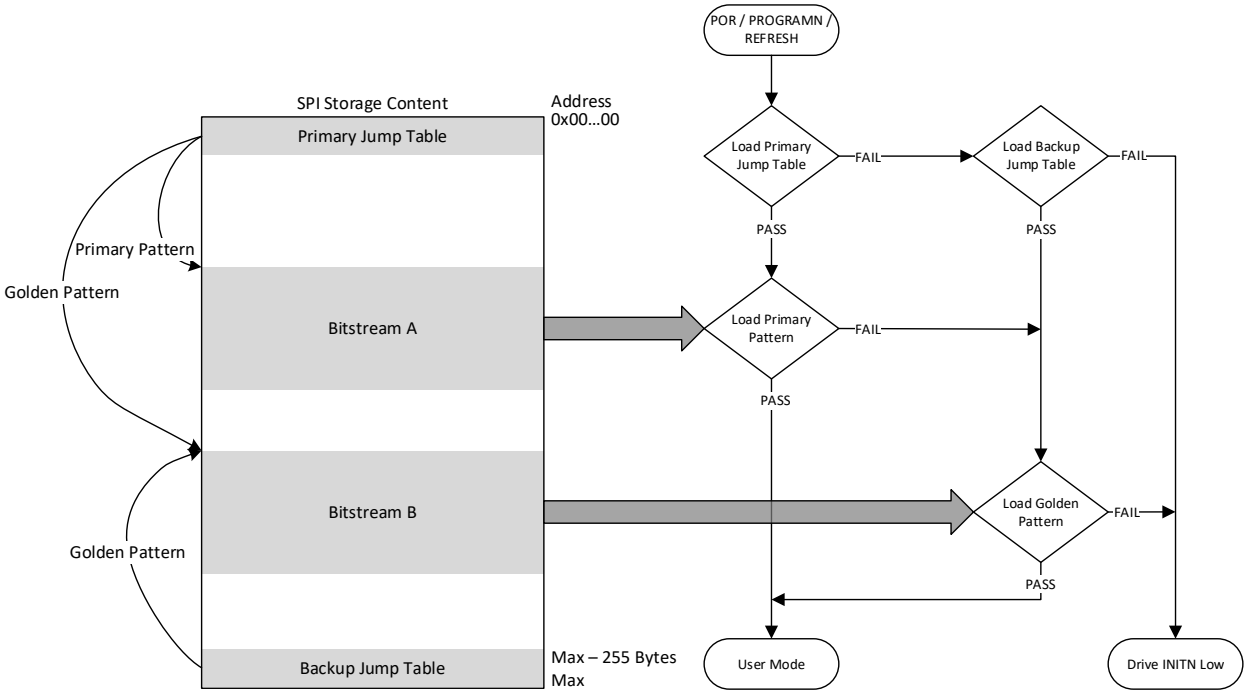
If the primary pattern fails to load, the Nexus 2 device drives the INITN pin low to indicate an error and resets the configuration engine. After clearing the configuration RAM, the device drives the INITN pin high and reads the JUMP command in the jump table that directs it to load the golden pattern in the SPI storage device.

Note: The jump table is located at the last 256 bytes of the SPI storage device. When the configuration engine attempts to read the jump table, it issues a read offset address at 0xFFFF00 for 24-bit SPI flash, 0xFFFF_FF00 for 32-bit SPI flash, or 0xFFFF_FFFF_FF00 for 45-bit xSPI flash. The SPI flash truncates the most significant bit or bits of the address depending on the density of the SPI flash. For example, a 32-Mb SPI flash receives the read offset address as 0x3FFF00 even though the configuration engine sends 0xFFFF00.

The Nexus 2 device performs the same checks for the golden pattern. If the golden pattern is also corrupted, configuration fails. The Nexus 2 device stops driving the SPI clock and drives the INITN pin low.

5. Ping-Pong Boot Mode

The Nexus 2 device ping-pong boot mode supports booting from one of two configuration bitstream patterns that reside in an external SPI storage device. These patterns are designated the primary bitstream and secondary bitstream. The Nexus 2 device boots from the bitstream pattern based on the assignment in the jump table. The jump table allows the device to boot from either bitstream without changing the physical location of the two bitstreams within the external SPI storage device. Only the jump table needs to be updated to change the boot bitstream pattern. When one bitstream is selected as the primary bitstream pattern, by default, the secondary bitstream becomes the golden bitstream pattern. The external SPI storage device can also store a backup jump table, in case the primary jump table becomes corrupted. Figure 5.1 shows the ping-pong boot flow diagram.



Notes:
1. The primary jump table points to the addresses of the primary and golden patterns.
2. The backup jump table points to the golden pattern.
3. To change the designation of which bitstream is primary and which is golden, only the jump tables need to be updated.

Figure 5.1. Ping-Pong Boot Flow Diagram

5.1. Ping-Pong Boot Flow

The ping-pong boot flow is triggered at power up, PROGRAMN pin pulsing, or REFRESH command execution. When a JUMP command in the primary jump table is executed, the device stops the SPI clock, drives the INITN pin low, resets the configuration engine, and clears the configuration RAM. After the configuration RAM is cleared, the device drives the INITN pin high, restarts the SPI clock, and reads the primary pattern from the SPI storage device as addressed by the jump table.

In ping-pong boot mode, the bitstream pattern selected might fail to load for one of the following reasons:

- Time-out check — Search for a 32-bit preamble code (0xFFFFBxB3) in the bitstream pattern, as part of the configuration protocol, fails. Refer to Table 4.1 for the time-out values.
- Device ID check — ID check at the beginning of the bitstream fails.
- Data corruption check — CRC checks detect data corruption in the bitstream.
- An illegal command is encountered.

If the bitstream fails to load, the Nexus 2 device drives the INITN pin low to indicate an error and resets the configuration engine. After clearing the configuration RAM, the device drives the INITN pin high and reads the address of the golden pattern in the backup jump table. If the golden pattern configuration fails, the Nexus 2 device stops driving the SPI clock and drives the INITN pin low.

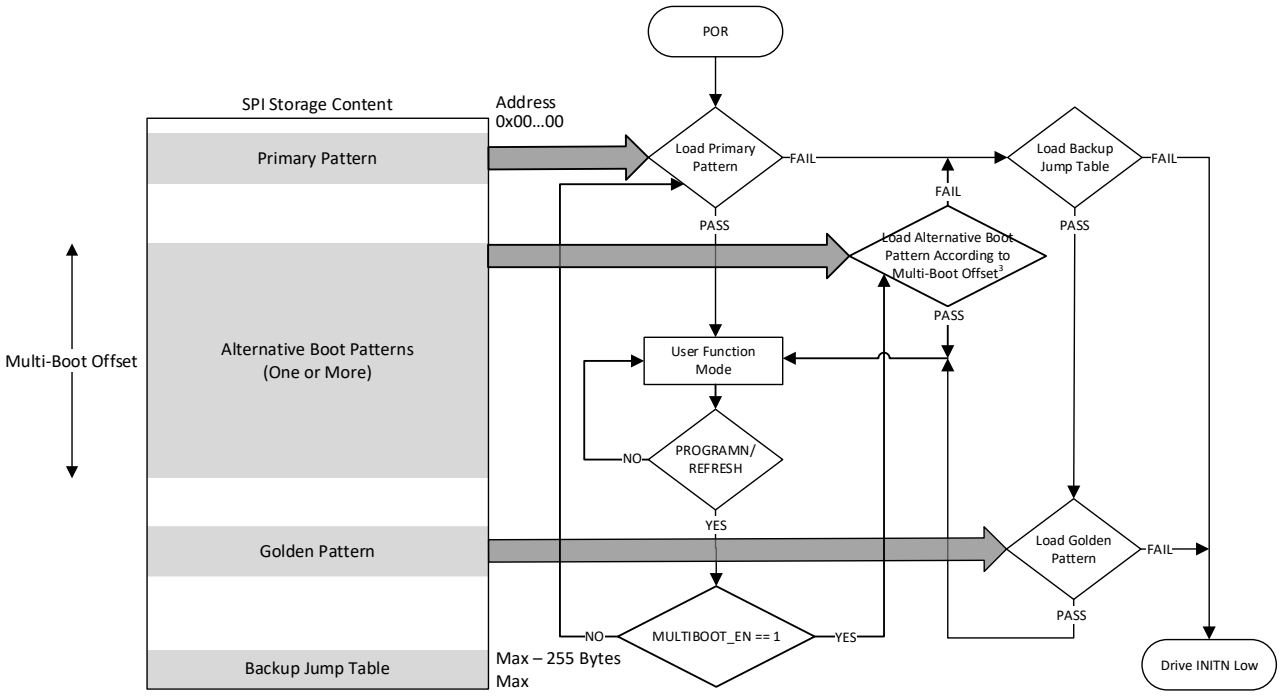
If the primary jump table is corrupted, the device loads the backup jump table, which then loads the golden pattern.

Note: The backup jump table is located at the last 256 bytes of the SPI storage device. When the configuration engine attempts to read the backup jump table, it issues a read offset address at 0xFFFF00 for 24-bit SPI flash, 0xFFFF_FF00 for 32-bit SPI flash, or 0xFFFF_FFFF_FF00 for 45-bit xSPI flash. The SPI flash truncates the most significant bit or bits of the address depending on the density of the SPI flash. For example, the 32-Mb SPI flash receives the read offset address as 0x3FFF00 even though the configuration engine sends 0xFFFF00.

If the backup jump table is corrupted or the golden pattern configuration fails, the Nexus 2 device stops driving the SPI clock and drives the INITN pin low.

6. Multi-Boot Mode

The Nexus 2 device multi-boot mode supports booting from at least three and up to 18 configuration bitstream patterns that reside in an external SPI storage device. These patterns are designated the primary pattern, golden pattern, and alternative boot patterns. When a Nexus 2 device boots up, it attempts to boot from the primary pattern. If the primary pattern fails to load, the device attempts to boot from the golden pattern. After successful configuration, the MULTI_BOOT_SEL attribute determines the subsequent loading of alternative boot patterns. Figure 6.1 shows the multi-boot flow diagram.



Notes:
1. The default primary boot offset is at address 0.
2. When MULTI_BOOT_SEL == STATIC, the multi-boot offset uses the value stored in the Multiboot Offset register. When MULTI_BOOT_SEL == DYNAMIC, the multi-boot offset uses the multi-boot address register programmed through the CONFIG_LMMI interface.
3. Load the primary pattern if the multi-boot offset points to the start address of the primary pattern in the SPI storage device.

Figure 6.1. Multi-Boot Flow Diagram

6.1. Multi-Boot Flow

If MULTI_BOOT_SEL == STATIC, after the primary pattern loads, reprogramming of the bitstream can be done by performing PROGRAMN pin pulsing or REFRESH command execution, which then triggers the loading of alternative boot pattern 1. Subsequent PROGRAMN pin pulsing or REFRESH command execution events load the next pattern defined in the multi-boot configuration. The bitstream pattern sequence, target address of the golden pattern, and target addresses of the alternative boot patterns are defined during the multi-boot configuration process in the Lattice Radiant™ Deployment Tool. If MULTI_BOOT_SEL == DYNAMIC, after the primary pattern loads, you can switch between all the different patterns stored in the external storage device freely.

If the primary pattern or alternative boot pattern fails to load, the device drives the INITN pin low to indicate an error and resets the configuration engine. After clearing the configuration RAM, the device drives the INITN pin high and reads the JUMP command in the backup jump table that directs it to load the golden pattern in the SPI storage device.

Note: The backup jump table is located at the last 256 bytes of the SPI storage device. When the configuration engine attempts to read the backup jump table, it issues a read offset address at 0xFFFF00 for 24-bit SPI flash, 0xFFFF_FF00 for 32-bit SPI flash, or 0xFFFF_FFFF_FF00 for 45-bit xSPI flash. The SPI flash truncates the most significant bit or bits of the address depending on the density of the SPI flash. For example, the 32-Mb SPI flash receives the read offset address as 0x3FFF00 even though the configuration engine sends 0xFFFF00.

If the golden pattern configuration fails, the Nexus 2 device stops driving the SPI clock and drives the INITN pin low.

The MULTI_BOOT_MODE and MULTI_BOOT_SEL attributes are set through the Global tab of the Device Constraint Editor in the Lattice Radiant software.

- MULTI_BOOT_MODE: Enables multi-boot mode.
- MULTI_BOOT_SEL: Selects the booting address for the multiple boot event when MULTI_BOOT_MODE is set to ENABLE.
  - STATIC (default): Use the value stored in the MultiBoot Offset register as the booting address. The MultiBoot Offset register value is set according to the bitstream pattern boot sequence in the Lattice Radiant Deployment Tool. Refer to the MultiBoot Offset section in the Lattice Nexus 2 sysCONFIG User Guide (FPGA-TN-02370) for more information about the MultiBoot Offset register.
  - DYNAMIC: Use the multi-boot address register programmed through the CONFIG_LMMI interface.

References

- Lattice Nexus 2 Platform – Overview Data Sheet (FPGA-DS-02122)
- Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121)
- Certus-N2 web page

A variety of technical notes for the Lattice Nexus 2 family are available.

- Lattice Nexus 2 Embedded Memory User Guide (FPGA-TN-02366)
- Lattice Nexus 2 High-Speed I/O and External Memory Interface (FPGA-TN-02372)
- Lattice Nexus 2 Power User Guide (FPGA-TN-02381)
- Lattice Nexus 2 sysCLOCK PLL Design and User Guide (FPGA-TN-02364)
- Lattice Nexus 2 Multi-Boot User Guide (FPGA-TN-02385)
- Lattice Nexus 2 sysCONFIG User Guide (FPGA-TN-02370)
- Lattice Nexus 2 SED/SEC User Guide (FPGA-TN-02380)
- Lattice Nexus 2 sysDSP User Guide (FPGA-TN-02362)
- Lattice Nexus 2 sysI/O User Guide (FPGA-TN-02365)
- Lattice Nexus 2 Configuration Security User Guide (FPGA-TN-02369)

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 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)
- High-Speed PCB Design Considerations (FPGA-TN-02178)
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

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 0.80, December 2024

Section	Change Summary
All	Initial preliminary release.

www.latticesemi.com