



# Lattice Nexus 2 sysCONFIG User Guide

***Preliminary*** Technical Note

FPGA-TN-02370-0.82

September 2026

## 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

|                                                  |    |
|--------------------------------------------------|----|
| Contents .....                                   | 3  |
| Abbreviations in This Document.....              | 12 |
| 1. Introduction .....                            | 14 |
| 2. Features .....                                | 15 |
| 3. Definition of Terms .....                     | 16 |
| 4. Configuration Details .....                   | 17 |
| 4.1. Bitstream and SPI Flash Sizes .....         | 17 |
| 4.2. Programming and Configuration Ports .....   | 17 |
| 4.3. Configuration Ports Arbitration .....       | 17 |
| 4.4. sysCONFIG Pins .....                        | 19 |
| 4.5. sysCONFIG Pin List.....                     | 20 |
| 4.6. System Pins .....                           | 21 |
| 4.6.1. CFGMODE.....                              | 21 |
| 4.6.2. PROGRAMN .....                            | 21 |
| 4.6.3. INITN.....                                | 21 |
| 4.6.4. DONE .....                                | 22 |
| 4.6.5. MCSNO/MSDO .....                          | 22 |
| 4.6.6. SCSNO/SSDO .....                          | 23 |
| 4.7. Controller SPI sysCONFIG Pins.....          | 23 |
| 4.7.1. MCLKP .....                               | 23 |
| 4.7.2. MCLKN .....                               | 23 |
| 4.7.3. MCSN.....                                 | 23 |
| 4.7.4. MMOSI/MDQ0 .....                          | 23 |
| 4.7.5. MMISO/MDQ1 .....                          | 24 |
| 4.7.6. MDQ[7:2].....                             | 24 |
| 4.7.7. MDS.....                                  | 24 |
| 4.7.8. MRSTN .....                               | 24 |
| 4.8. Target SPI sysCONFIG Pins .....             | 24 |
| 4.8.1. SCLKP .....                               | 24 |
| 4.8.2. SCSN .....                                | 24 |
| 4.8.3. SMOSI/SDQ0.....                           | 24 |
| 4.8.4. SMISO/SDQ1.....                           | 24 |
| 4.8.5. SDQ[7:2] .....                            | 24 |
| 4.8.6. SDS.....                                  | 24 |
| 4.9. JTAG Pins .....                             | 25 |
| 4.9.1. TCK .....                                 | 25 |
| 4.9.2. TMS.....                                  | 25 |
| 4.9.3. TDI .....                                 | 25 |
| 4.9.4. TDO .....                                 | 25 |
| 4.10. Port Persistence .....                     | 26 |
| 4.10.1. PERSISTENT Control Bits.....             | 26 |
| 4.10.2. sysCONFIG Port Persistence Options ..... | 26 |
| 5. Configuration Process and Flow .....          | 27 |
| 5.1. Power-Up Sequence .....                     | 28 |
| 5.2. Initialization.....                         | 28 |
| 5.3. Configuration .....                         | 29 |
| 5.3.1. Controller SPI Configuration.....         | 29 |
| 5.3.2. Target SPI Configuration.....             | 30 |
| 5.3.3. Self-Download Mode Configuration.....     | 31 |
| 5.4. Wake-Up .....                               | 31 |
| 5.4.1. Wake-Up Signals.....                      | 31 |
| 5.4.2. Wake-Up Sequence .....                    | 32 |

|         |                                                                      |    |
|---------|----------------------------------------------------------------------|----|
| 5.5.    | Early I/O Release .....                                              | 33 |
| 5.6.    | User Function Mode .....                                             | 33 |
| 6.      | Device Configuration .....                                           | 34 |
| 6.1.    | Configuration Modal States .....                                     | 34 |
| 6.2.    | Controller SPI Mode .....                                            | 35 |
| 6.2.1.  | Controller SPI Configuration Port Pins .....                         | 36 |
| 6.2.2.  | Enabling Controller SPI Port in Configuration Phase .....            | 36 |
| 6.2.3.  | Enabling Controller SPI Port Persistence in User Function Mode ..... | 36 |
| 6.2.4.  | Dual, Quad, and xSPI Controller SPI Read Mode .....                  | 37 |
| 6.3.    | Target SPI Mode .....                                                | 37 |
| 6.3.1.  | Target SPI Configuration Port Pins .....                             | 38 |
| 6.3.2.  | Enabling Target SPI Port in Configuration Phase .....                | 38 |
| 6.3.3.  | Enabling Target SPI Port Persistence in User Function Mode .....     | 38 |
| 6.3.4.  | Target SPI Port AC Timing Requirements .....                         | 38 |
| 6.3.5.  | Dual, Quad, and xSPI Target SPI Port .....                           | 39 |
| 6.3.6.  | Command Waveforms .....                                              | 39 |
| 6.3.7.  | Target SPI to Controller SPI Bridge .....                            | 41 |
| 6.4.    | JTAG Mode .....                                                      | 42 |
| 6.4.1.  | JTAG Port AC Timing Requirements .....                               | 42 |
| 6.4.2.  | JTAG to Controller SPI Bridge .....                                  | 43 |
| 6.4.3.  | JTAG Reveal Support .....                                            | 43 |
| 6.4.4.  | JTAG Operating Modes .....                                           | 44 |
| 6.5.    | Dual-Boot and Multi-Boot .....                                       | 46 |
| 6.6.    | Ping-Pong Boot .....                                                 | 47 |
| 6.7.    | Status Registers .....                                               | 48 |
| 6.7.1.  | Status Register 0 (STATUS0) .....                                    | 48 |
| 6.7.2.  | Status Register 1 (STATUS1) .....                                    | 50 |
| 6.7.3.  | Status Register 2 (STATUS2) .....                                    | 52 |
| 6.8.    | Control Registers .....                                              | 53 |
| 6.8.1.  | Control Register 0 (CR0) .....                                       | 53 |
| 6.8.2.  | Control Register 1 (CR1) .....                                       | 54 |
| 6.9.    | User Mode Register .....                                             | 56 |
| 6.10.   | Configuration Commands .....                                         | 57 |
| 6.10.1. | Command Format .....                                                 | 57 |
| 6.10.2. | Read Commands .....                                                  | 57 |
| 6.10.3. | Target Configuration Command Set .....                               | 58 |
| 6.10.4. | Target Configuration Command Details .....                           | 60 |
| 6.11.   | Self-Download Mode .....                                             | 74 |
| 6.12.   | Flash Memory Space Partition .....                                   | 74 |
| 7.      | Software Selectable Options .....                                    | 75 |
| 7.1.    | Accessing Software Selectable sysCONFIG Options .....                | 75 |
| 7.2.    | sysCONFIG Options .....                                              | 76 |
| 7.2.1.  | SLAVE_SPI_PORT .....                                                 | 77 |
| 7.2.2.  | MASTER_SPI_PORT .....                                                | 77 |
| 7.2.3.  | MSPI_RESET_PORT .....                                                | 78 |
| 7.2.4.  | MCCLK_FREQ .....                                                     | 78 |
| 7.2.5.  | COMPRESS_CONFIG .....                                                | 78 |
| 7.2.6.  | BOOT_SEL .....                                                       | 78 |
| 7.2.7.  | MSPI_RESET .....                                                     | 79 |
| 7.2.8.  | MULTI_BOOT_MODE .....                                                | 79 |
| 7.2.9.  | MULTI_BOOT_SEL .....                                                 | 79 |
| 7.2.10. | CONFIGIO_VOLTAGE_BANK1 .....                                         | 80 |
| 7.2.11. | CONFIGIO_VOLTAGE_BANK2 .....                                         | 80 |
| 7.2.12. | MSPI_SIGNATURE_TIMER .....                                           | 80 |

|         |                                                                                |     |
|---------|--------------------------------------------------------------------------------|-----|
| 7.2.13. | MSPI_TX_EDGE .....                                                             | 81  |
| 7.2.14. | MSPI_RX_EDGE .....                                                             | 83  |
| 7.2.15. | MSPI_CPHA .....                                                                | 83  |
| 7.2.16. | MSPI_CPOL .....                                                                | 83  |
| 7.2.17. | SSPI_TX_EDGE .....                                                             | 84  |
| 7.2.18. | SSPI_RX_EDGE .....                                                             | 84  |
| 7.2.19. | SSPI_CPHA .....                                                                | 84  |
| 7.2.20. | SSPI_CPOL .....                                                                | 85  |
| 7.2.21. | SSPI_SHIFT_ORDER .....                                                         | 85  |
| 7.2.22. | SSPI_DAISS_CHAIN_MODE .....                                                    | 85  |
| 7.2.23. | ERASE_EBR_ON_REFRESH .....                                                     | 85  |
| 7.2.24. | SIGNATURE_CHECK .....                                                          | 86  |
| 7.2.25. | MSPI_ADDRESS_32BIT .....                                                       | 86  |
| 7.2.26. | MSPI_COMMAND_32BIT .....                                                       | 86  |
| 7.2.27. | SSPI_IDLE_TIMER .....                                                          | 86  |
| 7.2.28. | MSPI_PREAMBLE_DETECTION_TIMER .....                                            | 87  |
| 7.2.29. | BACKGROUND_RECONFIG .....                                                      | 87  |
| 7.2.30. | DAISS_CHAIN .....                                                              | 87  |
| 7.2.31. | DAISS_CHAIN_WAIT_DONE .....                                                    | 88  |
| 7.2.32. | TRANSFR .....                                                                  | 88  |
| 7.2.33. | UserCode Format .....                                                          | 88  |
| 7.2.34. | UserCode .....                                                                 | 88  |
| 7.2.35. | MultiBoot Offset .....                                                         | 89  |
| 7.2.36. | BitstreamRevision Format .....                                                 | 89  |
| 7.2.37. | BitstreamRevision .....                                                        | 89  |
| 8.      | Daisy Chaining .....                                                           | 90  |
| 8.1.    | Flow-Through Mode .....                                                        | 90  |
| 8.2.    | Bypass Mode .....                                                              | 92  |
| 8.3.    | SCM Mode with External Host .....                                              | 93  |
| 8.4.    | Merging Daisy-Chain Bitstreams .....                                           | 94  |
|         | Appendix A. Nexus 2 Target SPI Programming Guide .....                         | 97  |
|         | Appendix B. Nexus 2 Device Information .....                                   | 98  |
|         | B.1. Device ID .....                                                           | 98  |
|         | B.2. Configuration Memory .....                                                | 98  |
|         | Appendix C. Modifying User OTP Settings .....                                  | 99  |
|         | C.1. Radiant Programmer Support for OTP Memory Access .....                    | 99  |
|         | C.2. Feature Rows Programming .....                                            | 100 |
|         | C.2.1. Feature Row Items .....                                                 | 100 |
|         | C.2.2. Program Feature Row .....                                               | 103 |
|         | C.2.3. Update Feature Row .....                                                | 104 |
|         | C.2.4. Program Control NV Register 1 .....                                     | 105 |
|         | C.3. Advanced Security Keys Programming .....                                  | 106 |
|         | Appendix D. Configuration Access from User Logic .....                         | 107 |
|         | D.1. CONFIG_LMMIC Primitive .....                                              | 107 |
|         | D.2. CONFIG_LMMIC Primitive Connection and Instantiation .....                 | 108 |
|         | D.3. CONFIG_LMMIC Access Protocol .....                                        | 109 |
|         | D.3.1. LMMI Offset Assignment for CFG Access .....                             | 109 |
|         | D.3.2. LMMI CFG Command Execution .....                                        | 110 |
|         | D.3.3. CONFIG_LMMIC Supported Commands .....                                   | 111 |
|         | Appendix E. Flash Programming .....                                            | 113 |
|         | E.1. Flash Memory Space Partition .....                                        | 113 |
|         | Appendix F. JTAG Daisy-Chain Configuration with External Host Controller ..... | 114 |
|         | F.1. Overview and Limitations .....                                            | 114 |
|         | F.2. Creating a JTAG Daisy-Chain Configuration .....                           | 115 |

|                                                                              |     |
|------------------------------------------------------------------------------|-----|
| F.3. Saving the .xcf File.....                                               | 116 |
| F.4. Generating a STAPL File from the .xcf File .....                        | 116 |
| F.5. Locating and Extracting Adjusted Bitstream Data in the STAPL File ..... | 120 |
| References .....                                                             | 121 |
| Technical Support Assistance .....                                           | 122 |
| Revision History .....                                                       | 123 |

## Figures

|                                                                                                        |     |
|--------------------------------------------------------------------------------------------------------|-----|
| Figure 4.1. Configuration Control Flow.....                                                            | 18  |
| Figure 4.2. sysCONFIG Pins .....                                                                       | 19  |
| Figure 4.3. Configuration from PROGRAMN Timing .....                                                   | 21  |
| Figure 4.4. Configuration Error Notification .....                                                     | 22  |
| Figure 5.1. Configuration Flow .....                                                                   | 27  |
| Figure 5.2. Configuration from POR Timing.....                                                         | 28  |
| Figure 5.3. Target SPI Configuration Flow .....                                                        | 30  |
| Figure 5.4. Wake-Up Sequence with DAISY_CHAIN_WAIT_DONE = DISABLED .....                               | 32  |
| Figure 5.5. Wake-Up Sequence with DAISY_CHAIN_WAIT_DONE = ENABLED.....                                 | 32  |
| Figure 6.1. Configuration Modal States .....                                                           | 34  |
| Figure 6.2. Nexus 2 Controller SPI Port with SPI Flash.....                                            | 35  |
| Figure 6.3. Nexus 2 Target SPI Port with CPU and Single or Multiple Devices .....                      | 37  |
| Figure 6.4. Read Command Waveforms .....                                                               | 39  |
| Figure 6.5. Data Read Command Waveforms .....                                                          | 39  |
| Figure 6.6. Data Write Command Waveforms .....                                                         | 40  |
| Figure 6.7. Immediate Action Command Waveforms .....                                                   | 40  |
| Figure 6.8. Delayed Action Command Waveforms.....                                                      | 40  |
| Figure 6.9. Target SPI to Controller SPI Bridge Functional Diagram .....                               | 41  |
| Figure 6.10. Target SPI to Controller SPI Bridge Block Diagram .....                                   | 41  |
| Figure 6.11. JTAG to Controller SPI Bridge Block Diagram.....                                          | 43  |
| Figure 6.12. JTAG CONFIGURATION_DATA_SHIFT .....                                                       | 45  |
| Figure 6.13. JTAG CONFIGURATION_BURST .....                                                            | 45  |
| Figure 6.14. Jump Table .....                                                                          | 47  |
| Figure 6.15. Configuration Command Format .....                                                        | 57  |
| Figure 6.16. Segmented Bitstream Burst Mode Timing Waveforms .....                                     | 60  |
| Figure 7.1. sysCONFIG Options in Global Page of Device Constraint Editor .....                         | 75  |
| Figure 7.2. CPOL, CPHA, and TX/RX_EDGE Settings versus Waveform Examples .....                         | 82  |
| Figure 8.1. Lattice Nexus 2 in Configuration Daisy Chain in Flow-Through Mode.....                     | 91  |
| Figure 8.2. Lattice Nexus 2 in Configuration Daisy Chain in Bypass Mode .....                          | 92  |
| Figure 8.3. Lattice Nexus 2 in Configuration Daisy Chain in SCM Mode with External Host .....          | 93  |
| Figure 8.4. Radiant Deployment Tool – Getting Started (sysCONFIG Daisy Chain) .....                    | 94  |
| Figure 8.5. Radiant Deployment Tool – Step 1 of 4: Select Input File(s) (sysCONFIG Daisy Chain) .....  | 94  |
| Figure 8.6. Radiant Deployment Tool – Step 2 of 4: sysCONFIG Daisy Chain Options .....                 | 95  |
| Figure 8.7. Radiant Deployment Tool – Step 3 of 4: Select Output File(s) (sysCONFIG Daisy Chain) ..... | 96  |
| Figure 8.8. Radiant Deployment Tool – Step 4 of 4: Generate Deployment (sysCONFIG Daisy Chain).....    | 96  |
| Figure C.1. Non-volatile Configuration Memory Access from Radiant Programmer.....                      | 99  |
| Figure C.2. Feature Rows Programming Operations .....                                                  | 100 |
| Figure C.3. Programming Feature Row .....                                                              | 103 |
| Figure C.4. Updating Feature Row .....                                                                 | 104 |
| Figure C.5. Programming Non-volatile Control Register 1.....                                           | 105 |
| Figure C.6. Control Register 1 Window.....                                                             | 105 |
| Figure D.1. CONFIG_LMMIC Primitive Pin Diagram.....                                                    | 107 |
| Figure D.2. CONFIG_LMMIC Primitive Connection .....                                                    | 108 |
| Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode .....                                     | 110 |
| Figure D.4. Waveform for LMMI Read Operation .....                                                     | 110 |
| Figure F.1. JTAG Daisy Chain – Radiant Programmer .....                                                | 115 |
| Figure F.2. JTAG Daisy Chain – Creating a New Blank Project .....                                      | 115 |
| Figure F.3. JTAG Daisy Chain – Configuring the Device .....                                            | 116 |
| Figure F.4. Radiant Deployment Tool – Getting Started Window .....                                     | 117 |
| Figure F.5. Radiant Deployment Tool – Step 1 of 4: Select Input File(s).....                           | 117 |
| Figure F.6. Radiant Deployment Tool – Step 2 of 4: STAPL Options .....                                 | 118 |
| Figure F.7. Radiant Deployment Tool – Step 3 of 4: Select Output File(s).....                          | 119 |

Figure F.8. Radiant Deployment Tool – Step 4 of 4: Generate Deployment .....119

## Tables

|                                                                     |    |
|---------------------------------------------------------------------|----|
| Table 3.1. Definition of Terms .....                                | 16 |
| Table 4.1. Bitstream Size versus Recommended SPI Flash Size .....   | 17 |
| Table 4.2. Nexus 2 Device Programming and Configuration Ports ..... | 17 |
| Table 4.3. Default State of sysCONFIG Pins .....                    | 20 |
| Table 4.4. sysCONFIG Port Persistence Options .....                 | 26 |
| Table 5.1. Wake-Up Signals .....                                    | 31 |
| Table 6.1. Controller SPI Configuration Port Pins .....             | 36 |
| Table 6.2. Target SPI Configuration Port Pins .....                 | 38 |
| Table 6.3. Target SPI Port AC Timing Requirements .....             | 38 |
| Table 6.4. JTAG AC Timing Requirements .....                        | 42 |
| Table 6.5. JTAG Direct Mode Instructions .....                      | 44 |
| Table 6.6. Status Register 0 .....                                  | 48 |
| Table 6.7. Status Register 1 .....                                  | 50 |
| Table 6.8. Status Register 2 .....                                  | 52 |
| Table 6.9. Control Register 0 .....                                 | 53 |
| Table 6.10. Control Register 1 .....                                | 54 |
| Table 6.11. User Mode Register .....                                | 56 |
| Table 6.12. Target Configuration Commands .....                     | 58 |
| Table 6.13. PROG_ENABLE Command .....                               | 60 |
| Table 6.14. PROG_DISABLE Command .....                              | 60 |
| Table 6.15. BITSTREAM_BURST Command .....                           | 60 |
| Table 6.16. CRAM_INIT_ADDRESS Command .....                         | 61 |
| Table 6.17. CRAM_INIT_ADDRESS Operand .....                         | 61 |
| Table 6.18. CRAM_WRITE_ADDRESS Command .....                        | 61 |
| Table 6.19. CRAM_WRITE_ADDRESS Operand .....                        | 61 |
| Table 6.20. CRAM_PROG_INC Command .....                             | 62 |
| Table 6.21. CRAM_PROG_INC Operand .....                             | 62 |
| Table 6.22. CRAM_READ_INC Command .....                             | 62 |
| Table 6.23. CRAM_READ_INC Operand .....                             | 62 |
| Table 6.24. CRAM_READ_INC Return Data .....                         | 62 |
| Table 6.25. INIT_READ_REG Command .....                             | 63 |
| Table 6.26. INIT_READ_REG Return Data .....                         | 63 |
| Table 6.27. INIT_READ_EBR Command .....                             | 63 |
| Table 6.28. INIT_READ_EBR Return Data .....                         | 63 |
| Table 6.29. PROG_DONE Command .....                                 | 63 |
| Table 6.30. PROG_DONE Operand .....                                 | 63 |
| Table 6.31. READ_IDCODE_PUB Command .....                           | 64 |
| Table 6.32. READ_IDCODE_PUB Return Data .....                       | 64 |
| Table 6.33. READ_IDCODE_PRV Command .....                           | 64 |
| Table 6.34. READ_IDCODE_PRV Return Data .....                       | 64 |
| Table 6.35. READ_UIDCODE_PUB_L Command .....                        | 64 |
| Table 6.36. READ_UIDCODE_PUB_L Return Data .....                    | 64 |
| Table 6.37. READ_UIDCODE_PUB_H Command .....                        | 64 |
| Table 6.38. READ_UIDCODE_PUB_H Return Data .....                    | 64 |
| Table 6.39. READ_SECURE_UID Command .....                           | 65 |
| Table 6.40. READ_SECURE_UID Return Data .....                       | 65 |
| Table 6.41. READ_USERCODE Command .....                             | 65 |
| Table 6.42. READ_USERCODE Return Data .....                         | 65 |
| Table 6.43. READ_DR_USERCODE Command .....                          | 65 |
| Table 6.44. READ_DR_USERCODE Return Data .....                      | 65 |
| Table 6.45. READ_STATUS0 Command .....                              | 66 |
| Table 6.46. READ_STATUS0 Return Data .....                          | 66 |

|                                                |    |
|------------------------------------------------|----|
| Table 6.47. READ_STATUS1 Command.....          | 66 |
| Table 6.48. READ_STATUS1 Return Data.....      | 66 |
| Table 6.49. READ_STATUS2 Command.....          | 66 |
| Table 6.50. READ_STATUS2 Return Data.....      | 66 |
| Table 6.51. CHECK_BUSY Command.....            | 67 |
| Table 6.52. CHECK_BUSY Return Data.....        | 67 |
| Table 6.53. PROG_CNTRL0 Command.....           | 67 |
| Table 6.54. READ_CNTRL0 Command.....           | 67 |
| Table 6.55. READ_CNTRL0 Return Data.....       | 67 |
| Table 6.56. PROG_CNTRL1 Command.....           | 68 |
| Table 6.57. READ_CNTRL1 Command.....           | 68 |
| Table 6.58. READ_CNTRL1 Return Data.....       | 68 |
| Table 6.59. READ_UMR Command.....              | 68 |
| Table 6.60. READ_UMR Return Data.....          | 68 |
| Table 6.61. DEVICE_CTRL Command.....           | 69 |
| Table 6.62. DEVICE_CTRL Operand.....           | 69 |
| Table 6.63. DRY_RUN_CTRL Command.....          | 69 |
| Table 6.64. DRY_RUN_CTRL Operand.....          | 69 |
| Table 6.65. REFRESH Command.....               | 69 |
| Table 6.66. SSPI_MODE Command.....             | 70 |
| Table 6.67. SSPI_MODE Operand.....             | 70 |
| Table 6.68. MSPI_BRIDGE Command.....           | 70 |
| Table 6.69. MSPI_BRIDGE_CLOCK Command.....     | 70 |
| Table 6.70. MSPI_BRIDGE_CLOCK Operand.....     | 70 |
| Table 6.71. PORT_REQUEST Command.....          | 71 |
| Table 6.72. PORT_REQUEST Operand.....          | 71 |
| Table 6.73. PORT_STATUS_READ Command.....      | 71 |
| Table 6.74. PORT_STATUS_READ Return Data.....  | 71 |
| Table 6.75. READ_SED_CRC Command.....          | 72 |
| Table 6.76. READ_SED_CRC Return Data.....      | 72 |
| Table 6.77. CALC_SED_CRC Command.....          | 72 |
| Table 6.78. READ_SED_CRC_CALC Command.....     | 72 |
| Table 6.79. READ_SED_CRC_CALC Return Data..... | 72 |
| Table 6.80. READ_OTP Command.....              | 72 |
| Table 6.81. READ_OTP Operand.....              | 72 |
| Table 6.82. READ_OTP Return Data.....          | 72 |
| Table 6.83. PROG_OTP Command.....              | 73 |
| Table 6.84. PROG_OTP Operand.....              | 73 |
| Table 6.85. PROG_OTP_SHADOW Command.....       | 73 |
| Table 6.86. PROG_OTP_SHADOW Operand.....       | 73 |
| Table 6.87. LOCK_OTP_ROW Command.....          | 73 |
| Table 6.88. LOCK_OTP_ROW Operand.....          | 73 |
| Table 6.89. LOCK_USER_OTP_BLOCK Command.....   | 74 |
| Table 7.1. sysCONFIG Options.....              | 76 |
| Table 7.2. SLAVE_SPI_PORT Option.....          | 77 |
| Table 7.3. MASTER_SPI_PORT Option.....         | 77 |
| Table 7.4. MSPI_RESET_PORT Option.....         | 78 |
| Table 7.5. MCCLK_FREQ Option.....              | 78 |
| Table 7.6. COMPRESS_CONFIG Option.....         | 78 |
| Table 7.7. BOOT_SEL Option.....                | 78 |
| Table 7.8. MSPI_RESET Option.....              | 79 |
| Table 7.9. MULTI_BOOT_MODE Option.....         | 79 |
| Table 7.10. MULTI_BOOT_SEL Option.....         | 79 |
| Table 7.11. CONFIGIO_VOLTAGE_BANK1 Option..... | 80 |

|                                                                                          |     |
|------------------------------------------------------------------------------------------|-----|
| Table 7.12. CONFIGIO_VOLTAGE_BANK2 Option.....                                           | 80  |
| Table 7.13. MSPI_SIGNATURE_TIMER Option .....                                            | 80  |
| Table 7.14. MSPI_TX_EDGE Option .....                                                    | 81  |
| Table 7.15. CPOL, CPHA, and TX/RX_EDGE Settings versus Tx and Rx Operations.....         | 81  |
| Table 7.16. MSPI_RX_EDGE Option .....                                                    | 83  |
| Table 7.17. MSPI_CPHA Option .....                                                       | 83  |
| Table 7.18. MSPI_CPOL Option.....                                                        | 83  |
| Table 7.19. SSPI_TX_EDGE Option .....                                                    | 84  |
| Table 7.20. SSPI_RX_EDGE Option.....                                                     | 84  |
| Table 7.21. SSPI_CPHA Option .....                                                       | 84  |
| Table 7.22. SSPI_CPOL Option .....                                                       | 85  |
| Table 7.23. SSPI_SHIFT_ORDER Option .....                                                | 85  |
| Table 7.24. SSPI_DAISSY_CHAIN_MODE Option .....                                          | 85  |
| Table 7.25. ERASE_EBR_ON_REFRESH Option .....                                            | 85  |
| Table 7.26. SIGNATURE_CHECK Option.....                                                  | 86  |
| Table 7.27. MSPI_ADDRESS_32BIT Option .....                                              | 86  |
| Table 7.28. MSPI_COMMAND_32BIT Option .....                                              | 86  |
| Table 7.29. SSPI_IDLE_TIMER Option .....                                                 | 86  |
| Table 7.30. MSPI_PREAMBLE_DETECTION_TIMER Option.....                                    | 87  |
| Table 7.31. BACKGROUND_RECONFIG Option .....                                             | 87  |
| Table 7.32. DAISY_CHAIN Option .....                                                     | 87  |
| Table 7.33. DAISY_CHAIN_WAIT_DONE Option .....                                           | 88  |
| Table 7.34. TRANSFR Option.....                                                          | 88  |
| Table 7.35. UserCode Format Option .....                                                 | 88  |
| Table 7.36. UserCode Option.....                                                         | 88  |
| Table 7.37. MultiBoot Offset Option .....                                                | 89  |
| Table 7.38. BitstreamRevision Format Option.....                                         | 89  |
| Table 7.39. BitstreamRevision Option .....                                               | 89  |
| Table 8.1. Required sysCONFIG Settings for Flow-Through Mode .....                       | 91  |
| Table 8.2. Required sysCONFIG Settings for Bypass Mode.....                              | 92  |
| Table 8.3. Required sysCONFIG Settings for SCM Mode with External Host.....              | 93  |
| Table B.1. Lattice Nexus 2 Device ID.....                                                | 98  |
| Table B.2. Lattice Nexus 2 CRRAM Frame Sizes .....                                       | 98  |
| Table C.1. Feature Row Items .....                                                       | 100 |
| Table D.1. CONFIG_LMMIC Primitive Pins <sup>1</sup> .....                                | 107 |
| Table D.2. LMMI CFG Offset List .....                                                    | 109 |
| Table D.3. CONFIG_LMMIC Supported Commands.....                                          | 111 |
| Table E.1. Number of Blocks of Flash Memory for LN2-MH-70ES1 Device .....                | 113 |
| Table E.2. Number of Blocks of Flash Memory for LN2-MH-20 and LN2-MH-20ES1 Devices ..... | 113 |

## Abbreviations in This Document

A list of abbreviations used in this document.

| Abbreviation | Definition                                                                                                          |
|--------------|---------------------------------------------------------------------------------------------------------------------|
| AES          | Advanced Encryption Standard                                                                                        |
| BGRAM        | Battery Backed RAM                                                                                                  |
| BSDL         | Boundary Scan Description Language                                                                                  |
| BSE          | Bitstream Engine                                                                                                    |
| CFG          | Configuration                                                                                                       |
| CID          | Customer ID                                                                                                         |
| CPHA         | Clock Phase                                                                                                         |
| CPOL         | Clock Polarity                                                                                                      |
| CRAM         | Configuration Random Access Memory (also known as configuration SRAM or SRAM configuration memory in this document) |
| CRC          | Cyclic Redundancy Check                                                                                             |
| DDR          | Double Data Rate                                                                                                    |
| DR           | Data Register (JTAG Shift-DR, Update-DR)                                                                            |
| DTR          | Dual Transfer Rate                                                                                                  |
| EBR          | Embedded Block RAM                                                                                                  |
| ECDSA        | Elliptic Curve Digital Signature Algorithm                                                                          |
| FPGA         | Field-Programmable Gate Array                                                                                       |
| FSM          | Finite State Machine                                                                                                |
| GOE          | Global Output Enable                                                                                                |
| GPIO         | General Purpose I/O                                                                                                 |
| GSRN         | Global Set/Reset                                                                                                    |
| GWE          | Global Write Enable                                                                                                 |
| IDCODE       | Identification Code                                                                                                 |
| ILA          | Internal Logic Analyzer                                                                                             |
| IMM          | Immediate Action                                                                                                    |
| IR           | Instruction Register (JTAG Shift-IR)                                                                                |
| JEDEC        | Joint Electron Device Engineering Council                                                                           |
| JTAG         | Joint Test Action Group                                                                                             |
| LMMI         | Lattice Memory Mapped Interface                                                                                     |
| LSB          | Least Significant Bit                                                                                               |
| LUT          | Look Up Table                                                                                                       |
| MSB          | Most Significant Bit                                                                                                |
| MSPI         | Controller Serial Peripheral Interface                                                                              |
| OTP          | One-Time Programmable                                                                                               |
| PCB          | Printed Circuit Board                                                                                               |
| POR          | Power On Reset                                                                                                      |
| PUF          | Physically Unclonable Function                                                                                      |
| RSA          | Rivest-Shamir-Adleman (cryptosystem)                                                                                |
| RTL          | Register-Transfer Level                                                                                             |
| SCM          | Serial Configuration Mode                                                                                           |
| SDM          | Self Download Mode                                                                                                  |
| SED          | Soft Error Detection                                                                                                |
| SFDP         | Serial Flash Discoverable Parameters                                                                                |
| SPI          | Serial Peripheral Interface                                                                                         |
| SRAM         | Static Random-Access Memory                                                                                         |

| Abbreviation | Definition                             |
|--------------|----------------------------------------|
| SSPI         | Target Serial Peripheral Interface     |
| STAPL        | Standard Test and Programming Language |
| TAP          | Test Access Port (JTAG)                |
| TCK          | Test Clock Pin                         |
| TDI          | Test Data Input                        |
| TDO          | Test Data Output                       |
| TDR          | Test Data Register (JTAG)              |
| TMS          | Test Mode Select                       |
| UES          | User Electronic Signature              |
| xSPI         | Expanded Serial Peripheral Interface   |

# 1. Introduction

The Lattice Nexus™ 2 FPGA platform combines Lattice's low power FPGA expertise with leading 16-nm semiconductor manufacturing technology to deliver device families with low power, high performance, high reliability, and small form factor. It supports many new and unique features, thereby making a Lattice Nexus 2 FPGA one of the best-in-class FPGAs in its logic density range. Lattice Nexus 2 FPGAs (herein referred to as Nexus 2 FPGA or Nexus 2 device), featuring the ultrafast I/O booting capability, have one of the fastest boot-up times and offer advanced options such as multi-boot to easily switch between FPGA bitstreams.

The configuration memory in the Nexus 2 FPGA is built using volatile SRAM; therefore, an external or internal non-volatile configuration memory or external controller is required to maintain the configuration data when power is removed. This non-volatile memory or external controller supplies the configuration data to the Nexus 2 device when it powers up or anytime the device configuration needs to be updated. The Nexus 2 device provides a rich set of features for configuring the FPGA or programming the external or internal non-volatile memory. The many options available provide the flexibility to build a programming or configuration solution that suits a particular set of needs. This document describes the available options.

The Nexus 2 FPGA provides internal one-time programmable (OTP) memory for user feature setup, security setup, and security locking for different memory sectors and configuration ports. It is strongly recommended that the write lock for the OTP feature row be set after device setup is finalized to prevent accidental changes to user features and security settings.

## Notes:

- Waveforms presented in this document are for reference only; for detailed timing recommendations, refer to the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).
- For the descriptions in this document, the following conditions apply:
  - Only Mach-N2 devices contain internal (on-chip) flash memory.
  - Only Mach-N2 devices support the self-download mode (SDM). Any information on SDM in this document only applies to Mach-N2 devices.
  - Mach-N2 devices do not support the controller SPI mode. Any information on controller SPI in this document only applies to Certus-N2 devices.

## 2. Features

The following are the main programming and configuration features of Nexus 2 devices:

- Ultrafast I/O configuration for instant-on support
- Fast device configuration from external flash with controller SPI (x1, x2, x4) and xSPI (x8, dual transfer rate) or from on-chip flash with SDM
- Bitstream dry-run support to ensure bitstream integrity
- Enhanced and flexible multi-boot support (32-bit addressing with jump table support)
- Configuration bridging for easy external SPI programming (target SPI to controller SPI bridge, JTAG to controller SPI bridge, LMMI to controller SPI bridge) or on-chip flash programming in either configuration mode or user function mode
- User-selectable booting sequence
- Bitstream encryption – AES-256. For detailed information, refer to the Lattice Nexus 2 Configuration Security User Guide (FPGA-TN-02369).
- Bitstream authentication – ECDSA and RSA. For detailed information, refer to the Lattice Nexus 2 Configuration Security User Guide (FPGA-TN-02369).
- Multiple programming and configuration interfaces:
  - JTAG (IEEE 1149.1)
  - Controller SPI (serial, dual, and quad modes) and xSPI (x8, dual transfer rate)
  - Target SPI (serial, dual, and quad modes) and xSPI (x8, dual transfer rate)
  - Configuration daisy chaining
- Ping-pong boot
- Readback security and encryption for design protection
- Bitstream compression

## 3. Definition of Terms

Table 3.1 lists the terms used in this document to describe common functions, features, or concepts.

**Table 3.1. Definition of Terms**

| Term                  | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BIT                   | The BIT file is the configuration data for the FPGA device that is stored in an external SPI storage device, internal flash memory, or other memory device. It is a binary file and is programmed unmodified into the non-volatile memory by the Lattice Radiant™ Programmer.                                                                                                                                                                                 |
| Configuration         | A change in the state of the SRAM memory cells.                                                                                                                                                                                                                                                                                                                                                                                                               |
| Configuration Command | An instruction issued through the bitstream to the configuration logic to perform an operation during device configuration.                                                                                                                                                                                                                                                                                                                                   |
| Configuration Data    | The data read from the non-volatile memory and loaded into the FPGA device SRAM configuration memory. This is also referred to as a bitstream or device bitstream.                                                                                                                                                                                                                                                                                            |
| Configuration Mode    | The method the FPGA device uses to acquire the configuration data from the non-volatile memory.                                                                                                                                                                                                                                                                                                                                                               |
| Dry-Run               | The process triggered by the DRY_RUN_CTRL command, which loads the bitstream and checks the CRC of the non-volatile bits without writing the bits to the configuration SRAM (that is, it is done in the background during normal device operation), to check the bitstream file integrity.                                                                                                                                                                    |
| Dual-Boot             | This feature allows the FPGA device to support two configuration images that reside in a non-volatile memory 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 memory.                                                                                                                                                                   |
| HEX                   | The HEX file is the configuration data for the FPGA device in the HEX format. It is normally requested by third-party programming vendors.                                                                                                                                                                                                                                                                                                                    |
| Multi-Boot            | The FPGA device determines and triggers the loading of the next image after a prior successful configuration. Multiple images (that is, two images or more) are available for the FPGA device to choose to load on demand. All images are stored in non-volatile memory.                                                                                                                                                                                      |
| Number Formats        | <p>The following nomenclature is used to denote the radix of numbers:</p> <ul style="list-style-type: none"> <li>0x: Numbers preceded by 0x are hexadecimal.</li> <li>b (suffix): Numbers suffixed with b are binary.</li> <li>All other numbers are decimal.</li> </ul> <p><b>Note:</b> When specifying binary numbers in relation to register bit settings, the suffix b may be excluded. Binary numbers may also be represented in the Verilog format.</p> |
| Ping-Pong Boot        | This feature allows the FPGA device to use the jump table to select an image for booting without changing the location of the image in the non-volatile memory storage device.                                                                                                                                                                                                                                                                                |
| Port                  | The physical connection used to perform programming and some configuration operations. Ports on the Nexus 2 device might include JTAG, target SPI, and controller SPI.                                                                                                                                                                                                                                                                                        |
| Programming           | The process used to alter the contents of the external or internal 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).                                                                                                                                                                                                                                                                                    |
| SPI                   | The serial peripheral interface is an industry-standard, full-duplex, synchronous serial data link or bus that uses a four-wire interface. The interface supports a single controller and single or multiple targets.                                                                                                                                                                                                                                         |
| Transparent Mode      | Also referred to as background mode, this mode is used to access the configuration memory while leaving the FPGA device in user function mode with all I/O pins remaining operational. When the device is in transparent mode, configuration register access for SERDES and I/O settings is disabled.                                                                                                                                                         |
| Unprogrammed Mode     | The FPGA device is in an unprogrammed mode with all I/O pins kept tri-stated.                                                                                                                                                                                                                                                                                                                                                                                 |
| User Function Mode    | The FPGA device is in user function mode when configuration is complete and the device is performing the logic functions it has been programmed to perform.                                                                                                                                                                                                                                                                                                   |

## 4. Configuration Details

The Nexus 2 device SRAM configuration memory contains the configuration data that defines the functional behavior of the FPGA in user function mode. Configuration data that is loaded into the SRAM configuration memory is either retrieved from an external or internal non-volatile memory or transmitted to the device through a configuration port.

### 4.1. Bitstream and SPI Flash Sizes

Nexus 2 devices are SRAM-based FPGAs. The SRAM configuration memory must be loaded from an external or internal 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 resources 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 4.1. Bitstream Size versus Recommended SPI Flash Size**

| Device                 | Scenario    | Bitstream Size (Mb) <sup>1</sup> | Recommended SPI Flash Size (Mb) |           |
|------------------------|-------------|----------------------------------|---------------------------------|-----------|
|                        |             |                                  | Single Boot                     | Dual Boot |
| LN2-CT-20              | No EBR      | 44                               | 64                              | 128       |
|                        | Maximum EBR | 64.1                             | 128                             | 256       |
| LN2-MH-20 <sup>2</sup> | No EBR      | 44                               | —                               | —         |
|                        | Maximum EBR | 64.1                             | —                               | —         |

**Notes:**

- Both unencrypted and encrypted bitstreams are the same size. Bitstream compression ratio varies depending on the bitstream, so only uncompressed bitstream sizes are shown.
- The LN2-MH-20 device contains 256 Mb of internal flash memory.

### 4.2. Programming and Configuration Ports

Table 4.2 shows the ports supported by Nexus 2 devices for programming and configuration, which include the industry-standard JTAG interface. Each port provides a method to access the Nexus 2 device SRAM configuration memory. The availability of these ports during device configuration is discussed in the [Configuration Ports Arbitration](#) section.

**Table 4.2. Nexus 2 Device Programming and Configuration Ports**

| Interface  | Port                                          | Description                            |
|------------|-----------------------------------------------|----------------------------------------|
| JTAG       | JTAG (IEEE 1149.1)                            | 4-wire JTAG interface                  |
| sysCONFIG™ | Target SPI (serial, dual, quad, xSPI-DTR)     | Target serial peripheral interface     |
|            | Controller SPI (serial, dual, quad, xSPI-DTR) | Controller serial peripheral interface |

### 4.3. Configuration Ports Arbitration

**Note:** Only Certus-N2 devices support the controller SPI mode. Only Mach-N2 devices support the self-download mode.

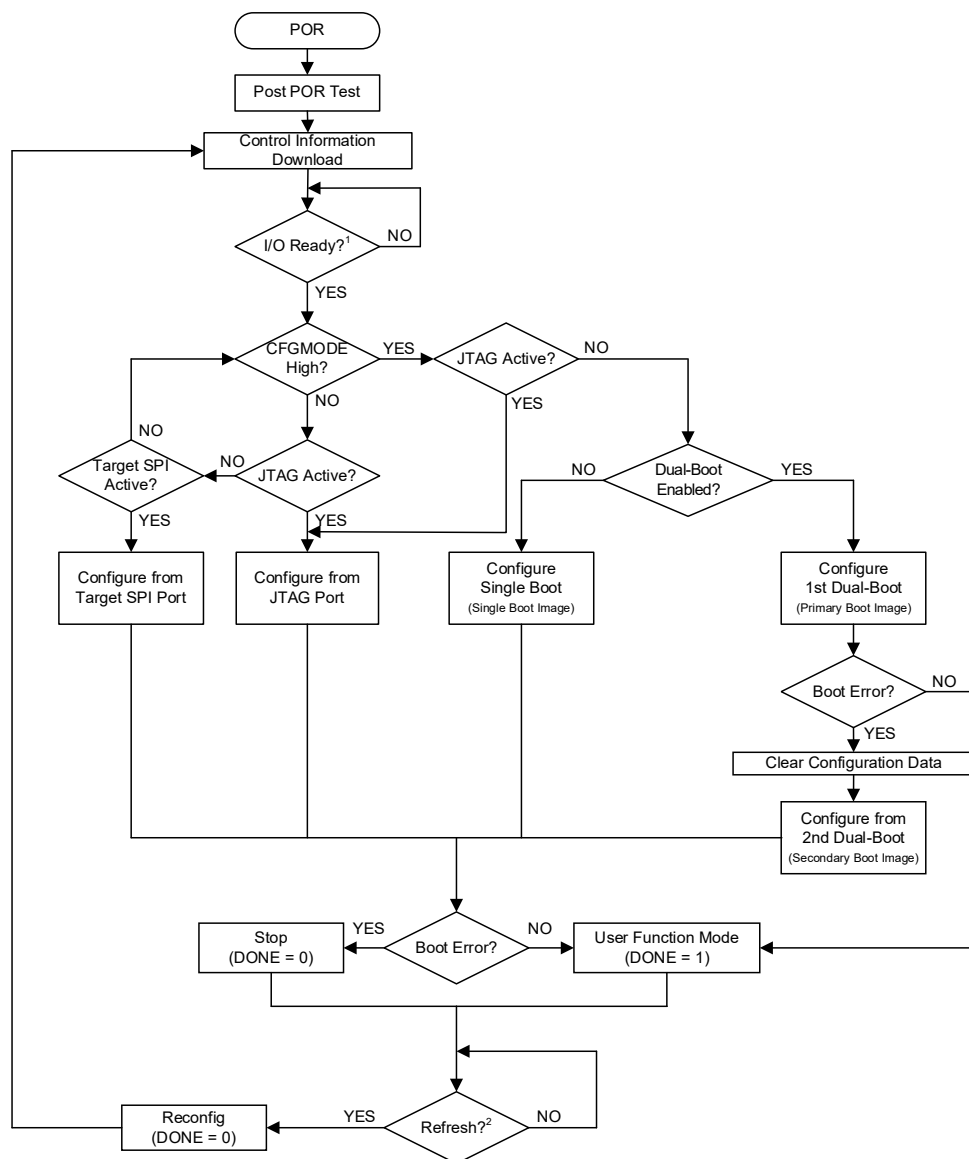
At power-up, PROGRAMN pin pulsing, or REFRESH command execution, the configuration logic erases the user logic configuration data in the device and optionally erases EBR data depending on settings in Control Register 1 (CR1). After the erasure process completes, if the CFGMODE pin is high, the device enters controller SPI mode (automatic booting) and starts the configuration data download from an external SPI flash or enters self-download mode and starts the configuration data download from the internal flash; if the CFGMODE pin is low, the device enters target mode and waits for a target request from either the JTAG or target SPI port. To avoid contention between the JTAG and target SPI ports, a port can request exclusive access using the PORT\_REQUEST command as described in the [PORT\\_REQUEST](#) section.

During configuration, the CFGMODE pin continues to dictate whether the device is in controller SPI/self-download or target SPI mode. If the CFGMODE pin is toggled low during controller SPI/self-download mode booting, the boot

process is aborted, and the device switches to target SPI mode. The status register provides an indicator that shows whether the boot process was aborted. For the external controller to reconfigure the device, it must either pulse the PROGRAMN pin or issue a REFRESH command to clear the configuration memory contents. If the CFGMODE pin is toggled high during target SPI mode configuration, the configuration process is aborted, and the device switches to controller SPI/self-download mode. To start the boot process over the controller SPI port or from on-chip flash, either pulse the PROGRAMN pin or issue a REFRESH command.

The PROGRAMN and CFGMODE pins are dedicated sysCONFIG pins. In user function mode, dedicated sysCONFIG pins retain their function. However, for the target SPI port shared (dual-purpose) pins to function as sysCONFIG pins in user function mode, the target SPI port must be persisted. When persisted, target SPI port pins function as sysCONFIG pins regardless of the CFGMODE pin state. If the target SPI port is not persisted, the shared (dual-purpose) pins can be used as general purpose I/O pins.

Figure 4.1 shows the device configuration control flow.



**Notes:**

1. *I/O Ready* is an operation that occurs during the Control Information Download process. The device verifies that the configuration I/O bank voltages ( $V_{CCIO1}$  and  $V_{CCIO2}$ ) have reached their minimum recommended operating levels. Note that during the period from POR to when INITN asserts high, sysCONFIG pins and general purpose I/O pins may be in a tri-state condition. Once INITN asserts high, the internal weak pull-up or pull-down resistors for these pins are activated.
2. Refresh is either PROGRAMN pulsing or REFRESH command execution.

**Figure 4.1. Configuration Control Flow**

## 4.4. sysCONFIG Pins

Nexus 2 devices provide a set of I/O pins that are used to program and configure the FPGA. These pins are grouped together to create ports (such as JTAG, target SPI, and controller SPI) as shown in Figure 4.2. These ports are used to interact with the FPGA for programming and configuration, and to access resources within the FPGA.

### Notes:

- In this document, I/O pins used for programming and configuration are generally referred to as sysCONFIG pins.
- Unless otherwise specified, sysCONFIG pins are powered by the  $V_{CCIO1}$  and  $V_{CCIO2}$  voltages. This is an important consideration when provisioning other logic attached to I/O bank 1 and I/O bank 2.

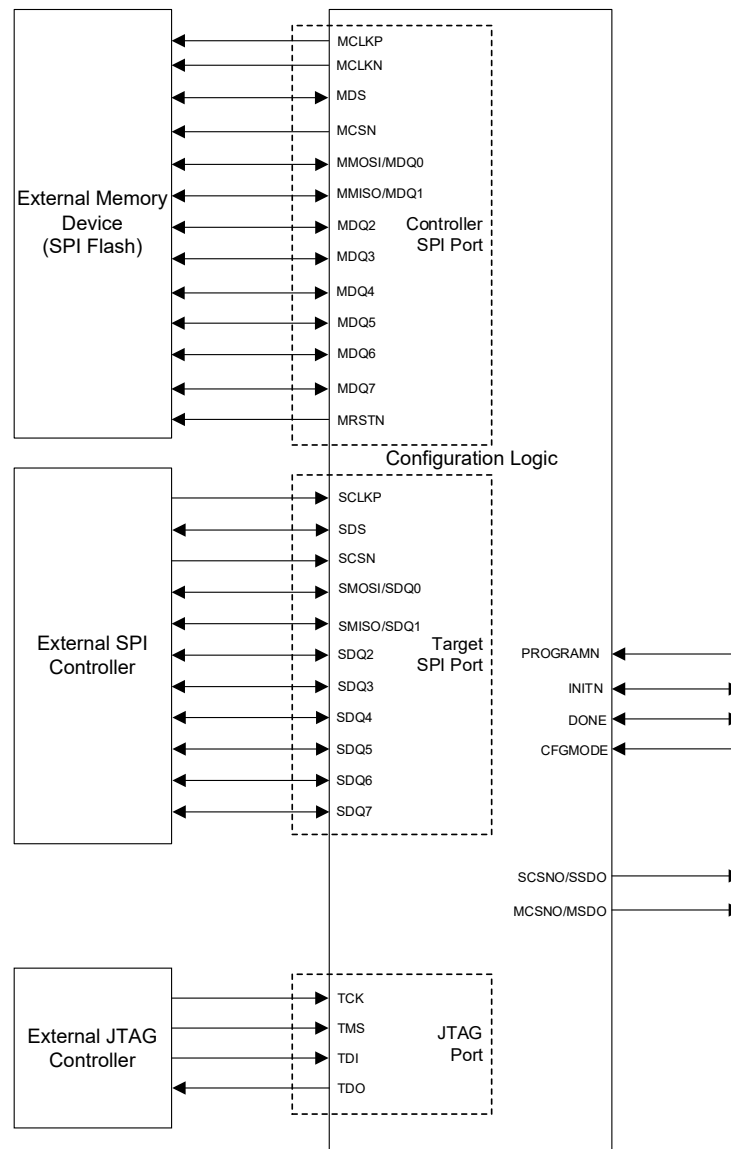


Figure 4.2. sysCONFIG Pins

When configuration of a Nexus 2 device is complete, the device enters user function mode. In user function mode, target SPI sysCONFIG pins default to become general purpose I/O pins. This means that the target SPI port cannot be used to configure the Nexus 2 device. To retain use of the target SPI port for device configuration or performing bitstream dry-run in user function mode, the port must be persisted.

The following are guidelines to configure dual-purpose I/O pins:

- To retain use of a configuration port in user function mode, port persistence must be enabled in the Lattice Radiant Device Constraint Editor under the Global tab. Refer to the [SLAVE\\_SPI\\_PORT](#) and [MASTER\\_SPI\\_PORT](#) sections.
- To use dual-purpose pins as general purpose I/O (GPIO) pins in user function mode, the unused configuration port must be disabled in the Lattice Radiant Device Constraint Editor under the Global tab. Refer to the [SLAVE\\_SPI\\_PORT](#) and [MASTER\\_SPI\\_PORT](#) sections.
- External logic must be prevented from interfering with device programming. If dual-purpose I/O pins are being used as general purpose I/O pins and not for configuration, any external logic connected to these pins should not toggle the pins in a way that could accidentally mimic configuration commands before the device enters user function mode.

## 4.5. sysCONFIG Pin List

Table 4.3 lists the sysCONFIG pins of the device and the default states of these pins.

**Table 4.3. Default State of sysCONFIG Pins**

| Group          | sysCONFIG Pins    |          |                     |                           |                            | Pull During Configuration | Configuration Modes |                   |                   |
|----------------|-------------------|----------|---------------------|---------------------------|----------------------------|---------------------------|---------------------|-------------------|-------------------|
|                | Name              | Location | Type                | Unprogrammed Mode Default | User Function Mode Default |                           | Controller SPI      | Target SPI        | JTAG              |
| System         | CFGMODE           | Bank 2   | Dedicated           | CFGMODE                   | CFGMODE                    | UP                        | 1'b1                | 1'b0 <sup>1</sup> | 1'b0 <sup>1</sup> |
|                | PROGRAMN          | Bank 2   | Dedicated           | PROGRAMN                  | PROGRAMN                   | UP                        | 1'b1                | 1'bX              | 1'bX              |
|                | INITN             | Bank 2   | Dedicated           | INITN                     | INITN                      | UP                        | INITN               |                   |                   |
|                | DONE              | Bank 2   | Dedicated           | DONE                      | DONE                       | UP                        | DONE                |                   |                   |
|                | MCSNO/MSDO        | Bank 1   | Shared <sup>2</sup> | MCSNO/MSDO                | GPIO                       | UP                        | MCSNO/MSDO          | —                 | —                 |
|                | SCSNO/SSDO        | Bank 2   | Shared <sup>2</sup> | MCSNO/MSDO                | GPIO                       | UP                        | —                   | SCSNO/SSDO        | —                 |
| Controller SPI | MCLKP             | Bank 1   | Shared <sup>2</sup> | MCLKP                     | GPIO                       | UP/DOWN <sup>3</sup>      | MCLKP               | —                 | —                 |
|                | MCLKN             | Bank 1   | Shared <sup>2</sup> | MCLKN                     | GPIO                       | UP/DOWN <sup>4</sup>      | MCLKN               | —                 | —                 |
|                | MCSN <sup>5</sup> | Bank 1   | Shared <sup>2</sup> | MCSN                      | GPIO                       | UP                        | MCSN                | —                 | —                 |
|                | MMOSI/MDQ0        | Bank 1   | Shared <sup>2</sup> | MMOSI/MDQ0                | GPIO                       | UP                        | MMOSI/MDQ0          | —                 | —                 |
|                | MMISO/MDQ1        | Bank 1   | Shared <sup>2</sup> | MMISO/MDQ1                | GPIO                       | UP                        | MMISO/MDQ1          | —                 | —                 |
|                | MDQ[7:2]          | Bank 1   | Shared <sup>2</sup> | MDQ[7:2]                  | GPIO                       | UP                        | MDQ[7:2]            | —                 | —                 |
|                | MDS               | Bank 1   | Shared <sup>2</sup> | MDS                       | GPIO                       | DOWN                      | MDS                 | —                 | —                 |
| Target SPI     | MRSTN             | Bank 1   | Shared <sup>2</sup> | MRSTN                     | GPIO                       | UP                        | MRSTN               | —                 | —                 |
|                | SCLKP             | Bank 2   | Shared <sup>2</sup> | SCLKP                     | GPIO                       | UP/DOWN <sup>3</sup>      | —                   | SCLKP             | —                 |
|                | SCSN <sup>6</sup> | Bank 2   | Shared <sup>2</sup> | SCSN                      | GPIO                       | UP                        | —                   | SCSN              | —                 |
|                | SMOSI/SDQ0        | Bank 2   | Shared <sup>2</sup> | SMOSI/SDQ0                | GPIO                       | UP                        | —                   | SMOSI/SDQ0        | —                 |
|                | SMISO/SDQ1        | Bank 2   | Shared <sup>2</sup> | SMISO/SDQ1                | GPIO                       | UP                        | —                   | SMISO/SDQ1        | —                 |
|                | SDQ[7:2]          | Bank 2   | Shared <sup>2</sup> | SDQ[7:2]                  | GPIO                       | UP                        | —                   | SDQ[7:2]          | —                 |
|                | SDS               | Bank 2   | Shared <sup>2</sup> | SDS                       | GPIO                       | DOWN                      | —                   | SDS               | —                 |
| JTAG           | TCK               | Bank 2   | Dedicated           | TCK                       | TCK                        | DOWN                      | —                   | —                 | TCK               |
|                | TMS               | Bank 2   | Dedicated           | TMS                       | TMS                        | UP                        | —                   | —                 | TMS               |
|                | TDI               | Bank 2   | Dedicated           | TDI                       | TDI                        | UP                        | —                   | —                 | TDI               |
|                | TDO               | Bank 2   | Dedicated           | TDO                       | TDO                        | UP                        | —                   | —                 | TDO               |

**Notes:**

1. CFGMODE should be set to 1'b0 when configuring the device through target SPI or JTAG after power-up or a refresh event. In user function mode, background programming of non-volatile memory through the target SPI or JTAG port can be performed regardless of the CFGMODE pin state.
2. Dual-purpose pin that may be used as a sysCONFIG pin or general purpose I/O pin in user function mode. This is user-selectable through the Lattice Radiant Device Constraint Editor. Achieved through SRAM fuse setting in bitstream.
3. Internal weak pull-up or pull down is determined by the CR1 bit 20 (CPOL) setting. UP if CPOL = 1; DOWN if CPOL = 0 (default).
4. Internal weak pull-up or pull down is determined by the CR1 bit 20 (CPOL) setting. DOWN if CPOL = 1; UP if CPOL = 0 (default).
5. MCSN should have 4.7 kΩ pull-up on-board resistor for controller SPI.
6. SCSN should have 4.7 kΩ pull-up resistor on-board for target SPI.

## 4.6. System Pins

System pins are dedicated I/O pins.

### 4.6.1. CFGMODE

CFGMODE is an input used to select the configuration mode. The CFGMODE pin is sampled when the device enters the configuration phase. If the CFGMODE pin is high, the device automatically boots from external flash through the controller SPI port or from internal flash; if the CFGMODE pin is low, the device waits for a target request from either the JTAG or target SPI port. Refer to the [Configuration Ports Arbitration](#) section for more information.

The following are conditions to be aware of in relation to the CFGMODE pin:

- Toggling the CFGMODE pin during the configuration phase aborts the configuration process.
- Toggling the CFGMODE pin does not affect the device in user function mode. In user function mode, background programming of external or internal flash memory through the target SPI or JTAG port can be performed independent of the CFGMODE pin. For external flash programming, the controller SPI port persistence must be enabled.

### 4.6.2. PROGRAMN

PROGRAMN is an input used to configure the FPGA. The PROGRAMN pin is low-level sensitive and has an internal weak pull-up. When PROGRAMN is asserted low, the FPGA exits user function mode and starts the initialization phase of the device configuration process. PROGRAMN must be asserted low for a minimum period of  $t_{\text{PROGRAMN}}$  for it to be recognized by the FPGA. This minimum time is defined in the sysCONFIG Port Timing Specifications of the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).

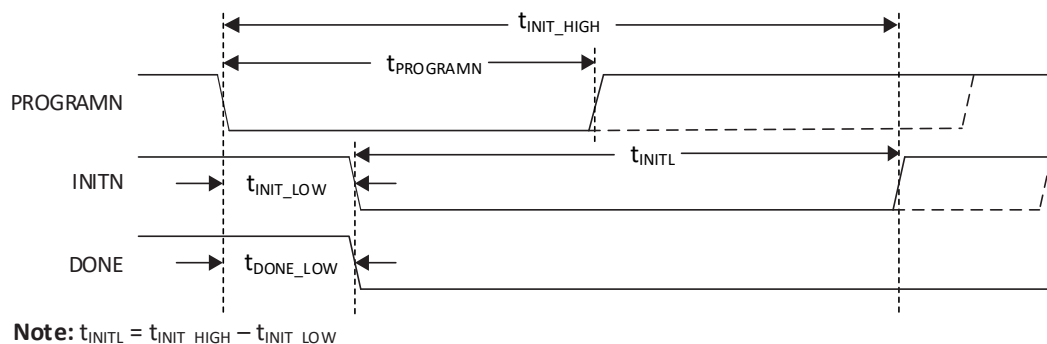


Figure 4.3. Configuration from PROGRAMN Timing

The following are conditions to be aware of in relation to the PROGRAMN pin:

- Toggling the PROGRAMN pin during the initialization phase ( $t_{\text{INITL}}$  period) does not disrupt the configuration process.
- Toggling the PROGRAMN pin during device configuration interrupts the process and restarts the configuration process.

### 4.6.3. INITN

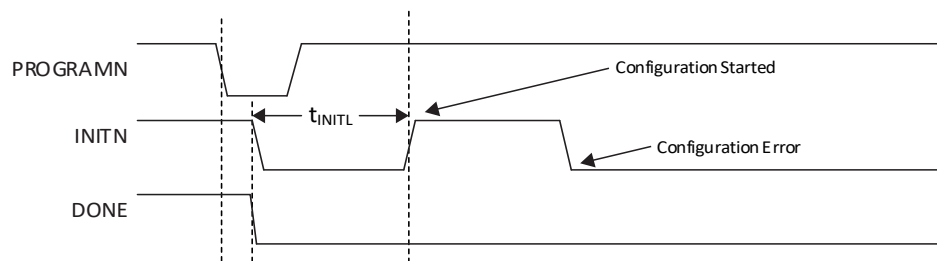
The INITN pin is a bi-directional open-drain control pin. The following conditions cause INITN to toggle low:

- Power is applied (power-up).
- PROGRAMN pin is pulsed (falling-edge has occurred).
- REFRESH command is received through a configuration port (JTAG or target SPI).

INITN toggles low, after a specified period of  $t_{\text{INIT\_LOW}}$ , to indicate that the initialization phase is in progress. After the  $t_{\text{INITL}}$  period has elapsed, the INITN pin is de-asserted (toggles high) to indicate that the device is ready to accept configuration data. The device begins loading configuration data from an external or internal memory device.

INITN can be asserted low by an external agent before the  $t_{\text{INITL}}$  period has elapsed (in effect holding INITN low) to prevent the FPGA from reading configuration data. This is useful when there are multiple programmable devices

chained together. The programmable device with the longest  $t_{INITL}$  period can hold all other devices in the chain from accepting configuration data until it is ready itself.



**Figure 4.4. Configuration Error Notification**

Once the  $t_{INITL}$  period has elapsed and INITN is de-asserted, INITN functions as an error signal. Any subsequent assertion of INITN indicates that the device has detected an error during configuration. The following are conditions that can cause device configuration to fail:

- Device ID mismatch is detected.
- Bitstream CRC error is detected.
- Invalid command error is detected.
- A preamble timeout error is encountered when loading from the external or internal memory device. This can occur when the device is in controller SPI configuration mode or self-download mode and the external or internal memory device is not programmed.
- The program done command is not received at the end of on-chip SRAM configuration.

When an error is detected during device configuration as indicated by INITN, the internal DONE bit is not set and the DONE pin remains low so the device does not wake up. The error can be cleared by correcting the configuration bitstream and forcing the FPGA back into the initialization phase.

The INITN pin must be pulled high by an external resistor (4.7 to 10 kΩ recommended) when initialization is complete.

#### 4.6.4. DONE

The DONE pin is a bi-directional open-drain pin with internal weak pull-up. The DONE pin is asserted low in tandem with the INITN pin when the FPGA enters the initialization phase. After an internal DONE status bit is set, the active-high DONE signal is used to indicate whether the FPGA is in user function mode. Setting the internal DONE status bit marks the beginning of the FPGA wake-up phase. The DONE pin is released high when the FPGA enters user function mode.

The FPGA can be prevented from entering user function mode indefinitely by having an external agent keep the DONE pin asserted low. An FPGA is ready to start operation only after DONE toggles high. A common reason for keeping DONE held low is to allow multiple FPGAs to finish configuration so that operation can start in unison only after configuration of the last FPGA. An external device can check if an FPGA has finished configuration by reading the DONE pin.

#### 4.6.5. MCSNO/MSDO

On the controller SPI port, MCSNO/MSDO is an output pin used for the following purposes:

- MCSNO – For configuration daisy chaining implemented with the flow-through attribute. This attribute allows the MCSNO pin to be driven when the DONE bit is set and configuration of the first device is complete. The MCSNO of the first device drives the CSN of the second device.
- MSDO – The MSDO pin is used in bypass mode. It is the serial data output for the downstream device that supports the legacy serial configuration mode (SCM).

#### 4.6.6. SCSNO/SSDO

On the target SPI port, SCSNO/SSDO is an output pin used for the following purposes:

- SCSNO – For configuration daisy chaining implemented with the flow-through attribute. This attribute allows the SCSNO pin to be driven when the DONE bit is set, and configuration of the first device is complete. SCSNO of the first device drives the CSN of the second device.
- SSDO – Used in the bypass mode. It is the serial data output for the downstream device that supports the legacy serial configuration mode.

### 4.7. Controller SPI sysCONFIG Pins

The following subsections describe the controller SPI sysCONFIG pins. These pins are dual-purpose I/O pins. Follow the guidelines presented in the [sysCONFIG Pins](#) section to use these pins as either sysCONFIG pins or general purpose I/O pins in user function mode. These pins are powered by the  $V_{CCIO1}$  voltage. If the external memory needs to be accessed using the controller SPI port while the device is in user function mode, the MASTER\_SPI\_PORT option must be enabled to persist these pins.

#### 4.7.1. MCLKP

On the controller SPI port, MCLKP is the output clock signal used to drive an external or internal memory device to sequentially load configuration data for the FPGA. Several different output clock frequencies are supported. The maximum MCLKP frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing specifications in the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121). MCLKP actively drives an external or internal memory device until all the configuration data is received. When the device enters user function mode, the MCLKP output tri-states. In most post-configuration applications, MCLKP is used as the reference clock for performing memory transactions with the external or internal memory device. Refer to the [Controller SPI Mode](#) section for details.

The Nexus 2 device generates MCLKP from an internal oscillator. The initial frequency of MCLKP is nominally 3.1 MHz. The MCLKP frequency can be altered using the MCCLK\_FREQ option. The MCCLK\_FREQ option is selected using the Lattice Radiant Device Constraint Editor. For a complete list of the supported MCLKP frequencies, refer to [Table 7.5](#).

At startup, the lowest frequency MCLKP is used by the FPGA. During the initial stages of device configuration, the frequency value specified using MCCLK\_FREQ contained in the bitstream is loaded into the FPGA. Once the device accepts the new MCCLK\_FREQ value, the MCLKP output begins driving the selected frequency. When selecting the MCLKP frequency, do not exceed the frequency specification of the configuration memory or the PCB. When making decisions on MCCLK\_FREQ value, first review the sysCONFIG Port Timing specifications in the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).

#### 4.7.2. MCLKN

MCLKN is the inverted MCLKP signal for differential clocking in controller xSPI (x8 DTR) mode. This document may only refer to the MCLKP signal for simplicity, but wherever MCLKP is described or illustrated, the differential pair of MCLKP and MCLKN may be used.

#### 4.7.3. MCSN

On the controller SPI port, MCSN becomes an active-low chip-select output that drives the SPI serial flash chip select. In user function mode, MCSN is a general purpose I/O with weak pull-down. Adding a 4.7 k $\Omega$  to 10 k $\Omega$  pull-up resistor to the MCSN pin on the Nexus 2 device is recommended. MCSN must ramp in tandem with the SPI flash  $V_{CC}$  input.

#### 4.7.4. MMOSI/MDQ0

On the controller SPI port, the MOSI pin is the serial data output for SPI command and data. It becomes D0 of the data bus in dual, quad, or xSPI mode.

#### 4.7.5. MMISO/MDQ1

On the controller SPI port, the MISO pin is the serial data input. It becomes D1 of the data bus in dual, quad, or xSPI mode.

#### 4.7.6. MDQ[7:2]

On the controller SPI port, MDQ[3:2] becomes D[3:2] of the data bus in quad mode, or MDQ[7:2] becomes D[7:2] of the data bus in xSPI mode.

#### 4.7.7. MDS

On the controller SPI port, MDS becomes the data strobe of the data bus in xSPI mode.

#### 4.7.8. MRSTN

On the controller SPI port, MRSTN is the active-low output for SPI flash hardware RESET. When the MSPI\_RESET\_PORT option is set to ENABLE, the MRSTN pin outputs a pulse (active low) to reset the flash device into x1 SPI mode. A pulse is triggered by each of the following events:

- Power up
- PROGRAMN pin pulsing
- REFRESH command execution
- Start of a bitstream load (primary or secondary image)

### 4.8. Target SPI sysCONFIG Pins

The following sub-sections discuss the target SPI sysCONFIG pins. These pins are dual-purpose I/O pins. Follow the guidelines presented in the [sysCONFIG Pins](#) section to use these pins as either sysCONFIG pins or general purpose I/O pins in user function mode. These pins are powered by the V<sub>CCIO2</sub> voltage.

#### 4.8.1. SCLKP

On the target SPI port, SCLKP is the clock input for the target SPI configuration interface. The maximum SCLKP frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing Specifications of the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).

#### 4.8.2. SCSN

On the target SPI port, SCSN is the active-low chip-select input for the target SPI configuration interface. Adding a 4.7 kΩ external pull-up resistor to the SCSN pin is recommended.

#### 4.8.3. SMOSI/SDQ0

On the target SPI port, the MOSI pin is the serial data input for SPI command and data. It becomes D0 of the data bus in dual, quad, or xSPI mode.

#### 4.8.4. SMISO/SDQ1

On the target SPI port, the MISO pin is the serial data output for SPI data. It becomes D1 of the data bus in dual, quad, or xSPI mode.

#### 4.8.5. SDQ[7:2]

On the target SPI port, SDQ[3:2] becomes D[3:2] of the data bus in quad mode, or SDQ[7:2] becomes D[7:2] of the data bus in xSPI mode.

#### 4.8.6. SDS

On the target SPI port, SDS becomes the data strobe of the data bus in xSPI mode.

## 4.9. JTAG Pins

JTAG pins are dedicated I/O pins.

### 4.9.1. TCK

The TCK pin serves as the test clock pin (TCK) for the JTAG interface or test access port (TAP). It provides the clock used to time the other JTAG port pins. Data is shifted into the instruction or data registers on the rising edge of TCK and shifted out on the falling edge of TCK. TAP is a static design permitting TCK to be stopped in either the high or low state. The maximum input frequency for TCK is specified in Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121). An internal pull-down resistor on the TCK pin is provided.

### 4.9.2. TMS

The TMS pin serves as the test mode select (TMS) pin for the JTAG interface. It is an input pin that controls the progression through the IEEE 1149.1-compliant state machine states. The TMS pin is sampled on the rising edge of TCK. The JTAG state machine remains in or transitions to a new TAP state depending on the current state of the TAP and the present state of the TMS input. An internal pull-up resistor on the TMS pin is provided according to the JTAG specification.

### 4.9.3. TDI

The TDI pin serves as the test data input (TDI) pin for the JTAG interface. It is used to shift in serial test instructions and data. This pin should be wired to TDI of the JTAG connector, or to TDO of an upstream device in a JTAG chain. An internal pull-up resistor on the TDI pin is provided.

### 4.9.4. TDO

The TDO pin serves as the test data output (TDO) pin for the JTAG interface. It is used to shift out serial test instructions and data. When TDO is not being driven by the internal circuitry, the pin is in a high-impedance state. The only time TDO is not in a high-impedance state is when the JTAG state machine is in the Shift-IR or Shift-DR state. This pin should be wired to TDO of the JTAG connector, or to TDI of a downstream device in a JTAG chain. An internal pull-up resistor on the TDO pin is provided.

## 4.10. Port Persistence

Port persistence allows configuration ports to be used in user function mode. This is determined by internal PERSISTENT control bits set through port persistence options.

### 4.10.1. PERSISTENT Control Bits

The internal PERSISTENT control bits are used to determine whether the dual-purpose controller and target SPI sysCONFIG pins remain as sysCONFIG pins during normal operation (in user function mode), for example, to support transparent programming, dry run, or configuration. Nexus 2 devices have several PERSISTENT physical SRAM cells that determine the existence of the controller SPI port or target SPI port after the device enters user function mode.

### 4.10.2. sysCONFIG Port Persistence Options

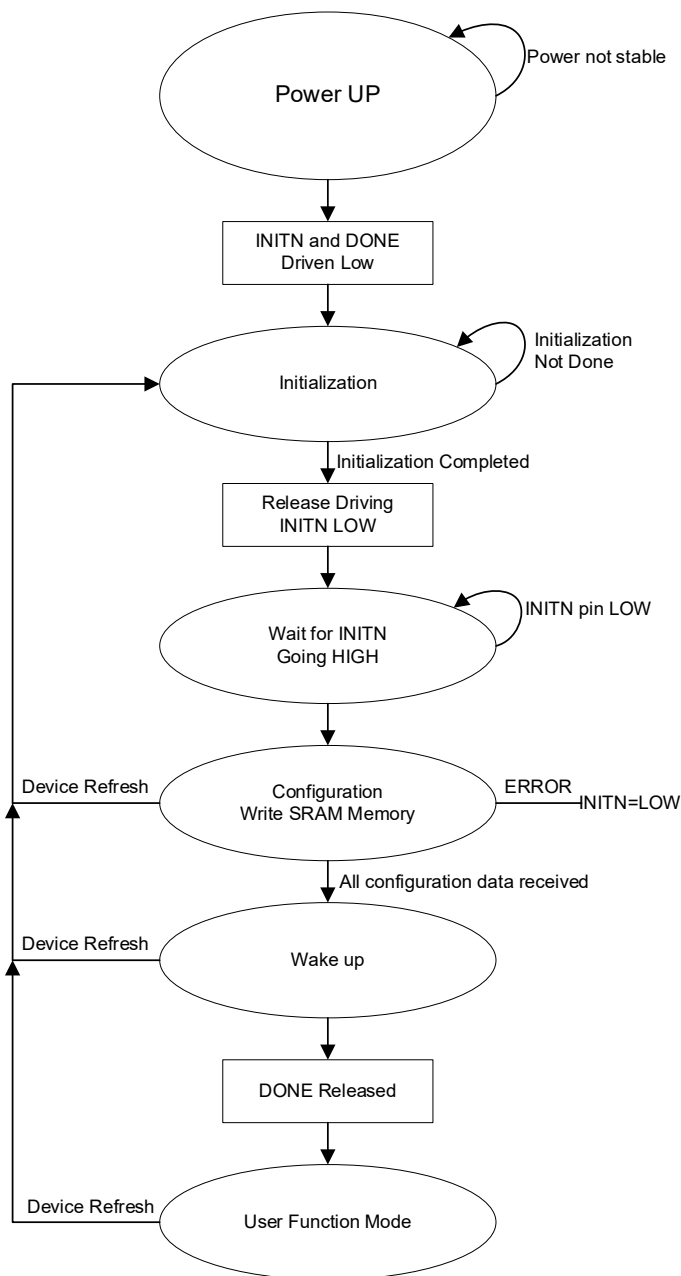
Table 4.4 lists the sysCONFIG port persistence options. The port persistence settings can be set in the Lattice Radiant Device Constraint Editor under the Global tab.

**Table 4.4. sysCONFIG Port Persistence Options**

| Option Name     | Setting       | Pins Affected                                             | Description                                      |
|-----------------|---------------|-----------------------------------------------------------|--------------------------------------------------|
| MASTER_SPI_PORT | SERIAL        | MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1                       | If enabled, persisted for configuration purpose. |
|                 | DUAL          | MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1                       |                                                  |
|                 | QUAD          | MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[3:2]             |                                                  |
|                 | XSPI          | MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[7:2], MDS        |                                                  |
|                 | XSPI_DIFF_CLK | MCLKP, MCLKN, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[7:2], MDS |                                                  |
| SLAVE_SPI_PORT  | SERIAL        | SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1                       | If enabled, persisted for configuration purpose. |
|                 | DUAL          | SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1                       |                                                  |
|                 | QUAD          | SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1, SDQ[3:2]             |                                                  |
|                 | XSPI          | SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1, SDQ[7:2], SDS        |                                                  |

## 5. Configuration Process and Flow

Before entering user function mode, the Nexus 2 device goes through a sequence of phases, including initialization, configuration, and wake-up.



**Figure 5.1. Configuration Flow**

## 5.1. Power-Up Sequence

For the Nexus 2 device to operate, power must be applied to the device. During a short period of time, as power supplies ramp, the FPGA stays in an indeterminate state. As power ramp-up continues, a power-on reset (POR) circuit inside the FPGA becomes active. Once active, the POR circuit ensures that the external I/O pins are in a high-impedance state. It also monitors the  $V_{CC}$ ,  $V_{CCAUX}$ ,  $V_{CCIO1}$ , and  $V_{CCIO2}$  input rails. When these power supplies reach the minimum operation level internally, the POR circuit releases an internal reset strobe allowing the device to begin its initialization process. The Nexus 2 device drives INITN and DONE low. When INITN and DONE are asserted low, the device enters the initialization phase.

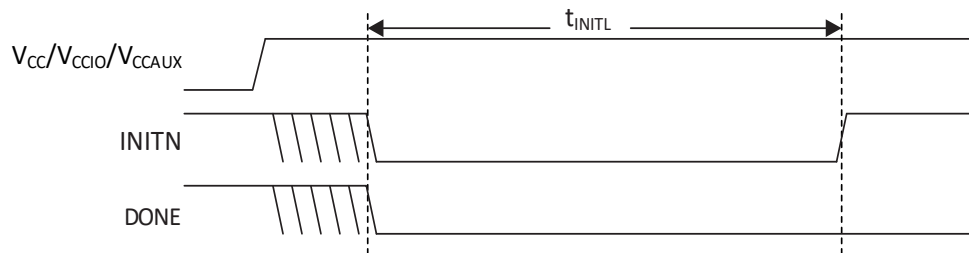


Figure 5.2. Configuration from POR Timing

## 5.2. Initialization

The Nexus 2 device enters the initialization phase immediately after the POR circuit drives INITN and DONE low. The purpose of the initialization phase is to clear the SRAM configuration memory of the FPGA.

The FPGA remains in the initialization phase until all the following conditions are met:

- The  $t_{INITL}$  period has elapsed.
- The PROGRAMN pin is de-asserted (high).
- The INITN pin is no longer asserted low by an external controller if applicable.

INITN has two functions during the initialization phase. The first is to indicate that the FPGA is currently clearing its configuration SRAM. The second is to act as an input preventing the transition from the initialization phase to the configuration phase.

During the  $t_{INITL}$  period, the FPGA clears the configuration SRAM. When the Nexus 2 device is part of a chain of devices, each device has a different  $t_{INITL}$  initialization time. The FPGA with the slowest  $t_{INITL}$  parameter can prevent other devices in the chain from starting to configure. Prematurely driving INITN high in a multi-device chain may cause configuration of one or more chained devices to intermittently fail.

The active-low, open-drain initialization signal INITN must be pulled high by an external resistor when initialization is complete. To synchronize the configuration of multiple FPGAs, the INITN pins should be wired in a logical AND configuration. If at least one FPGA or an external device holds INITN low, the FPGA remains in the initialization phase.

The GPIO pins of the device default to tri-stated outputs with active weak pull-downs at power-up. This default avoids inadvertent effects of the inputs rising while powering up. In some cases, this can cause a problem if other connected devices on the board reset or trigger from an active-high signal.

## 5.3. Configuration

Releasing the INITN pin so that the signal goes high causes the Nexus 2 device to enter the configuration phase. The FPGA device can then accept a configuration bitstream created by the Lattice Radiant software. Depending on the CFGMODE pin, the device either enters controller SPI/self-download mode or target mode. Refer to the [Configuration Ports Arbitration](#) section for more information. During configuration, the dual-purpose sysCONFIG I/O pins have the pull condition specified in [Table 4.3](#).

**Note:** Only Certus-N2 devices support the controller SPI mode. Only Mach-N2 devices support the self-download mode.

### 5.3.1. Controller SPI Configuration

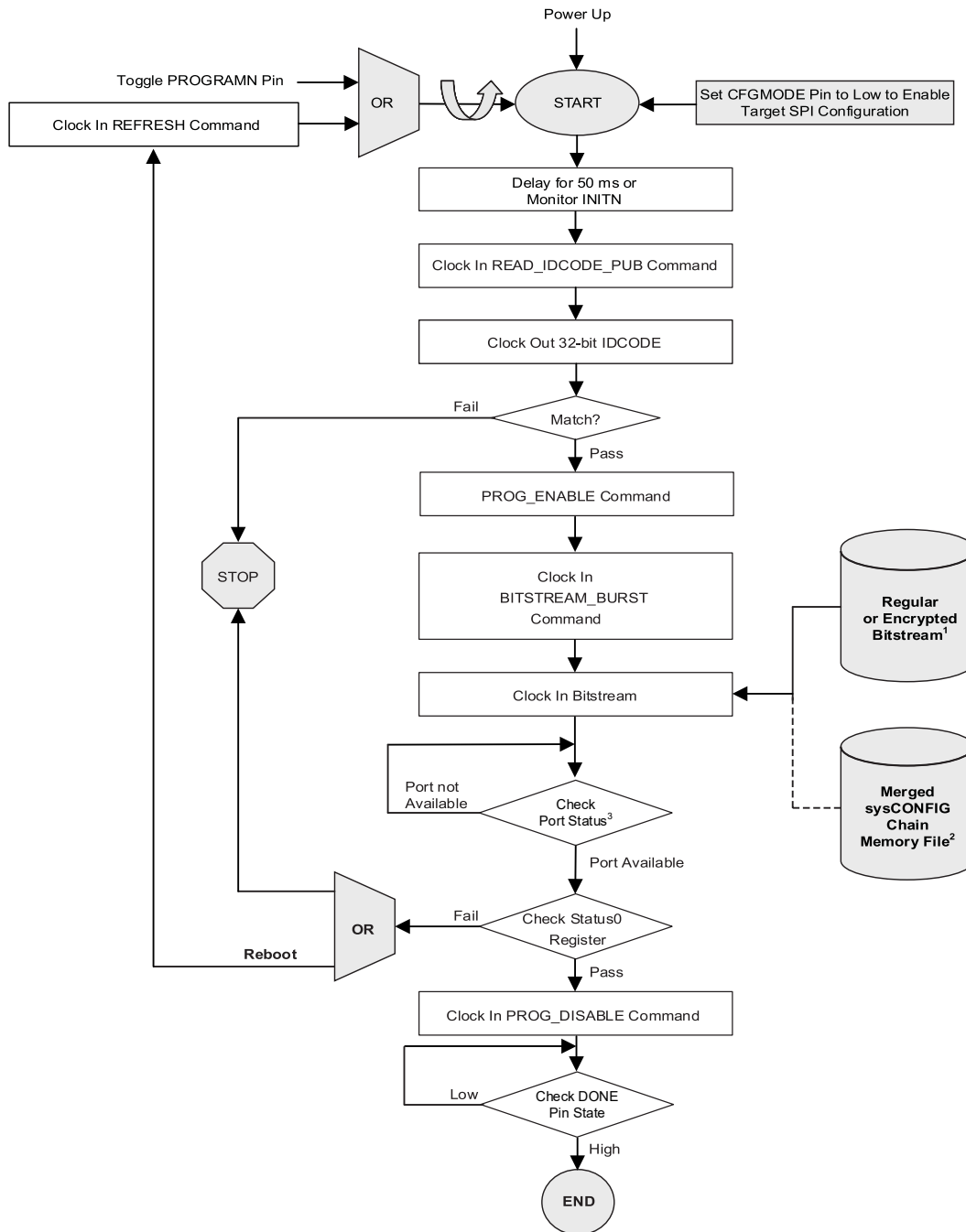
If CFGMODE is high, the Nexus 2 device enters the controller SPI mode and starts the external SPI flash boot process with signature verification. The device attempts a signature read-back in a finite loop until the expected result is received or the loop count is exceeded. A correct signature read allows the device to immediately proceed to preamble verification. This process allows for the fastest possible boot times.

The signature verification process verifies either the Lattice-specified LSCC signature or JEDEC standard SFDP code, depending on the value of bit 12 of Control Register 1 (CR1[12]). If CR1[12] is 0 (default), the Nexus 2 device reads the boot bitstream image from the base boot address (default is 0x000000 but this can be changed through OTP settings) and checks for the LSCC signature (0x4C534343) using SPI flash command code 8'H03. If CR1[12] is 1, the Nexus 2 device performs a SFDP read (SPI flash command code 8'H5A) and checks for the SFDP code (0x50444653) contained in SFDP-compliant SPI flash devices. The Nexus 2 device retries the SFDP/LSCC signature read until three consecutive matches are found. When successful, the Nexus 2 device sets the internal signature successful flag and proceeds to the preamble verification process. If the loop timer expires (about 200 ms), the device proceeds to the preamble verification process without setting the signature successful flag. When the internal signature successful flag is set, signature verification is bypassed for subsequent warm-boot events (for example, PROGRAMN pin pulsed or REFRESH command issued).

For proper bitstream data alignment, the bitstream preamble must be detected once. If the preamble does not come before the preamble timer (defined in Control Register 1 bit[2:0]) expires, a boot failure is declared, and the boot process aborts. Once the preamble is detected, the Nexus 2 device continues fetching data from non-volatile memory to configure the FPGA SRAM configuration memory. The Nexus 2 device does not leave the configuration phase if there is no valid configuration data. INITN is used as an error signal. INITN remains high if configuration proceeds without issues. INITN toggles low if a configuration error occurs.

### 5.3.2. Target SPI Configuration

If CFGMODE is low, the Nexus 2 device enters the target SPI mode and starts the process to configure the FPGA SRAM configuration memory from an external controller. Figure 5.3 shows the target SPI configuration flow.



**Notes:**

1. For a single FPGA device, the input file is a bitstream, which may be a standard or encrypted bitstream.
2. For a sysCONFIG chain of devices, the input file can be a merged memory file.
3. Check that the `cfg_mode` bit toggles from 1 to 0 to indicate that the configuration engine has switched from bitstream mode to command mode.

**Figure 5.3. Target SPI Configuration Flow**

### 5.3.3. Self-Download Mode Configuration

If CFGMODE is high, the Mach-N2 device enters the self-download mode and starts the internal flash boot process.

For proper bitstream data alignment, the bitstream preamble must be detected once. If the preamble does not come before the preamble timer (defined in Control Register 1 bit[2:0]) expires, a boot failure is declared, and the boot process aborts. Once the preamble is detected, the Mach-N2 device continues fetching data from internal flash memory to configure the FPGA SRAM configuration memory. The Mach-N2 device does not leave the configuration phase if there is no valid configuration data. INITN is used as an error signal. INITN remains high if configuration proceeds without issues. INITN toggles low if a configuration error occurs.

## 5.4. Wake-Up

The wake-up phase covers the transition from the configuration phase to user function mode. When configuration is complete (after configuration memory has been loaded), the device goes through a wake-up sequence involving a set of internal and external signals. The FPGA asserts an internal DONE status bit that starts the wake-up state machine to sequentially release four global control signals. The FPGA enters user function mode when the wake-up phase completes.

### 5.4.1. Wake-Up Signals

Table 5.1 lists the internal and external (global control) signals.

**Table 5.1. Wake-Up Signals**

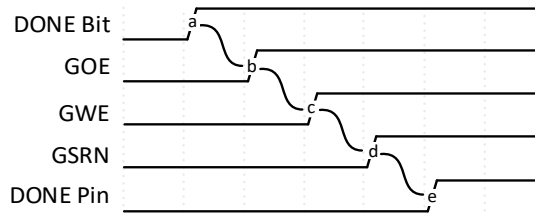
| Signal Name                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DONE                       | Internal DONE bit (STATUS0[7]) set by the PROG_DONE command at the end of the bitstream after all configuration data has been loaded.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| External DONE              | The external DONE pin is a bi-directional, open-drain I/O pin with a weak internal pull-up resistor when enabled. It can be connected to other FPGAs in a system to synchronize wake-up sequences. For example, an external agent can hold the external DONE pin low to prevent the wake-up process of the FPGA from proceeding until all FPGAs are ready. The wake-up phase completes when the external DONE signal toggles high, indicating that configuration is complete and that no errors were detected. Wake-up completes uninterrupted when the external DONE pin is not enabled. |
| Global Output Enable (GOE) | When GOE is low, the device I/O buffers are prevented from driving the pins. GOE controls the output drivers. When GOE asserts high, the device I/O pins exit the high-impedance state and take on their programmed output function.<br>The input drivers are enabled when the DONE bit is set. Therefore, the FPGA inputs are always active. However, the input signals are prevented from performing any action on the FPGA flip-flops by the assertion of GSRN.                                                                                                                        |
| Global Write Enable (GWE)  | GWE is a control signal that overrides the write enable strobe for all RAM logic inside the FPGA and is de-asserted (low) before device wake-up. Since the inputs on the FPGA are always active, keeping GWE de-asserted prevents accidental corruption of the instantiated RAM resources inside the FPGA, thereby safeguarding the integrity of the EBRs and LUTs in the device.                                                                                                                                                                                                         |
| Global Set/Reset (GSRN)    | GSRN is a control signal that, when asserted (low), causes all I/O flip-flops, look-up table (LUT) flip-flops, distributed RAM output flip-flops, and embedded block RAM output flip-flops that have the <i>GSR enabled</i> attribute to be set or cleared per their hardware description language definition. GSRN is used to set and reset the core of the device when asserted and de-asserted, respectively. GSRN is asserted (low) during configuration and de-asserted (high) in the wake-up sequence. In user function mode, the user design controls the GSRN signal.             |

### 5.4.2. Wake-Up Sequence

The wake-up sequence depends on the DAISY\_CHAIN\_WAIT\_DONE setting.

If the Nexus 2 device is the only FPGA in a system or the last device in a daisy chain, the DAISY\_CHAIN\_WAIT\_DONE option should be set to DISABLED. In this case, the following wake-up sequence executes:

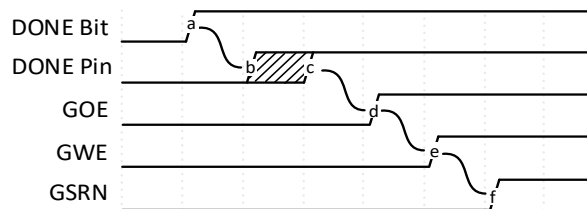
- DONE bit set to 1.
- Global Output Enable (GOE) signal asserts.
- Global Write Enable (GWE) signal asserts.
- Global Set/Reset (GSRN) signal de-asserts (high).
- DONE pin is released (high-impedance state). This causes the DONE signal to go high due to an internal or external pull-up resistor.



**Figure 5.4. Wake-Up Sequence with DAISY\_CHAIN\_WAIT\_DONE = DISABLED**

If the Nexus 2 device is in a daisy chain and is not the last device in the chain, the DAISY\_CHAIN\_WAIT\_DONE option should be set to ENABLED. In this case, the following wake-up sequence executes:

- DONE bit set to 1.
- DONE pin is released (high-impedance state). Other devices connected to the DONE pin may drive this pin low.
- Wait for DONE pin to go high, which indicates that all connected devices have been configured.
- Global Output Enable (GOE) signal asserts.
- Global Write Enable (GWE) signal asserts.
- Global Set/Reset (GSRN) signal de-asserts (high).



**Figure 5.5. Wake-Up Sequence with DAISY\_CHAIN\_WAIT\_DONE = ENABLED**

## 5.5. Early I/O Release

The Nexus 2 device supports an early I/O release feature, which allows I/O pins in the device to assume user-defined drive states at the beginning of bitstream processing. This feature is enabled by setting the EARLY\_IO preferences in the Port tab of the Lattice Radiant Device Constraint Editor.

In addition, early I/O release requires instantiating an output buffer register with an asynchronous set or reset function to indicate the desired drive 1 or drive 0 behavior, respectively, during the early release period. Unregistered outputs are in the high-impedance state until device configuration is complete. Be aware that some of the I/O pins in I/O bank 1 and I/O bank 2, including the dual-purpose sysCONFIG I/O pins, cannot be used as early I/O pins. If bitstream authentication and early I/O release are enabled in the Nexus 2 device, you must provide an AES key in addition to an ECDSA or RSA key.

## 5.6. User Function Mode

The Nexus 2 device enters user function mode immediately after the wake-up phase completes. In user function mode, the device begins performing its programmed logic operations as defined by the user design. All GPIO pins included in the user design wake up in the user-defined condition. GPIO pins not defined in the user design remain output tri-stated and input enabled with a weak pull-down.

The device remains in user function mode until one of the following events occurs:

- PROGRAMN pin is pulsed.
- REFRESH command is received through a configuration port.
- Power is cycled or power supply levels drop below their specified trigger levels.
- User watchdog timer is triggered. Refer to the [Watchdog Timer IP User Guide \(FPGA-IPUG-02097\)](#) for more information.

When one of these events occurs, the device exits user function mode and goes through the initialization, configuration, and wake-up phases again.

## 6. Device Configuration

The Nexus 2 device provides multiple options for loading configuration data into the SRAM configuration memory, either using external or internal non-volatile memory or through a target interface (target SPI, JTAG). This section describes the functionality of each of the different configuration modes, the status and control registers, and the configuration commands.

### 6.1. Configuration Modal States

Well-defined and predictable I/O behavior is of paramount importance to designers of boards and systems who must ensure that the power-up behavior and system initialization processes of their designs are robust. This is of particular concern where a design is programmable. It must exhibit robust behavior before, during, and after the programming process. The I/O controls of the configuration logic are based on these well-defined configuration states.

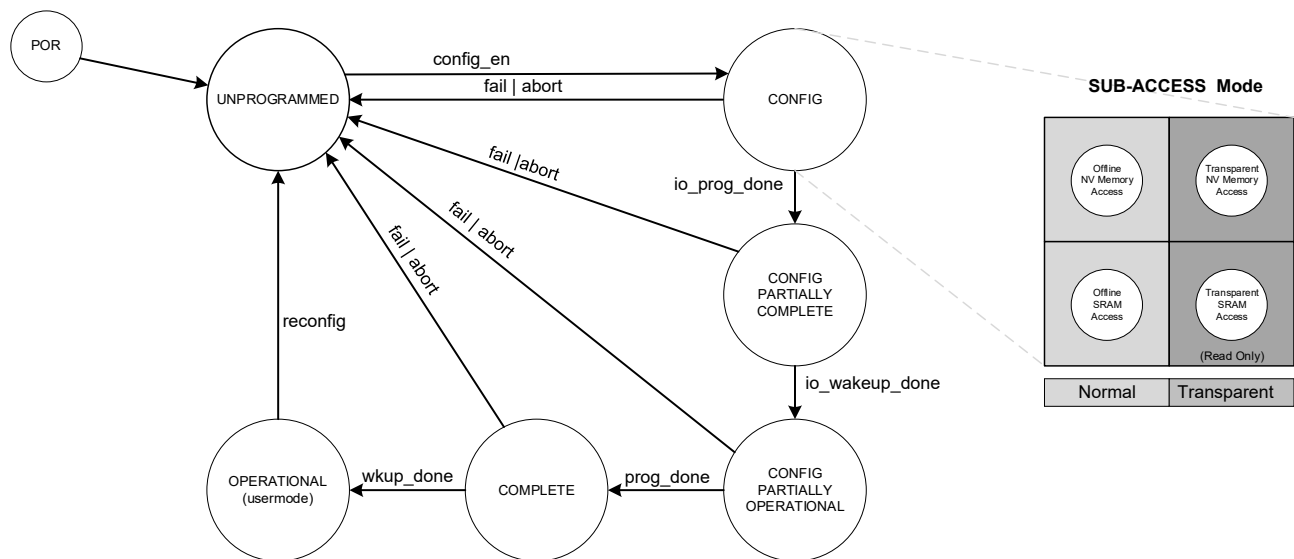


Figure 6.1. Configuration Modal States

- **Unprogrammed** – In this state, the device may be blank or incompletely programmed. The device system I/O pins are disabled (tri-stated) such that they cannot contend with output drivers that may exist in external devices connected to the device. Any program-qualified command will have its operation nullified if executed in this state. Upon power-up, incompletely programmed non-volatile devices, volatile devices, and blank devices should reside in this state. OTP programming can be done in this state.
- **Config** – In this state, config commands may be used (when permitted) to read, write, verify, protect, or erase the device. In normal sub-access mode, the device's system I/O pins are disabled (tri-stated). In transparent sub-access mode, the device's system I/O pins resume user function. Once programming is complete without error, the state transitions to the Config Partial Complete state.
- **Config Partial Complete** – This state exists so an external algorithm can control transitions from partially programmed to unprogrammed or partially operational. This state is temporary. Once the fabric is partially programmed without error and partial wake-up sequence is complete, the modal state will transition to the Config Partial Operational state. In this state, config could be waiting for bitstream authentication results from the security engine.
- **Config Partial Operational** – In this state, the device I/O pins are put into a state defined by their programming. The device's system I/O pins take on their programmed values high, low, or tri-state. Once the remainder of fabric is programmed without error, the state will transition to the Complete state.

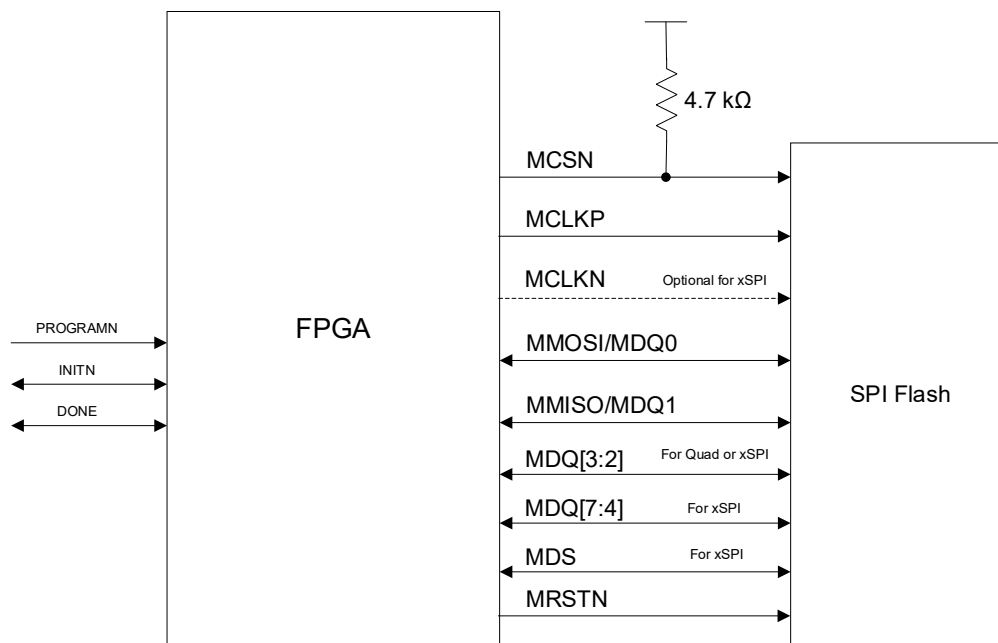
- Complete – This state exists so an external algorithm can control transitions to the Unprogrammed or Operational state after programming operations are complete. This state is temporary. This state will be entered once all programming is complete. In this state, config could be waiting for bitstream authentication results from the security engine.
- Operational – In this state, the device is ready for its *operational* mode defined by its programming. Any program qualified command will have its operation nullified if executed in this state. OTP programming and transparent programming can be done in this state.

## 6.2. Controller SPI Mode

**Note:** Only Certus-N2 devices support the controller SPI mode.

In controller SPI mode, the Nexus 2 device begins retrieving configuration data from the SPI flash when power is applied, the PROGRAMN pin is pulsed, or a REFRESH command is received. The MCLKP pin begins driving a nominal 3.1 MHz clock to the SPI flash device SCLKP input. MCSN is asserted low, commands are transmitted to the SPI flash over the MMOSI/MDQ0 output pin, and data is read on the MMISO/MDQ1 input pin. When all the configuration data has been retrieved, the MCSN pin is de-asserted and the controller SPI output pins are tri-stated.

Configuration data always starts downloading at the nominal MCLKP frequency of 3.1 MHz. The MCCLK\_FREQ option, accessed using the Device Constraint Editor, can be used to increase the frequency. Setting this option adds configuration commands to the configuration data in the SPI flash device. After reading the MCLKP configuration commands, the Nexus 2 device reads the remaining configuration data bytes using the new MCLKP frequency.



**Note:** For x1 mode only, MMOSI and MMISO are uni-directional.

**Figure 6.2. Nexus 2 Controller SPI Port with SPI Flash**

Once the SPI flash contains the configuration data, the configuration can be tested by pulsing the PROGRAMN pin, transmitting a REFRESH command, or cycling power to the board. The Nexus 2 device then configures from the external SPI flash.

## 6.2.1. Controller SPI Configuration Port Pins

Table 6.1 lists the controller SPI configuration port pins.

**Table 6.1. Controller SPI Configuration Port Pins**

| Pin Name          | Function | Direction    | Description                                                                                                                                                    |
|-------------------|----------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCLKP             | CLK      | Output       | Controller clock used to time data transmission/reception between the Nexus 2 device configuration logic and an external SPI flash.                            |
| MCLKN             | CLKN     | Output       | Optional complementary controller clock output for differential clock support in xSPI (x8, DTR) mode.                                                          |
| MCSN <sup>1</sup> | CSN      | Output       | Chip select used to enable an external SPI flash containing configuration data.                                                                                |
| MMOSI/MDQ0        | MOSI     | Output       | Carries output data from the Nexus 2 device configuration logic to the external SPI flash.                                                                     |
|                   | D0       | Input/Output | Input/Output pin for bitstream data, used in dual, quad, or xSPI mode.                                                                                         |
| MMISO/MDQ1        | MISO     | Input        | Carries input data from the target SPI flash to the Nexus 2 device configuration logic.                                                                        |
|                   | D1       | Input/Output | Input/Output pin for bitstream data, used in dual, quad, or xSPI mode.                                                                                         |
| MDQ[3:2]          | D[3:2]   | Input/Output | Input/Output pins for bitstream data from SPI flash, used only in quad or xSPI mode.                                                                           |
| MDQ[7:4]          | D[7:4]   | Input/Output | Input/Output pins for bitstream data from xSPI flash, used only in xSPI mode.                                                                                  |
| MDS               | DS       | Input/Output | xSPI data strobe                                                                                                                                               |
| MRSTN             | Reset    | Output       | When the MSPI_RESET_PORT option is set to ENABLE, the MRSTN pin outputs a pulse (active low) to reset the flash device into x1 SPI mode after a trigger event. |

**Note:**

1. Use a 4.7 kΩ pull-up resistor.

## 6.2.2. Enabling Controller SPI Port in Configuration Phase

The controller SPI port is enabled by driving the CFGMODE pin high. When the device is powered up, the PROGRAMN pin is pulsed, or the REFRESH command is executed, the device starts the initialization phase of the device configuration process. After the initialization phase and as the device enters the configuration phase, the controller SPI port is selected as the configuration port when CFGMODE is high. Refer to the [Configuration Ports Arbitration](#) section for more information.

## 6.2.3. Enabling Controller SPI Port Persistence in User Function Mode

The controller SPI port can be persisted in user function mode by setting the desired value (SERIAL, DUAL, QUAD, XSPI, or XSPI\_DIFF\_CLK) for MASTER\_SPI\_PORT in the Lattice Radiant Device Constraint Editor. Once set, the configuration bitstream contains optional controller SPI persistence bits. When the device completes configuration and wakes up, it checks the persistence bits to determine if the controller SPI port is to remain operational once in user function mode. This selection is independent of configuration port arbitration during the configuration phase. A port enabled by persistence is available in user function mode. The pins for this port are reserved from being occupied by user logic. Note that both the DONE pin and the INITN pin must be high, indicating that the device is in user function mode. Otherwise, the persistence bits have no effect.

### 6.2.4. Dual, Quad, and xSPI Controller SPI Read Mode

The controller SPI mode in the Nexus 2 device supports the industry-standard quad I/O SPI flash memory and octal I/O xSPI flash memory for faster read performance. To change the SPI read mode to fast read (dual, quad, or xSPI), the Lattice Radiant Deployment Tool must be used to generate the hex file for programming the SPI flash device. The Lattice Radiant flow only generates bitstreams with default SPI read mode, which is slow serial read (03h) mode. The sysCONFIG options (MASTER\_SPI\_PORT and SLAVE\_SPI\_PORT) in the Lattice Radiant Device Constraint Editor are used by the software to persist appropriate pins to help with the user design flow.

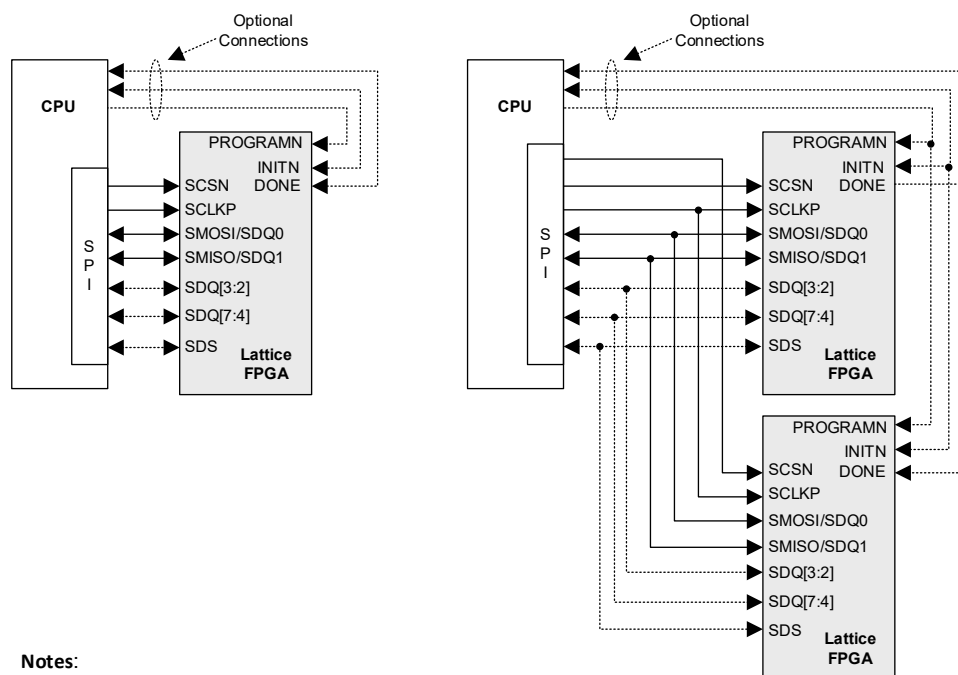
## 6.3. Target SPI Mode

The Nexus 2 device provides a target SPI configuration port to access features provided by the configuration logic. It supports reprogramming the configuration SRAM and accessing status/control registers within the configuration logic block.

In the target SPI mode, the SCLKP pin serves as the target SPI clock. Input data is read into the device on the SMOSI pin at the rising edge of SCLKP. Output data is valid on the SMISO pin at the falling edge of SCLKP. The SCSN acts as the chip select signal. When SCSN is high, the target SPI port is deselected and the SMISO pin is tri-stated and pulled up through an internal pull-up resistor. Commands can be written into and data read from the device when SCSN is asserted. SCSN should be de-asserted on a data byte boundary (in other words, the total number of clocks while SCSN is asserted should be a multiple of 8). If the total bits sent is not a multiple of 8, then the last incomplete byte is dropped. If the incomplete byte is part of a command (not an extra byte), then the device returns an error. If the incomplete byte is part of a command but an extra byte, then the command is processed.

Before the device enters user function mode, the target SPI port can be used when the CFGMODE pin is low. After the device enters user function mode, the target SPI port can be used if the port is persisted by setting the desired value (SERIAL, DUAL, QUAD, or xSPI) for SLAVE\_SPI\_PORT in the Lattice Radiant Device Constraint Editor.

Lattice provides C source code called sspiembedded to simplify the process of programming the Nexus 2 device through the target SPI port. Refer to the Lattice Radiant Online Help to learn more about sspiembedded.



#### Notes:

- The dotted lines indicate optional connections.
- The wake-up time of the device varies with the bitstream size and the speed of the SPI port. Lattice recommends connecting the DONE pin to the CPU to monitor when the configuration is complete.

**Figure 6.3. Nexus 2 Target SPI Port with CPU and Single or Multiple Devices**

### 6.3.1. Target SPI Configuration Port Pins

Table 6.2 lists the target SPI configuration port pins.

**Table 6.2. Target SPI Configuration Port Pins**

| Pin Name          | Function | Direction               | Description                                                                                                                                                                                                                      |
|-------------------|----------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCLKP             | CLK      | Input with weak pull-up | Clock used to time data transmission/reception from an external SPI controller device to the Nexus 2 device configuration logic.                                                                                                 |
| SCSN <sup>1</sup> | CSN      | Input with weak pull-up | Nexus 2 device configuration logic target SPI chip select input. CSN is an active-low input.<br>High to low transition – Starts transmitting a command.<br>Low to high transition – Completes or terminates the current command. |
| SMOSI/SDQ0        | MOSI     | Input                   | Carries output data from the external SPI controller to the Nexus 2 device configuration logic.                                                                                                                                  |
|                   | D0       | Input/Output            | D0 of the data bus for dual or quad mode.                                                                                                                                                                                        |
| SMISO/SDQ1        | MISO     | Input/Output            | Carries output data from the Nexus 2 device configuration logic to the external SPI controller. It is normally tri-stated with an internal pull-up. It becomes active only when the command is a read type command.              |
|                   | D1       | Input/Output            | D1 of the data bus for dual or quad mode.                                                                                                                                                                                        |
| SDQ[3:2]          | D[3:2]   | Input/Output            | D[3:2] of the data bus for quad or xSPI mode.                                                                                                                                                                                    |
| SDQ[7:4]          | D[7:4]   | Input/Output            | D[7:4] of the data bus for xSPI mode.                                                                                                                                                                                            |
| SDS               | DS       | Input/Output            | xSPI data strobe                                                                                                                                                                                                                 |

**Note:**

1. Use an external 4.7 kΩ pull-up resistor.

### 6.3.2. Enabling Target SPI Port in Configuration Phase

The target SPI port is enabled by driving the CFGMODE pin low. When the device is powered up, the PROGRAMN pin is pulsed, or the REFRESH command is executed, the device starts the initialization phase of the device configuration process. After the initialization phase and as the device enters the configuration phase, the target SPI port is selected as the configuration port when CFGMODE is low. Refer to the [Configuration Ports Arbitration](#) section for more information.

### 6.3.3. Enabling Target SPI Port Persistence in User Function Mode

The target SPI port can be persisted in user function mode by setting the desired value (SERIAL, DUAL, QUAD, or xSPI) for SLAVE\_SPI\_PORT in the Lattice Radiant Device Constraint Editor. Once set, the configuration bitstream contains optional target SPI persistence bits. When the device completes configuration and wakes up, it checks the persistence bits to determine if the target SPI port is to remain operational once in user function mode. This selection is independent of configuration port arbitration during the configuration phase. A port enabled by persistence is available in user function mode. The pins for this port are reserved from being occupied by user logic. Note that both the DONE pin and the INITN pin must be high to qualify the target SPI port as a read back port. Otherwise, the device is not in user function mode and the persistence bits have no effect.

### 6.3.4. Target SPI Port AC Timing Requirements

The target SPI port maximum operation frequency requirement is shown in Table 6.3.

**Table 6.3. Target SPI Port AC Timing Requirements**

| Description     | Parameter         | Min | Max | Unit |
|-----------------|-------------------|-----|-----|------|
| SCLKP Frequency | f <sub>CCLK</sub> | —   | 180 | MHz  |

For detailed AC timing requirements for the Nexus 2 target SPI configuration port, refer to the sysCONFIG Port Timing Specifications of the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).

### 6.3.5. Dual, Quad, and xSPI Target SPI Port

By default, the SPI port operates in serial mode with a data width of x1: one bit of input and one bit of output. However, to allow for faster loading of configuration information, wider SPI ports are supported. The Nexus 2 device target SPI port supports x2 (dual), x4 (quad), and xSPI (x8 DTR) modes of operation. For target SPI port, the dual, quad, and xSPI modes are selected using the SSPI\_MODE command. Refer to the [SSPI\\_MODE](#) section for more information.

At power-up, PROGRAMN pin pulsing, or REFRESH command execution, the target SPI port resets to the default serial (x1) mode.

### 6.3.6. Command Waveforms

#### 6.3.6.1. Read Command Waveforms

Read commands read data out from the Nexus 2 device. Bit 0 of the data or bitstream is read out first. After the 32-bit command, target SPI sends 1 on MISO until the return data is ready with one byte 0 as the data valid flag.

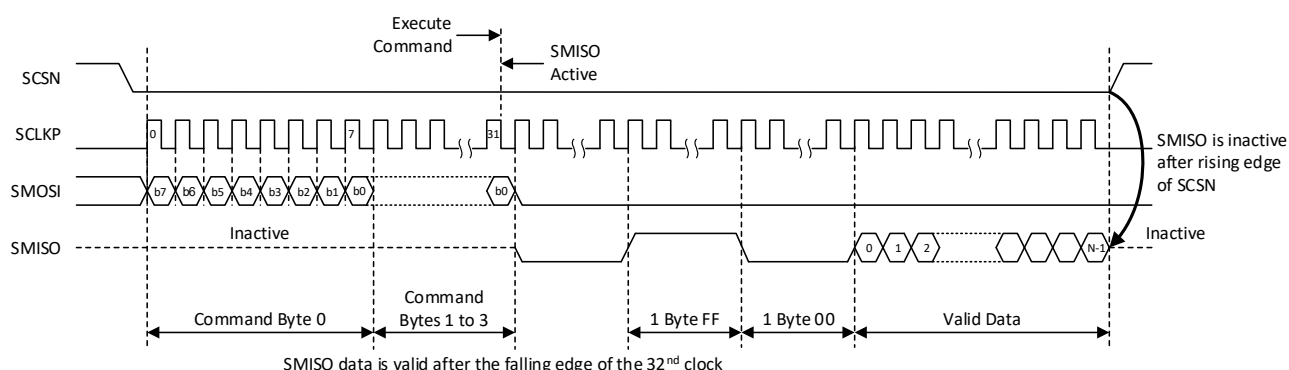


Figure 6.4. Read Command Waveforms

#### 6.3.6.2. Data Read Command Waveforms

Data Read commands include a data field immediately following the command and reads data out from the Nexus 2 device. The length of the command data depends on the command. See [Table 6.12](#) for the required data length for each command. After the command and command data are transmitted, target SPI sends 1 on MISO until the return data is ready with one byte 0 as the data valid flag.

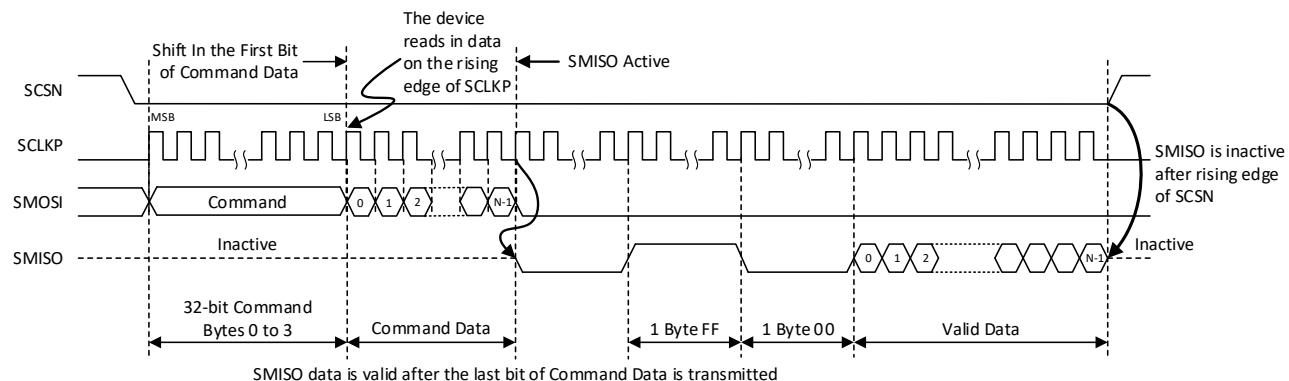
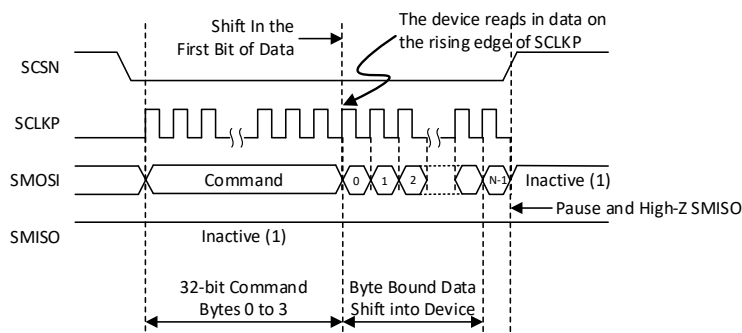


Figure 6.5. Data Read Command Waveforms

### 6.3.6.3. Data Write Command Waveforms

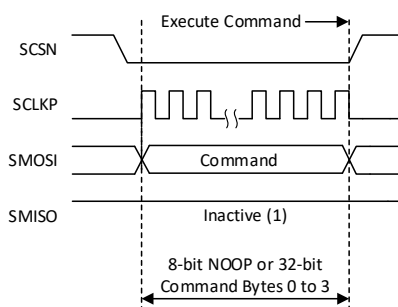
Data Write commands include a command data field immediately following the command. The length of the command data depends on the command. See [Table 6.12](#) for the required data length for each command.



### Figure 6.6. Data Write Command Waveforms

#### 6.3.6.4. Immediate Action (IMM) Command Waveforms

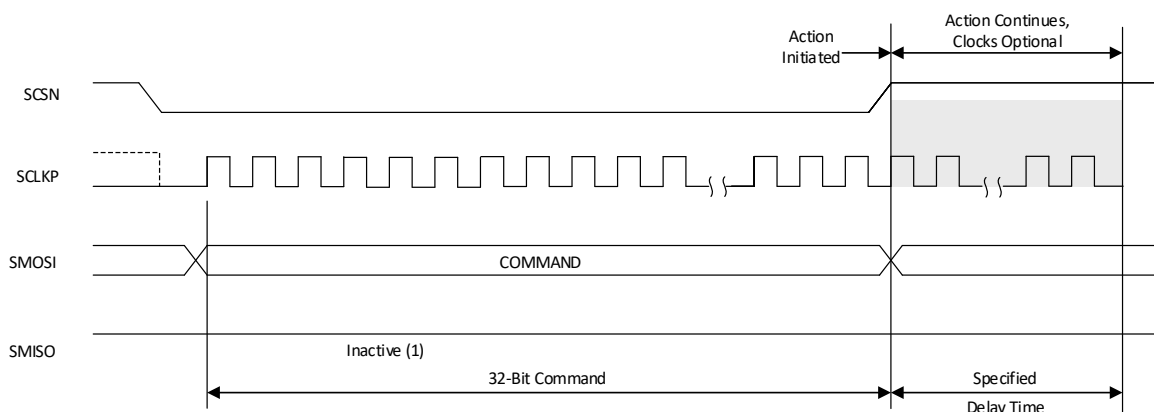
Immediate action commands do not require any data to be shifted in or out and execute immediately.



### Figure 6.7. Immediate Action Command Waveforms

#### 6.3.6.5. Delayed Action Command Waveforms

Delayed action commands do not need to shift data in or out and require additional time to execute the action associated with the command. After the last bit of the command is shifted in, the command starts execution and continues until the requested action completes. If the same command is sent a second time while the previous action is in progress, the new command is ignored. This type of command includes actions that may take some time to complete such as CALC SED CRC and REFRESH.



### Figure 6.8. Delayed Action Command Waveforms

### 6.3.7. Target SPI to Controller SPI Bridge

As the controller SPI port and target SPI port on the Nexus 2 device use separate pins, the Nexus 2 device provides the target SPI to controller SPI bridge, which enables programming the external SPI flash through the Nexus 2 device target SPI configuration port.

The MSPI\_BRIDGE command is used to enable this bridging. Once this bridging is enabled, any data following this command on the target SPI port is directed to the controller SPI port until the transaction is terminated by the external controller, by de-asserting the chip-select (CSN) signal of the target SPI. When CSN is de-asserted, the bridging function is disabled and normal target access for configuration continues. The functional diagram of the target SPI to controller SPI bridge is shown in Figure 6.9.

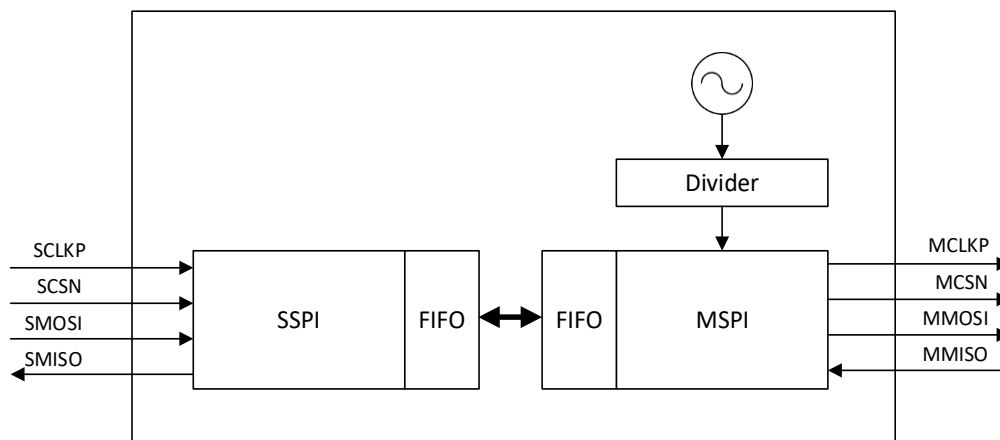


Figure 6.9. Target SPI to Controller SPI Bridge Functional Diagram

The default support for this bridging is serial data (x1) mode only. Figure 6.10 shows a block diagram example of target SPI to controller SPI bridge use.

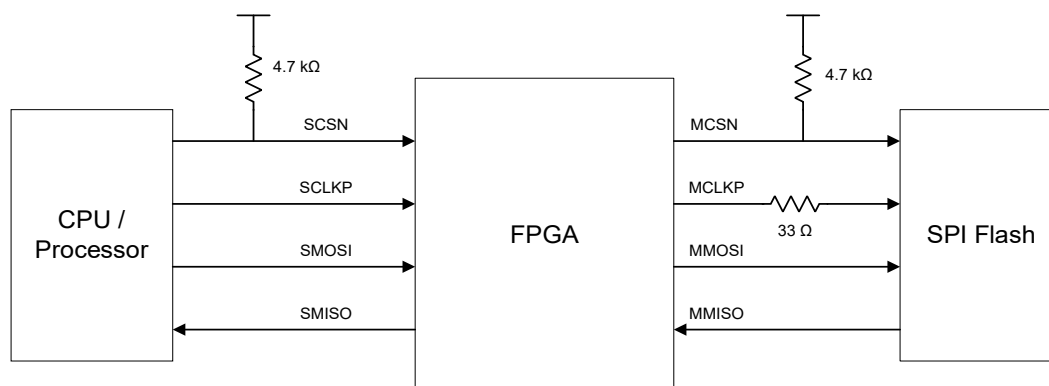


Figure 6.10. Target SPI to Controller SPI Bridge Block Diagram

## 6.4. JTAG Mode

The Nexus 2 device provides a dedicated four-pin JTAG interface, which is fully compliant with IEEE 1149.1 (IEEE Standard for Test Access Port and Boundary-Scan Architecture). The JTAG port on Nexus 2 devices provides:

- External flash memory programming in normal or transparent mode (background programming)
- Direct SRAM configuration
- Full access to the Nexus 2 device configuration logic
- Device chaining
- IEEE 1149.1/1149.6 testability

The advantages of the JTAG port include:

- Multi-chain architectures – The JTAG port is the only configuration and programming port that permits the Nexus 2 device to be combined in a chain of other programmable logic.  
**Note:** Refer to [Appendix F](#) for more information on JTAG daisy-chain configuration with an external host controller.
- Reveal Debug – The Lattice Reveal debug tool is an embeddable logic analyzer tool that allows analysis of the user design implemented in a Nexus 2 device, similar to how an external logic analyzer permits analysis of board-level logic. Reveal Debug access is only available through the JTAG port.
- Configuration memory readback – The JTAG port can access the configuration memory. It is occasionally necessary to perform failure analysis. A key component for failure analysis is reading the configuration memory.
- Boundary scan testability – Board-level connectivity testing performed using IEEE 1149.1 and IEEE 1149.6 JTAG is a key capability for assuring the quality of assembled printed circuit boards. Lattice provides Boundary Scan Description Language (BSDL) files for the Nexus 2 device on the Lattice website.

### 6.4.1. JTAG Port AC Timing Requirements

The JTAG port AC timing requirements are listed in [Table 6.4](#).

**Table 6.4. JTAG AC Timing Requirements**

| Description   | Parameter | Min | Max | Unit |
|---------------|-----------|-----|-----|------|
| TCK Frequency | $f_{MAX}$ | —   | 25  | MHz  |

For detailed AC timing requirements for the Nexus 2 JTAG port, refer to the JTAG Port Timing Specifications of the Lattice Nexus 2 Platform – Specifications Data Sheet (FPGA-DS-02121).

### 6.4.2. JTAG to Controller SPI Bridge

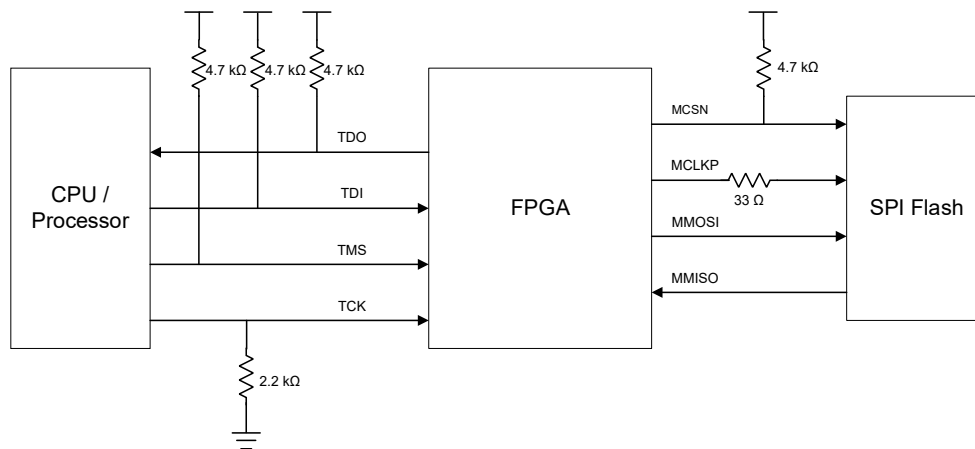
Lattice Nexus 2 devices provide the JTAG to controller SPI bridge, which enables programming the external SPI flash through the Nexus 2 device JTAG port.

**Notes:**

- In user function mode, to program the external SPI flash through the JTAG port, controller SPI port persistence must be enabled.
- The JTAG to controller SPI bridge is accessible regardless of whether CFGMODE is high or low.

The PROG\_SPI instruction is used to enable this bridging. There is a 16-bit test data register (TDR) between TDI and TDO for this instruction, which must be loaded with a key value of 16'h3A3A. Upon the TAP controller going through the Update-DR state the first time with the proper key loaded, the internal TDI, gated TCK, Shift-DR, and internal TDO signals are connected to the MOSI/MD0, MCLKP, MCSN, and MISO/MD1 pins, respectively.

Figure 6.11 shows a block diagram example of the JTAG to controller SPI bridge use.



**Figure 6.11. JTAG to Controller SPI Bridge Block Diagram**

### 6.4.3. JTAG Reveal Support

Nexus 2 devices support the addition of internal logic analyzers (ILAs). These ILAs provide features similar to external logic analyzers such as programmable events, trigger conditions, and deep trace memory. Internal logic analyzer functionality in Nexus 2 devices can be implemented with the Reveal software. For more information on the Reveal software, refer to the Reveal User Guide for Radiant Software.

## 6.4.4. JTAG Operating Modes

The JTAG port supports two operating modes. The first is direct mode where instructions are executed locally within the JTAG interface. The second is configuration command mode that interacts with the Nexus 2 device configuration controller using two specific JTAG instructions. Configuration command mode is not available when the device is loading configuration data through the controller SPI port or if the target SPI port has requested exclusive access.

### 6.4.4.1. JTAG Direct Mode

Table 6.5 lists the JTAG direct mode instructions.

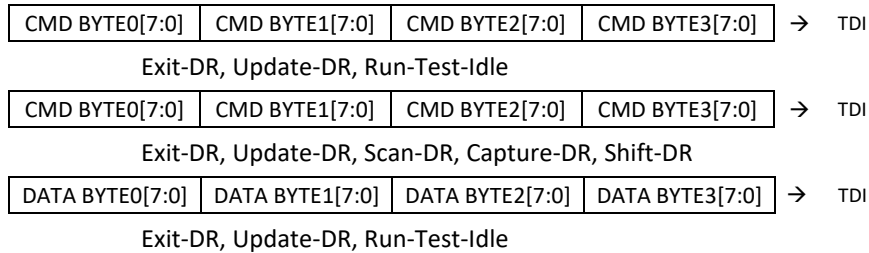
**Table 6.5. JTAG Direct Mode Instructions**

| Instruction              | Opcode | Data Register Length (Bits) | Supported in Test Only Mode | Description                                                                                                                                                                                    |
|--------------------------|--------|-----------------------------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EXTEST                   | 0x15   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| CLAMP                    | 0x78   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| HIGHZ                    | 0x18   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| SAMPLE/PRELOAD           | 0x1C   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| EXTEST_PULSE             | 0x2D   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| EXTEST_TRAIN             | 0x2E   | —                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| BYPASS                   | 0xFF   | 1                           | Y                           | IEEE 1149.1 boundary scan                                                                                                                                                                      |
| IDCODE_PUB               | 0xE0   | 32                          | Y                           | Read out the public IDCODE of the device. If the CUST_IDCODE_EN bit is set in OTP, the CUSTOMER_IDCODE stored in OTP is returned. If CUST_IDCODE_EN is clear, the hardware IDCODE is returned. |
| IDCODE_PRV               | 0x16   | 32                          | Y                           | Read out the 32-bit private (hardware) IDCODE of the device.                                                                                                                                   |
| UIDCODE                  | 0x19   | 64                          | Y                           | Read out the TraceID.                                                                                                                                                                          |
| USERCODE                 | 0xC0   | 32                          | Y                           | Read out the User Electronic Signature.                                                                                                                                                        |
| READ_STATUS              | 0x3C   | 64                          | Y                           | Read out STATUS1:STATUS0. To read STATUS2, use target configuration commands as shown in Table 6.12.                                                                                           |
| CHECK_BUSY               | 0xF0   | 32                          | Y                           | Read out the busy flag.                                                                                                                                                                        |
| PORT_REQUEST             | 0x7A   | 8                           | Y                           | Refer to OP3 in the <a href="#">PORT_REQUEST</a> section for data register details.                                                                                                            |
| PORT_STATUS_READ         | 0x7B   | 32                          | Y                           | Refer to the return data in the <a href="#">PORT_STATUS_READ</a> section for data register details.                                                                                            |
| BSCAN_CFG                | 0x2F   | 8                           | Y                           | 1149.6 boundary scan setup                                                                                                                                                                     |
| CONFIGURATION_DATA_SHIFT | 0xF1   | 32                          | N                           | Shift configuration command/data in and out.                                                                                                                                                   |
| CONFIGURATION_BURST      | 0xF2   | 1                           | N                           | Stream configuration commands/data in.                                                                                                                                                         |
| PROG_SPI                 | 0x3A   | 16                          | Y                           | Refer to the <a href="#">JTAG to Controller SPI Bridge</a> section.                                                                                                                            |
| REFRESH                  | 0x79   | 1                           | Y                           | The REFRESH instruction is equivalent to toggling the PROGRAMN pin.                                                                                                                            |

#### 6.4.4.2. JTAG Configuration Command Mode

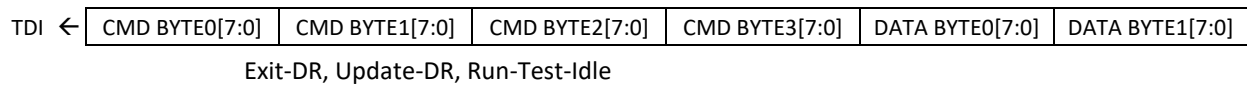
The CONFIGURATION\_DATA\_SHIFT and CONFIGURATION\_BURST instructions are used to access configuration command mode. Target configuration commands are described in the [Target Configuration Command Set](#) section.

CONFIGURATION\_DATA\_SHIFT uses a 32-bit shift register during the Shift-DR TAP state to receive commands and data, and capture returned data. Following IEEE 1149.1, the data register is shifted LSB first. If more than 32 bits are needed, the Shift-DR TAP state can be re-entered as many times as needed. Entering the Run-Test-Idle TAP state terminates the transaction. For configuration commands that provide read back data, refer to the [JTAG Read](#) section.



**Figure 6.12. JTAG CONFIGURATION\_DATA\_SHIFT**

CONFIGURATION\_BURST allows for a continuous stream of configuration commands and data. Once in the Shift-DR TAP state, data is streamed from the TDI pin MSB first. Read back is not supported during CONFIGURATION\_BURST. Entering the Update-DR TAP state terminates the transaction.



**Figure 6.13. JTAG CONFIGURATION\_BURST**

## 6.5. Dual-Boot and Multi-Boot

Dual-boot and multi-boot are applicable to both the controller SPI mode and self-download mode.

Both the primary and secondary configuration images are stored in an external SPI storage device or internal flash memory. The device loads the secondary image (golden image) if it fails to load the primary image. The primary image can fail for one of the following reasons:

- A time-out error is encountered while waiting for a valid bitstream.
- Device ID checking fails at the beginning of the bitstream.
- An illegal command is encountered.
- A bitstream CRC error is detected.
- The bitstream fails security checks, for example, bitstream authentication.
- The bitstream revision is less than the minimum bitstream revision allowed by the device.

A CRC error is caused by incorrect data being read from non-volatile memory. Data is read from the external SPI storage device or internal flash memory. As data enters the configuration engine, the data is checked for CRC consistency. Before the data enters the configuration SRAM, the CRC must be correct. Any incorrect CRC causes the device to erase the configuration SRAM and retrieve configuration data from the golden image.

The dual-boot configuration mode requires two configuration images. One of the two configuration images is a fail-safe image that is rarely, if ever, updated. The other configuration image is a working image that is routinely updated. Both the working (primary) image and the fail-safe image are stored in the external SPI storage device or internal flash memory. One Lattice Radiant project can be used to create both the working and the fail-safe configuration images. Configure the Lattice Radiant project with an implementation named *working*, and an implementation named *failsafe*. Read the Lattice Radiant Online Help for more information about using Lattice Radiant implementations.

The Nexus 2 device supports dual-boot with the controller SPI/self-download mode. If the primary image fails to load correctly, the device starts loading data from the golden sector in the non-volatile memory. A blank external or internal flash memory causes a dual-boot event failure indicated by INITN going low. This is due to the absence of a primary or golden boot image.

The dual-boot feature allows a non-volatile memory storage device to be split into two sections, the first containing a *golden boot* file, and the second containing an updatable *primary boot* file, which can be erased and reprogrammed. By default, the FPGA loads the primary boot file from boot address 0x00010000. If the FPGA fails to configure, it automatically loads the golden boot file as addressed through the Program Secondary Boot Address command or the backup jump table. This allows the system to boot to a known operable state so that it continues to operate if for some reason (such as a power failure) the non-volatile memory fails to be programmed correctly. For more information regarding dual-boot mode, refer to [Lattice Nexus 2 Multi-Boot User Guide \(FPGA-TN-02385\)](#).

Multi-boot, or the ability to dynamically reconfigure from multiple design bitstreams, is similar to the dual-boot, where there is one primary (working) bitstream and one golden bitstream, but also supports up to 16 alternative bitstreams.

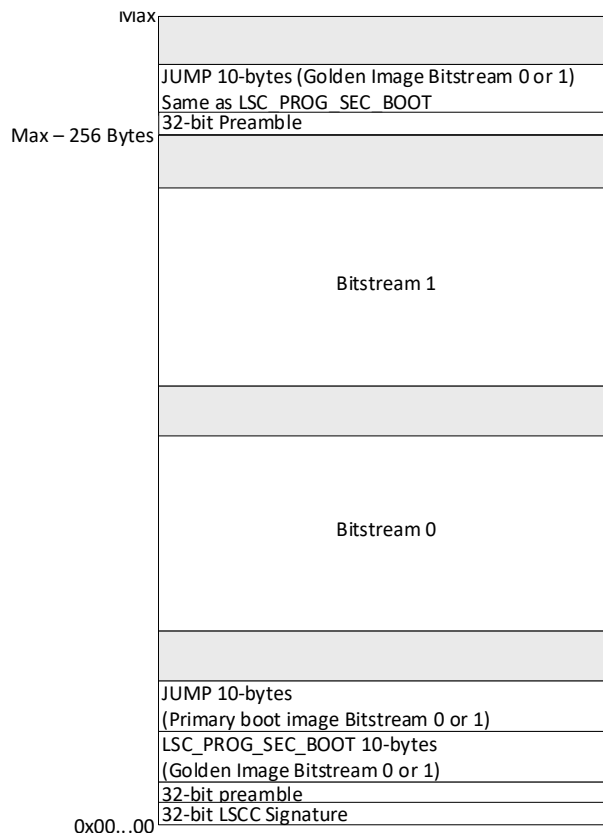
For multi-boot operation, the device boots up from the primary image. If the primary image fails to load, the device attempts to boot from the golden image. After successful configuration with the primary image, the MULTI\_BOOT\_SEL attribute determines the subsequent loading of alternative images from toggling the PROGRAMN pin or issuing a REFRESH command. For more information regarding multi-boot mode, refer to [Lattice Nexus 2 Multi-Boot User Guide \(FPGA-TN-02385\)](#).

The Lattice Radiant Deployment Tool can assemble flash images formatted to correctly match the hardware sector mapping.

## 6.6. Ping-Pong Boot

Ping-pong boot is applicable to both the controller SPI mode and self-download mode.

The ping-pong boot mode uses a jump table to select an image for booting without changing the location of the image in flash. This is done with a jump table starting at address 0x000000 (can be changed through OTP setting) in flash containing two instructions PROG\_SEC\_BOOT and JUMP. For backup, in case the jump table becomes corrupted, a backup JUMP instruction should be programmed at the last 256 bytes of the flash pointing to the golden boot image. Refer to [Figure 6.14](#) for the two images stored in flash Bitstream 0 and Bitstream 1. The secondary bitstream offset is programmed both with the command PROG\_SEC\_BOOT located in the jump table starting at offset 0x000000 and the JUMP instruction located in the backup jump table at the last 256 bytes of the flash, and the primary bitstream is selected with the JUMP command in the jump table. To swap the order of booting, re-program these three instructions. For more information regarding ping-pong boot mode, refer to [Lattice Nexus 2 Multi-Boot User Guide \(FPGA-TN-02385\)](#).



**Figure 6.14. Jump Table**

## 6.7. Status Registers

Status registers provide information about the device during and after the configuration process. The registers can be accessed using the READ\_STATUS0, READ\_STATUS1, and READ\_STATUS2 commands. The default value of the status registers after powering up, PROGRAMN pin pulsing, or REFRESH command execution is 0x00000000.

### 6.7.1. Status Register 0 (STATUS0)

**Table 6.6. Status Register 0**

| Field   | Name                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:29] | HW Command Mode Error Code    | 000 – No error has occurred<br>001 – Command error 0: Transaction ended before next command received<br>010 – Command error 1: Transaction did not end before next command received<br>011 – Invalid command error<br>100 – Blocked command received<br>101 – Reserved<br>...<br>111 – Reserved                                                                                                                                                                                                                                                                                                                                                                                                                      |
| [28:25] | BSE Secondary Boot Error Code | 0000 – No error has occurred.<br>0001 – There was a mismatch to the device ID code<br>0010 – There was an illegal command detected<br>0011 – There was a Frame CRC checksum error<br>0100 – No preamble before preamble timeout in controller SPI<br>0101 – Bitstream Engine execution was aborted by the user<br>0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed)<br>0111 – There was a CRC checksum error<br>1000 – Authentication Error<br>1001 – Authentication Setup Error<br>1010 – Bitstream Timeout Error<br>1011 – RX FIFO Overflow Error<br>1100 – Global CRC Checksum Error<br>1101 – Command and SFDP Conflict<br>1110 – Security Error<br>1111 – Misc Error |
| [24:21] | BSE Primary Boot Error Code   | 0000 – No error has occurred.<br>0001 – There was a mismatch to the device ID code<br>0010 – There was an illegal command detected<br>0011 – There was a Frame CRC checksum error<br>0100 – No preamble before preamble timeout in controller SPI<br>0101 – Bitstream Engine execution was aborted by the user<br>0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed)<br>0111 – There was a CRC checksum error<br>1000 – Authentication Error<br>1001 – Authentication Setup Error<br>1010 – Bitstream Timeout Error<br>1011 – RX FIFO Overflow Error<br>1100 – Global CRC Checksum Error<br>1101 – Command and SFDP Conflict<br>1110 – Security Error<br>1111 – Misc Error |

| Field   | Name                       | Description                                                                                                                                                                                                                                                                                                                              |
|---------|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [20:19] | MSPI Primary Boot Failed   | Non-zero status shows that the primary booting has failed, even if the secondary booting is successful when dual boot is enabled. Once non-zero, it can only be cleared by REFRESH command execution, or PROGRAMN pin toggle low.<br>00 – None<br>01 – Failed due to error in bitstream<br>10 – Failed due to unstable V <sub>CCIO</sub> |
| [18:17] | Preamble                   | Indicates what preamble was received through the last bitstream segment.<br>00 – Plain<br>01 – Auth Header<br>10 – Authenticated<br>11 – Encrypted + Authenticated                                                                                                                                                                       |
| [16]    | Reserved                   | Reserved                                                                                                                                                                                                                                                                                                                                 |
| [15]    | CID_EN                     | Status bit indicates that the customer ID (CID) EN bit inside the OTP is programmed. The IDCODE_PUB command will read out the User ID, which is the 32-bit user IDCODE inside the OTP feature row. The IDCODE_PRV command will still read out the hardware device ID code when this bit is set.                                          |
| [14]    | Decrypt Only               | Decrypt only is set in OTP, and only encrypted bitstreams will be accepted. See command table for what commands are accepted outside an encrypted bitstream when this bit is set.                                                                                                                                                        |
| [13]    | Reserved                   | Reserved                                                                                                                                                                                                                                                                                                                                 |
| [12]    | Fail Flag                  | Indicates previous HW command failed. Valid in command mode for commands.                                                                                                                                                                                                                                                                |
| [11]    | Busy Flag                  | Config engine is busy.                                                                                                                                                                                                                                                                                                                   |
| [10]    | INITN Pin State            | INITN pin state.<br>0 – Low<br>1 – High                                                                                                                                                                                                                                                                                                  |
| [9]     | Auth DONE                  | Authentication is done.<br>1 – Authentication done                                                                                                                                                                                                                                                                                       |
| [8]     | DONE Pin State             | Indicates the current state of the DONE pin. Device wakeup is complete when the DONE pin is high.<br>0 – Low<br>1 – High                                                                                                                                                                                                                 |
| [7]     | DONE                       | Device prog_done bit is set from PROG_DONE command. Indicates whether the PROG_DONE command was received through the bitstream. Does not indicate that configuration is complete.                                                                                                                                                        |
| [6]     | Read Enable                | Configuration memory read is enabled, qualified with security state.                                                                                                                                                                                                                                                                     |
| [5]     | Write Enable               | Configuration memory write is enabled, qualified with security state.                                                                                                                                                                                                                                                                    |
| [4]     | Erase Enable               | Configuration memory erase is enabled, qualified with security state.                                                                                                                                                                                                                                                                    |
| [3:1]   | FW Command Mode Error Code | 000 – No error has occurred<br>001 – Reserved<br>010 – Reserved<br>011 – Invalid Command error<br>100 – Blocked Command received<br>101 – ID Error                                                                                                                                                                                       |
| [0]     | Transparent Mode Enabled   | Device is in transparent programming mode.                                                                                                                                                                                                                                                                                               |

**Note:**

1. Bitstream Engine Error Status fields are disabled when bitstream authentication is enabled.

## 6.7.2. Status Register 1 (STATUS1)

**Table 6.7. Status Register 1**

| Field   | Name                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31]    | FEA Read Lock                 | OTP feature row is locked for read                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [30]    | FEA Write Lock                | OTP feature row is locked for write                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| [29]    | AES Write Lock                | OTP AES Key is locked for Write                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [28]    | PUB Key Write Lock            | OTP PUB Key is locked for Write                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [27]    | Reserved                      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [26]    | SSPI -> MSPI Bridge Lock      | Target SPI to controller SPI bridge is locked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [25]    | JTAG -> MSPI Bridge Lock      | JTAG to controller SPI bridge is locked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [24]    | LMMI -> MSPI Bridge Lock      | LMMI to controller SPI bridge is locked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [23:20] | BS Auth Mode                  | Bitstream authentication mode.<br>0x0 – None<br>0x1 – AES256-GMAC<br>0x2 – Reserved<br>0x3 – Reserved<br>0x4 – ECDSA256 (digest = SHA2-256 or AES256-GCM1)<br>0x5 – ECDSA384 (digest = SHA2-384 or AES256-GCM1)<br>0x6 – ECDSA521 (digest = SHA2-512 or AES256-GCM1)<br>0x7 – Reserved<br>...<br>0x9 – Reserved<br>0xA – RSA2048 (digest = SHA2-256 or AES256-GCM1)<br>0xB – RSA3072 (digest = SHA2-384 or AES256-GCM1)<br>0xC – RSA4096 (digest = SHA2-512 or AES256-GCM1)<br>0xD – Reserved<br>...<br>0xF – Reserved |
| [19:18] | Device State                  | 00 – Security engine off<br>01 – Security engine on<br>10 – Reserved<br>11 – Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| [17]    | ATM Global Alarm              | There was a system ATM alarm trigger event.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| [16]    | PUF_Enrolled                  | 0 – PUF not enrolled<br>1 – PUF enrolled from OTP                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| [15]    | se_haltstate Asserted         | se_haltstate of the security engine has asserted at least once since last refresh.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| [14]    | JTAG sysCONFIG Lock           | JTAG sysCONFIG commands are disabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| [13]    | JTAG ispTracy Lock            | JTAG ispTracy instructions are disabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [12]    | JTAG Boundary Scan Lock       | JTAG boundary scan instructions are disabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| [11]    | User WDT Reboot               | Device was rebooted because the user watchdog timed out.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 10      | User WDT Busy                 | User watchdog timer is running                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [9]     | SFDP Timeout                  | Boot timed out reading the flash signature. SFDP/LSCC                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| [8]     | SSPI Timeout                  | The target SPI timed out before receiving the done command, while sending a bitstream.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| [7]     | Daisy Chain Flow Through Mode | Device has completed configuration and is in daisy-chain flow-through state.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| [6]     | Daisy Chain Bypass Mode       | Device has completed configuration and is in daisy-chain bypass state.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Field | Name                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [5:2] | Dry Run BSE Error Code | 0000 – No error has occurred<br>0001 – There was a mismatch to the device ID code<br>0010 – There was an illegal command detected<br>0011 – There was a Frame CRC checksum error<br>0100 – No preamble before preamble timeout in controller SPI<br>0101 – Bitstream Engine execution was aborted by the user<br>0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed)<br>0111 – There was a CRC checksum error<br>1000 – Authentication Error<br>1001 – Authentication Setup Error<br>1010 – Bitstream Engine Timeout Error<br>1011 – RX FIFO Overflow Error<br>1100 – Global CRC checksum Error<br>1101 – Command and SFDP Conflict<br>1110 – Security Error<br>1111 – Misc Error |
| [1]   | Dry Run Auth Done      | The dry-run bitstream has completed authentication.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| [0]   | Dry Run Done           | The dry-run bitstream done command was received and no errors.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

### 6.7.3. Status Register 2 (STATUS2)

**Table 6.8. Status Register 2**

| Field   | Name                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:25] | Reserved                    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [24:23] | OTP Keys Erased             | OTP keys all programmed to 1s<br>00 – Normal OTP<br>01 – Key erase in progress<br>10 – Key erase in progress<br>11 – Keys erased                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| [22:21] | BBRAM Status                | Bit 0: Other Data Valid<br>1 – If the following are all true: <ul style="list-style-type: none"> <li>• BBRAM initialized</li> <li>• Any of the following rows are non-zero: <ul style="list-style-type: none"> <li>• Min bitstream version</li> <li>• User Data 0-7</li> <li>• Monotonic counter 0-1</li> <li>• Unused BBRAM words</li> </ul> </li> </ul> Bit 1: BS Keys Valid<br>1 – If the following are all true: <ul style="list-style-type: none"> <li>• BBRAM initialized.</li> <li>• Bitstream AES Key is valid.</li> <li>• Bitstream Authentication Public Key hash is valid.</li> </ul> |
| [20]    | Fabric Lockdown Zeroization | 0 – Fabric has not been zeroized due to lockdown<br>1 – Fabric has been zeroized (erased and verified) due to lockdown                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| [19]    | Reserved                    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [18]    | Key Erase Triggered         | Key erase triggered from erase_key_pin asserted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| [17]    | BBRAM Enabled               | BBRAM is enabled through OTP feature setting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| [16:13] | PUF Diagnostic Score        | 4-bit value indicates PUF diagnostic score.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| [12:2]  | Reserved                    | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [1]     | FIPS Enabled                | 0 – FIPS mode disabled<br>1 – FIPS mode enabled. From OTP feature setting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| [0]     | BBRAM Valid                 | BBRAM has gone through initialization.<br>0 – Contents invalid<br>1 – Contents valid                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

## 6.8. Control Registers

Control Register 0 (CR0) and Control Register 1 (CR1) are used to control configuration logic behavior during and after configuration. CR0 can be written using the PROG\_CNTRL0 command in the bitstream and read using the READ\_CNTRL0 command. The default value of CR1 is loaded from OTP during boot, before the bitstream is loaded. CR1 can be written using the PROG\_CNTRL1 command in bitstream and read using the READ\_CNTRL1 command.

### 6.8.1. Control Register 0 (CR0)

**Table 6.9. Control Register 0**

| Field   | Name                   | Default | Description                                                                                                                                                                                                                                        |
|---------|------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:29] | Reserved               | 0       | Reserved                                                                                                                                                                                                                                           |
| [28]    | TRANS_FR               | 0       | Enable TransFR to latch and freeze I/O during reconfiguration                                                                                                                                                                                      |
| [27:26] | Reserved               | 0       | Reserved                                                                                                                                                                                                                                           |
| [25]    | TRAN_CRAM <sup>1</sup> | 0       | Enable transparent CRAM programming                                                                                                                                                                                                                |
| [24]    | TRAN_INIT <sup>1</sup> | 0       | Enable transparent INIT bus programming                                                                                                                                                                                                            |
| [23]    | TRAN_SEC_ENGINE        | 0       | Security engine mode<br>0 – Bitstream Security<br>1 – User Security                                                                                                                                                                                |
| [22]    | PRESERVE_REG           | 0       | Preserve CR1 register values. When set, CR1 will not revert to its default value after REFRESH command execution or PROGRAMN pin pulsing.                                                                                                          |
| [21]    | NO BOOT                | 0       | Do not boot on REFRESH command execution or PROGRAMN pin pulsing                                                                                                                                                                                   |
| [20]    | Reserved               | 0       | Reserved                                                                                                                                                                                                                                           |
| [19]    | MULTIBOOT_EN           | 0       | Enable multi-boot                                                                                                                                                                                                                                  |
| [18:17] | DAISYCHAIN_CTRL        | 0       | Enable daisy-chain mode.<br>00 – None<br>01 – None<br>10 – Bypass Mode<br>11 – Flow-Through Mode                                                                                                                                                   |
| [16:15] | INTN OPT               | 0       | Override INITN. If CR0[16] is set to 1, the DONE pin is overridden by CR0[15].                                                                                                                                                                     |
| [14:13] | DONE OPT               | 0       | Override DONE. If CR0[14] is set to 1, the DONE pin is overridden by CR0[13].                                                                                                                                                                      |
| [12:10] | Reserved               | 0       | Reserved                                                                                                                                                                                                                                           |
| [9]     | WAIT_DONE              | 0       | Wait for DONE pin before wake-up. In daisy chain mode, wake-up is held off until all external devices have released the DONE pin.                                                                                                                  |
| [8]     | Fast Slew Rate Enable  | 0       | Force fast slew rate when controller SPI clock frequency is below 25 MHz. The default sysCONFIG I/O slew rate is slow when the controller SPI clock frequency is below 25 MHz and fast when the controller SPI clock frequency is 25 MHz or above. |
| [7:0]   | Reserved               | 0       | Reserved                                                                                                                                                                                                                                           |

**Note:**

1. When bitstream authentication is enabled, TRAN\_CRAM and TRAN\_INIT are disabled.

## 6.8.2. Control Register 1 (CR1)

**Table 6.10. Control Register 1**

| Field   | Name                       | Default | Description                                                                                                                                                                                                             |
|---------|----------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [31:30] | Reserved                   | 0       | Reserved                                                                                                                                                                                                                |
| [29:27] | MSPI Signature Timer Count | 0       | Controller SPI signature timeout value.<br>0 – 200 ms<br>1 – 100 ms<br>2 – 50 ms<br>3 – 40 ms<br>4 – 20 ms<br>5 – 1 ms<br>6 – 500 $\mu$ s<br>7 – 100 $\mu$ s                                                            |
| [26]    | MSPI_CPHA                  | 0       | Controller SPI CPHA<br>0 – Data sampled at the leading (first) edge of the clock<br>1 – Data sampled at the trailing (second) edge of the clock                                                                         |
| [25]    | MSPI_CPOL                  | 0       | Controller SPI CPOL<br>0 – Active high clock. Clock is low in idle state.<br>1 – Active low clock. Clock is high in idle state.                                                                                         |
| [24]    | MSPI_TX_Edge               | 0       | Invert controller SPI transmit data edge                                                                                                                                                                                |
| [23]    | MSPI_RX_Edge               | 0       | Invert controller SPI receive data edge                                                                                                                                                                                 |
| [22]    | Reserved                   | 0       | Reserved                                                                                                                                                                                                                |
| [21]    | SSPI_CPHA                  | 0       | Target SPI CPHA<br>0 – Data sampled at the leading (first) edge of the clock<br>1 – Data sampled at the trailing (second) edge of the clock                                                                             |
| [20]    | SSPI_CPOL                  | 0       | Target SPI CPOL<br>0 – Active high clock. Clock is low in idle state.<br>1 – Active low clock. Clock is high in idle state.                                                                                             |
| [19]    | SSPI_TX_Edge               | 0       | Invert target SPI transmit data edge                                                                                                                                                                                    |
| [18]    | SSPI_RX_Edge               | 0       | Invert target SPI receive data edge                                                                                                                                                                                     |
| [17]    | SSPI_LSBF                  | 0       | Change target SPI bit shift order to LSB first                                                                                                                                                                          |
| [16:14] | SSPI_AUTO                  | 0       | Enable device as a downstream device in a daisy chain<br>000 – SPI auto/SCM mode disabled<br>001 – SCM mode<br>010 – SPI auto x1 mode<br>011 – SPI auto x2 mode<br>100 – SPI auto x4 mode<br>101 – SPI auto DDR x8 mode |
| [13]    | EBR Erase Disable          | 0       | Disable the erase of EBR contents on REFRESH command execution or PROGRAMN pin pulsing.                                                                                                                                 |
| [12]    | SFDP_Enable                | 0       | Enable SFDP signature check for flash devices supporting SFDP                                                                                                                                                           |
| [11]    | Signature_Disable          | 0       | Disable Flash Signature check at power-up                                                                                                                                                                               |
| [10]    | Signature Infinite Retry   | 0       | Do not timeout on signature check                                                                                                                                                                                       |
| [9]     | 32-bit MSPI Address        | 0       | Enable 32-bit controller SPI address (default: 24-bit)                                                                                                                                                                  |
| [8]     | 32-bit MSPI Commands       | 0       | Enable 32-bit controller SPI commands (default: 24-bit)                                                                                                                                                                 |
| [7]     | Disable IO Glitch          | 0       | Disable the I/O glitch filter during configuration                                                                                                                                                                      |

| Field | Name                       | Default | Description                                                                                                                                                                                                                      |
|-------|----------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [6:3] | SSPI/JTAG Idle Timer Count | 0       | Target SPI idle timeout value<br>0 – Disabled (default) – bitstream sent in one continuous stream<br>1 – 200 s<br>2 – 100 s<br>3 – 50 s<br>4 – 25 s<br>5 – 10 s<br>6 – 5 s<br>7 – 1 s<br>8 – 750 ms<br>9 – 500 ms<br>10 – 250 ms |
| [2:0] | MSPI Preamble Timer Count  | 0       | Controller SPI preamble timeout value<br>0 – 200 ms<br>1 – 100 ms<br>2 – 50 ms<br>3 – 40 ms<br>4 – 20 ms<br>5 – 1 ms<br>6 – 500 $\mu$ s<br>7 – 100 $\mu$ s                                                                       |

## 6.9. User Mode Register

The user mode register bits control the behavior of the device after entering user function mode. These bits are programmed by the bitstream.

**Table 6.11. User Mode Register**

| Field     | Name                      | Default | Module  | Description                                                                                               |
|-----------|---------------------------|---------|---------|-----------------------------------------------------------------------------------------------------------|
| [127:122] | reserved                  | 0       | —       | Reserved                                                                                                  |
| [121]     | user_security_en          | 0       | —       | Enable security engine fabric interfaces and user functions                                               |
| [120]     | tran_init_en              | 0       | Control | Enable user control of transparent INIT Bus Programming through CR0.tran_init                             |
| [119]     | tran_sec_engine_en        | 0       | Control | Enable user control of transparent security engine access through CR0.tran_sec_engine                     |
| [118]     | en_tsall                  | 0       | I/O     | Enable CIB control to drive TS_ALL after configuration                                                    |
| [117]     | tsall_inv                 | 0       | I/O     | Enable CIB TS_ALL control to be active low                                                                |
| [116]     | en_user_gsrn              | 0       | CLKRST  | When set, enables the user gsrn from fabric.                                                              |
| [115]     | user_gsr_inv              | 0       | CLKRST  | When set, inverts the polarity of user gsrn from fabric.                                                  |
| [114]     | user_gsr_clk_pos          | 0       | CLKRST  | When set, inverts the fabric user gsrn clock                                                              |
| [113]     | user_gsr_sync             | 0       | CLKRST  | 0 – gsrn is asynchronous<br>1 – User gsrn is synchronous                                                  |
| [112]     | mspi_addr_source_sel      | 0       | BSE     | Controller SPI address selection for multiple boot<br>0 – mc_mspi_addr[47:0]<br>1 – lmmi_mspim_addr[47:0] |
| [111:64]  | mspi_multiboot_addr[47:0] | 0       | BSE     | SPI boot flash sector mc1 address, address sent for multi-boot when mspi_addr_source_sel=0                |
| [63:32]   | wdt_value[31:0]           | 0       | USERWDT | User watchdog timer count value                                                                           |
| [31]      | wdt_en                    | 0       | USERWDT | Use to enable the user watchdog timer                                                                     |
| [30]      | wdt_mode                  | 0       | USERWDT | Select watchdog timer mode<br>0 – Single shot<br>1 – Continuous                                           |
| [29:25]   | reserved                  | 0       | —       | Reserved                                                                                                  |
| [24]      | sedc_enable               | 0       | SEDC    | Enable SEDC function                                                                                      |
| [23:16]   | sedc_clock_div[7:0]       | 0       | SEDC    | Set the SEDC divider value.<br>$sedc\_clk = clk\_osc / (sedc\_clock\_div + 1)$                            |
| [15]      | Reserved                  | 0       | —       | Reserved                                                                                                  |
| [14]      | sspi_en                   | 0       | SSPI    | Enable the target SPI port in user function mode                                                          |
| [13]      | lmmi_en                   | 0       | LMMI    | Enable LMMI port in user function mode                                                                    |
| [12]      | er1_exist                 | 0       | JTAG    | Enable ispTracy TDO1                                                                                      |
| [11]      | er2_exist                 | 0       | JTAG    | Enable ispTracy TDO2                                                                                      |
| [10]      | glb_gsr_n_dis             | 0       | CLKRST  | Disable global gsrpi                                                                                      |
| [9]       | lmmi_gsrn_dis             | 0       | CLKRST  | Disable global gsr for lmmi                                                                               |
| [8]       | wdt_gsrn_dis              | 0       | CLKRST  | Disable global gsr for wdt                                                                                |
| [7]       | sedc_gsrn_dis             | 0       | CLKRST  | Disable global gsr for sedc                                                                               |
| [6]       | jtag_gsrn_dis             | 0       | CLKRST  | Disable global gsr for jtag                                                                               |
| [5]       | lmmi_clk_inv              | 0       | CLKRST  | Invert the LMMI clock                                                                                     |
| [4]       | sec_gsrn_dis              | 0       | CLKRST  | Disable global gsr for security engine                                                                    |
| [3:0]     | reserved                  | 0       | —       | Reserved                                                                                                  |

## 6.10. Configuration Commands

A configuration command is an instruction issued through the bitstream to the configuration logic to perform an operation during device configuration.

### 6.10.1. Command Format

All commands except no operation (NOOP) consist of four mandatory bytes plus a variable number of optional data bytes that are defined on a per-command basis. NOOP is a single-byte command (0x00, 0x7F, or 0xFF). The configuration command format is shown in Figure 6.15.

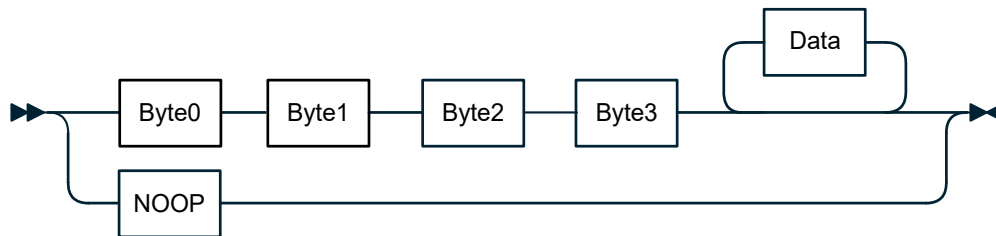


Figure 6.15. Configuration Command Format

### 6.10.2. Read Commands

Some commands read information from the device and output the result data back through the interface that sent the command. The number of bytes of result data is determined by the command that was sent. Table 6.12 defines the size of the result data for each command.

Latency for returning result data from a read command may be variable. Each target interface (target SPI, JTAG) implements an interface-specific protocol to indicate when the result data is present and valid.

#### 6.10.2.1. JTAG Read

- Read Command Sequence
  - Host loads CONFIGURATION\_DATA\_SHIFT into the instruction shift register.
  - Host sends a 4-byte read command through the data shift register.
  - The data shift register captures 0xFFFFFFFF until return data is ready.
  - Host shifts in NOOPs and observes captured data.
  - Once return data is ready, the shift register captures 0xFFFFF00 and then captures the return data.
  - When return data is complete, the shift register captures 0xFFFFFFFF.
- Usage
  - Host monitors captured data until it sees 0xFFFFF00, then samples return data.

#### 6.10.2.2. Target SPI Read

- Read Command Sequence
  - Controller sends a 4-byte read command over MOSI (SPI) or SIO0-7 (DSPI/QSPI/xSPI).
  - Controller sends dummy bits on MOSI (SPI) or tri-states SIO0-7 (DSPI/QSPI/xSPI) while sampling return data.
  - Immediately after the command is received, target sends 1 on MISO (SPI) or tri-states SIO0-7 (DSPI/QSPI/xSPI) until return data is ready (minimum of 1 byte of 0xFF).
    - For DSPI/QSPI/xSPI, controller receives 0xFF because I/O pins are pulled up internally (weak pull-up).
  - When return data is ready, target sends a data start byte (0x00) followed immediately by the return data.
  - Target SPI tri-states I/O pins when return data FIFO is empty or SCSN is de-asserted.
- Usage
  - Controller monitors return data until it sees 0xFF -> 0x00 transition, then samples the return data.

### 6.10.3. Target Configuration Command Set

The supported commands for all target configuration interfaces such as target SPI, JTAG, and LMMI are shown in Table 6.12.

**Table 6.12. Target Configuration Commands**

| Command Name       | Byte0 | Byte1       | Byte2          | Byte3         | # DATA BYTES | # RETURN BYTES          | PROG_QUALIFY <sup>1</sup> | Type <sup>2</sup> | Description                                                                                                                                                                       |
|--------------------|-------|-------------|----------------|---------------|--------------|-------------------------|---------------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PROG_ENABLE        | 0x80  | 0x00        | 0x0c           | 0x00–0x02     | —            | —                       | —                         | IMM               | Enable programming based on byte3 value:<br>0x00 – CRAM/INIT/OTP<br>0x01 – CRAM/INIT only<br>0x02 – OTP only                                                                      |
| PROG_DISABLE       | 0x80  | 0x00        | 0x0e           | 0x00          | —            | —                       | —                         | IMM               | Disable programming of CRAM/INIT/OTP.                                                                                                                                             |
| BITSTREAM_BURST    | 0x05  | 0x00        | 0x00           | 0x00          | —            | —                       | Y                         | IMM               | Switch to bitstream mode to load bitstream through target interface.                                                                                                              |
| CRAM_INIT_ADDRESS  | 0x06  | Level, Zone | 0x00           | 0x00          | —            | —                       | Y                         | IMM               | Initialize the address counter to zero.                                                                                                                                           |
| CRAM_WRITE_ADDRESS | 0x07  | Level, Zone | 0x00           | 0x00          | 2            | —                       | Y                         | Data Write        | Write the 16-bit address register for random access.                                                                                                                              |
| CRAM_PROG_INC      | 0x08  | Level, Zone | NFRAMES [15:8] | NFRAMES [7:0] | Variable     | —                       | Y                         | Data Write        | Write the configuration data to the configuration memory frame at current address and post increment the address.<br><br># data bytes = $\sum_{i=1}^{NFRAMES} (frame\_len_i + 3)$ |
| CRAM_READ_INC      | 0x0a  | Level, Zone | NFRAMES [15:8] | NFRAMES [7:0] | —            | See note <sup>3</sup> . | Y                         | Read              | Read configuration memory data frames selected by the address register and post increment the address.<br>Target SPI port only supports x1 mode for this command.                 |
| INIT_READ_REG      | 0x0f  | 0x00        | NFRAMES [15:8] | NFRAMES [7:0] | 4            | See note <sup>4</sup>   | Y                         | Data Read         | Read INIT register data.                                                                                                                                                          |
| INIT_READ_EBR      | 0x0f  | 0x08        | NFRAMES [15:8] | NFRAMES [7:0] | 4            | See note <sup>5</sup>   | Y                         | Data Read         | Read EBR data.                                                                                                                                                                    |
| PROG_DONE          | 0x80  | 0x00        | 0x25           | CHECK_CRC     | —            | —                       | Y                         | IMM               | Configuration data programming is done.                                                                                                                                           |
| READ_IDCODE_PUB    | 0x01  | 0x01        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read 32-bit public IDCODE of the device.                                                                                                                                          |
| READ_IDCODE_PRV    | 0x01  | 0x02        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read 32-bit private IDCODE for the device.                                                                                                                                        |
| READ_UIDCODE_PUB_L | 0x01  | 0x03        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read bits[31:0] of TraceID[63:0]                                                                                                                                                  |
| READ_UIDCODE_PUB_H | 0x01  | 0x04        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read bits[63:32] of TraceID[63:0]                                                                                                                                                 |
| READ_SECURE_UID    | 0x84  | 0x00        | 0x28           | 0x00          | —            | 32                      | —                         | Read              | Read the device 256-bit Secure Unique ID.                                                                                                                                         |
| READ_USERCODE      | 0x01  | 0x05        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read 32-bit user code register.                                                                                                                                                   |
| READ_DR_USERCODE   | 0x01  | 0x06        | 0x00           | 0x00          | —            | 4                       | Y                         | Read              | Read dry-run user code shadow register.                                                                                                                                           |
| READ_STATUS0       | 0x01  | 0x07        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read Status Register 0.                                                                                                                                                           |
| READ_STATUS1       | 0x01  | 0x08        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read Status Register 1.                                                                                                                                                           |
| READ_STATUS2       | 0x01  | 0x09        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read Status Register 2.                                                                                                                                                           |
| CHECK_BUSY         | 0x01  | 0x0b        | 0x00           | 0x00          | —            | 4                       | —                         | Read              | Read busy flag from Status Register 0.                                                                                                                                            |
| PROG_CNTRL0        | 0x80  | 0x08        | 0x03           | 0x00          | 8            | —                       | Y                         | Data Write        | Write Control Register 0.                                                                                                                                                         |
| READ_CNTRL0        | 0x84  | 0x00        | 0x04           | 0x00          | —            | 4                       | —                         | READ              | Read Control Register 0.                                                                                                                                                          |

| Command Name        | Byte0 | Byte1 | Byte2 | Byte3 | # DATA BYTES | # RETURN BYTES | PROG_QUALIFY <sup>1</sup> | Type <sup>2</sup> | Description                                              |
|---------------------|-------|-------|-------|-------|--------------|----------------|---------------------------|-------------------|----------------------------------------------------------|
| PROG_CNTRL1         | 0x80  | 0x08  | 0x05  | 0x00  | 8            | —              | Y                         | Data Write        | Write Control Register 1.                                |
| READ_CNTRL1         | 0x84  | 0x00  | 0x06  | 0x00  | —            | 4              | —                         | Read              | Read Control Register 1.                                 |
| READ_UMR            | 0x84  | 0x00  | 0x08  | 0x00  | —            | 16             | —                         | Read              | Read user mode register.                                 |
| DEVICE_CTRL         | 0x80  | 0x00  | 0x01  | CMD   | —            | —              | —                         | IMM               | Device control.                                          |
| DRY_RUN_CTRL        | 0x80  | 0x00  | 0x09  | MODE  | —            | —              | —                         | IMM               | Dry run control.                                         |
| REFRESH             | 0x80  | 0x00  | 0x0a  | 0x00  | —            | —              | —                         | Delay             | Equivalent to toggling the PROGRAMN pin.                 |
| SSPI_MODE           | 0x80  | 0x00  | 0x0f  | MODE  | —            | —              | —                         | IMM               | Change target SPI mode.                                  |
| MSPI_BRIDGE         | 0x16  | 0x00  | 0x00  | 0x00  | Variable     | —              | —                         | Data Write        | Bridge Data to Controller SPI.                           |
| MSPI_BRIDGE_CLOCK   | 0x80  | 0x00  | 0x1d  | MODE  | —            | —              | —                         | IMM               | Change controller SPI bridge clock settings.             |
| PORT_REQUEST        | 0x7a  | 0x00  | 0x00  | TYPE  | —            | —              | —                         | IMM               | Port ownership request.                                  |
| PORT_STATUS_READ    | 0x7b  | 0x00  | 0x00  | 0x00  | —            | 4              | —                         | Read              | Port ownership status.                                   |
| READ_SED_CRC        | 0x01  | 0x0c  | 0x00  | 0x00  | —            | 4              | Y                         | Read              | Read expected SED 32-bit CRC.                            |
| CALC_SED_CRC        | 0x80  | 0x00  | 0x4f  | 0x00  | —            | —              | —                         | Delay             | Run SED scan and calculate 32-bit SED CRC in the device. |
| READ_SED_CRC_CALC   | 0x01  | 0x0d  | 0x00  | 0x00  | —            | 4              | Y                         | Read              | Read 32-bit SED CRC calculated by the device.            |
| READ_OTP            | 0x84  | 0x00  | 0x4E  | ROW   | —            | 4              | —                         | Read              | Read user OTP.                                           |
| PROG_OTP            | 0x80  | 0x04  | 0x4C  | ROW   | 4            | —              | Y                         | Data Write        | Write user OTP.                                          |
| PROG_OTP_SHADOW     | 0x80  | 0x08  | 0x55  | ROW   | 8            | —              | Y                         | Data Write        | Write user OTP shadow register.                          |
| LOCK_OTP_ROW        | 0x80  | 0x00  | 0x4D  | ROW   | —            | —              | Y                         | IMM               | Lock user OTP row.                                       |
| LOCK_USER_OTP_BLOCK | 0x80  | 0x00  | 0x54  | 0x00  | —            | —              | Y                         | IMM               | Lock user block of OTP rows.                             |

**Notes:**

1. Refer to the [BACKGROUND\\_RECONFIG](#) section.
2. The different command types are described in the [Command Waveforms](#) section.
3. Refer to the [CRAM\\_READ\\_INC](#) section.
4. Refer to the [INIT\\_READ\\_REG](#) section.
5. Refer to the [INIT\\_READ\\_EBR](#) section.

## 6.10.4. Target Configuration Command Details

### 6.10.4.1. PROG\_ENABLE

The PROG\_ENABLE command puts the device in programming mode.

**Table 6.13. PROG\_ENABLE Command**

| Byte0 | Byte1 | Byte2 | Byte3     |
|-------|-------|-------|-----------|
| 0x80  | 0x00  | 0x0C  | 0x00–0x02 |

### 6.10.4.2. PROG\_DISABLE

If the prog\_done bit is set (see the PROG\_DONE command) and the device is in configuration mode, the PROG\_DISABLE command starts the wake-up sequence at the end of which the device enters user function mode. If the prog\_done bit is set and the device is in user function mode, the device continues to operate in user function mode. If the prog\_done bit is not set, this command causes the device to enter configuration mode.

**Table 6.14. PROG\_DISABLE Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x0E  | 0x00  |

### 6.10.4.3. BITSTREAM\_BURST

Sending the BITSTREAM\_BURST command through a target interface (target SPI, JTAG, LMMI) puts the device in bitstream mode. After sending this command, a valid bitstream must be sent. After the full bitstream is sent, the device automatically switches back to command mode.

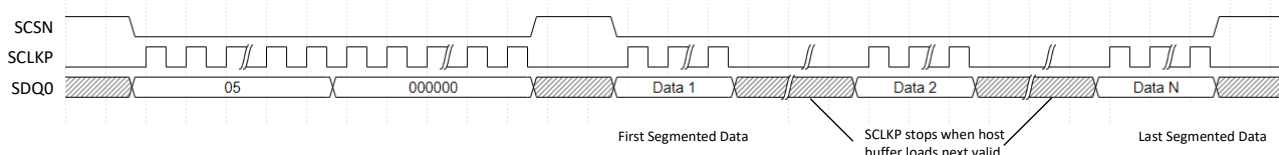
**Table 6.15. BITSTREAM\_BURST Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x05  | 0x00  | 0x00  | 0x00  |

There are two bitstream burst modes for configuring the device using target SPI mode: continuous and segmented. Continuous bitstream burst is recommended if conditions allow (adequate buffer size on host, host availability). In continuous bitstream burst mode, the host asserts SCSN low, then sends the bitstream data continuously after the BITSTREAM\_BURST command. The host de-asserts SCSN high at the end of the bitstream burst.

Refer to the following steps for the segmented bitstream burst mode:

1. Host asserts SCSN low.
2. Clock in the BITSTREAM\_BURST command. Host de-asserts SCSN high.
3. Host asserts SCSN low. Clock in the segmented bitstream. The segmented bitstream can be any size. For example, the bitstream can be split into 4096 bytes per data block and sent to the FPGA block by block. In between the loading of bitstream segments, SCLKP must remain idle or stop toggling when a valid segmented bitstream is unavailable or the host buffer is in the process of loading the next valid segmented bitstream on the SDQ0 (x1 mode) interface (refer to Figure 6.16).
4. Host de-asserts SCSN high at the end of the last segmented bitstream.



**Figure 6.16. Segmented Bitstream Burst Mode Timing Waveforms**

#### 6.10.4.4. CRAM\_INIT\_ADDRESS

The CRAM\_INIT\_ADDRESS command resets the address register in the specified configuration memory region to point to the first data frame in that region. The region is specified in OP1[4:0]. If OP1[4:0] == 0x1F, the address registers in all regions are reset (broadcast write).

**Table 6.16. CRAM\_INIT\_ADDRESS Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x06  | OP1   | 0x00  | 0x00  |

**Table 6.17. CRAM\_INIT\_ADDRESS Operand**

| Operand | Field | Description |
|---------|-------|-------------|
| OP1     | [7:5] | Reserved    |
|         | [4]   | Level       |
|         | [3:0] | Zone        |

#### 6.10.4.5. CRAM\_WRITE\_ADDRESS

The CRAM\_WRITE\_ADDRESS command writes the specified starting frame address to the specified configuration memory region. The region is specified in OP1[4:0]. If OP1[4:0] == 0x1F, the address registers in all regions are written with the specified address (broadcast write).

**Table 6.18. CRAM\_WRITE\_ADDRESS Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data                  |
|-------|-------|-------|-------|-----------------------|
| 0x07  | OP1   | 0x00  | 0x00  | ADDR[15:8], ADDR[7:0] |

**Table 6.19. CRAM\_WRITE\_ADDRESS Operand**

| Operand | Field | Description |
|---------|-------|-------------|
| OP1     | [7:5] | Reserved    |
|         | [4]   | Level       |
|         | [3:0] | Zone        |

#### 6.10.4.6. CRAM\_PROG\_INC

The CRAM\_PROG\_INC command writes configuration data into the destination configuration memory frames. The starting value for the frame address in each region is set using the CRAM\_INIT\_ADDRESS or CRAM\_WRITE\_ADDRESS command. The starting value for the destination region is set by OP1[4:0]. After each frame of data, the address register in the target region and the zone number are incremented. When the zone number reaches the maximum value for the target device and resets to 0, the level bit is toggled. Command byte2 and byte3 specify the total number of frames to be programmed. The length of each frame and the number of region levels and zones vary depending on the device size. See Table B.2 for details.

At the end of each frame there is a 16-bit CRC followed by a 1-byte padding of 0x00. The 16-bit CRC uses the polynomial  $x^{16} + x^{15} + x^2 + 1$  and is calculated over the data in that frame.

**Table 6.20. CRAM\_PROG\_INC Command**

| Byte0 | Byte1 | Byte2         | Byte3        | Data                                                                                                                                        |
|-------|-------|---------------|--------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| 0x08  | OP1   | NFRAMES[15:8] | NFRAMES[7:0] | Frame0 Data, CRC[15:8], CRC[7:0], 0x00,<br>Frame1 Data, CRC[15:8], CRC[7:0], 0x00,<br>...<br>FrameNFRAMES-1 Data, CRC[15:8], CRC[7:0], 0x00 |

**Table 6.21. CRAM\_PROG\_INC Operand**

| Operand | Bits  | Description |
|---------|-------|-------------|
| OP1     | [7:5] | Reserved    |
|         | [4]   | Level       |
|         | [3:0] | Zone        |

#### 6.10.4.7. CRAM\_READ\_INC

The CRAM\_READ\_INC command reads out the configuration memory frames from the specified region at the current frame address in that region. The frame address is incremented after each frame is read. The configuration memory region is specified in OP1[4:0]. The zone and level do not increment after each frame is read. The frame byte count and number of region levels and zones vary depending on the device size. See Table B.2 for details.

The following equation determines the number of bytes that are sent back from the device as a result of executing this command:

$$\text{Readback Size} = (\text{Frame Byte Count}) \times \text{NFRAMES}$$

**Table 6.22. CRAM\_READ\_INC Command**

| Byte0 | Byte1 | Byte2         | Byte3        |
|-------|-------|---------------|--------------|
| 0x0A  | OP1   | NFRAMES[15:8] | NFRAMES[7:0] |

**Table 6.23. CRAM\_READ\_INC Operand**

| Operand | Bits  | Description |
|---------|-------|-------------|
| OP1     | [7:5] | reserved    |
|         | [4]   | Level       |
|         | [3:0] | Zone        |

**Table 6.24. CRAM\_READ\_INC Return Data**

| Return Data Format                           |
|----------------------------------------------|
| Frame0 Data, Frame1 Data, ..., FrameN-1 Data |

#### 6.10.4.8. INIT\_READ\_REG

The INIT\_READ\_REG command reads data from one or more INIT registers starting at the specified ID and address. The address is auto incremented for each frame. The command returns the specified number of frames as read data. Each frame is 2 bytes long.

**Table 6.25. INIT\_READ\_REG Command**

| Byte0 | Byte1 | Byte2         | Byte3        | Data                                         |
|-------|-------|---------------|--------------|----------------------------------------------|
| 0x0F  | 0x00  | NFRAMES[15:8] | NFRAMES[7:0] | ID1[15:8], ID0[7:0], ADDR1[15:8], ADDR0[7:0] |

**Table 6.26. INIT\_READ\_REG Return Data**

| Return Data Format              |
|---------------------------------|
| Frame1[15:0], ..., FrameN[15:0] |

#### 6.10.4.9. INIT\_READ\_EBR

The INIT\_READ\_EBR command reads data from an EBR at the specified ID and address. The address is auto incremented for each frame. The command returns the specified number of frames as read data. Each frame is 9 bytes (72 bits) long.

**Table 6.27. INIT\_READ\_EBR Command**

| Byte0 | Byte1 | Byte2         | Byte3        | Data                                         |
|-------|-------|---------------|--------------|----------------------------------------------|
| 0x0F  | 0x80  | NFRAMES[15:8] | NFRAMES[7:0] | ID1[15:8], ID0[7:0], ADDR1[15:8], ADDR0[7:0] |

**Table 6.28. INIT\_READ\_EBR Return Data**

| Return Data                     |
|---------------------------------|
| Frame1[71:0], ..., FrameN[71:0] |

#### 6.10.4.10. PROG\_DONE

The PROG\_DONE command conditionally sets the prog\_done bit to indicate the configuration data programming is done. If the Check CRC bit is 1, the CRC checksum previously calculated by the CALC\_SED\_CRC command is compared with the expected CRC checksum previously programmed by the PROG\_SED\_CRC command. An error is flagged if the values do not match. If the values match or if the Check CRC bit is 0, the prog\_done bit is set if the device is in configuration mode.

**Table 6.29. PROG\_DONE Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x25  | OPR   |

**Table 6.30. PROG\_DONE Operand**

| Operand | Bits  | Description |
|---------|-------|-------------|
| OPR     | [7:1] | Reserved    |
|         | [0]   | Check CRC   |

#### 6.10.4.11. READ\_IDCODE\_PUB

The READ\_IDCODE\_PUB command reads out the 32-bit public IDCODE of the device. If the CUST\_IDCODE\_EN bit is set in OTP, the CUSTOMER\_IDCODE stored in OTP is returned. If CUST\_IDCODE\_EN is clear, the hardware IDCODE is returned.

**Table 6.31. READ\_IDCODE\_PUB Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x01  | 0x00  | 0x00  |

**Table 6.32: READ\_IDCODE\_PUB Return Data**

| Return Data      |
|------------------|
| IDCODE_PUB[31:0] |

#### 6.10.4.12. READ\_IDCODE\_PRV

The READ\_IDCODE\_PRV command reads out the 32-bit private (hardware) IDCODE of the device. The hardware IDCODE (also known as the Device ID) is fixed in the device and is not customer modifiable.

**Table 6.33. READ\_IDCODE\_PRV Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x02  | 0x00  | 0x00  |

**Table 6.34. READ\_IDCODE\_PRV Return Data**

| Return Data      |
|------------------|
| IDCODE_PRV[31:0] |

#### 6.10.4.13. READ\_UIDCODE\_PUB\_L

The READ\_UIDCODE\_PUB\_L command reads out bits[31:0] of the 64-bit TraceID.

**Table 6.35. READ\_UIDCODE\_PUB\_L Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x03  | 0x00  | 0x00  |

**Table 6.36. READ\_UIDCODE\_PUB\_L Return Data**

| Return Data    |
|----------------|
| TRACE_ID[31:0] |

#### 6.10.4.14. READ\_UIDCODE\_PUB\_H

The READ\_UIDCODE\_PUB\_H command reads out bits[63:32] of the 64-bit TraceID.

**Table 6.37. READ\_UIDCODE\_PUB\_H Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x04  | 0x00  | 0x00  |

**Table 6.38. READ\_UIDCODE\_PUB\_H Return Data**

| Return Data     |
|-----------------|
| TRACE_ID[63:32] |

#### 6.10.4.15. READ\_SECURE\_UID

The READ\_SECURE\_UID command reads out the 256-bit device Secure Unique ID. The Secure Unique ID is unique for every physical device and is derived from a Physically Unclonable Function (PUF) to guarantee uniqueness and device authenticity.

**Table 6.39. READ\_SECURE\_UID Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x84  | 0x00  | 0x28  | 0x00  |

**Table 6.40. READ\_SECURE\_UID Return Data**

| Return Data       |
|-------------------|
| SECURE_UID[255:0] |

#### 6.10.4.16. READ\_USERCODE

The READ\_USERCODE command reads out the 32-bit User Electronic Signature (UES) programmed in the register by the PROG\_USERCODE command in the bitstream.

**Table 6.41. READ\_USERCODE Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x05  | 0x00  | 0x00  |

**Table 6.42. READ\_USERCODE Return Data**

| Return Data    |
|----------------|
| USERCODE[31:0] |

#### 6.10.4.17. READ\_DR\_USERCODE

The READ\_DR\_USERCODE command reads out the dry run User Electronic Signature (UES) register. When dry run booting is executed using the DRY\_RUN\_CTRL command, the UES in the bitstream is redirected to the dry run UES shadow register. To read the UES of the currently loaded bitstream, use the READ\_USERCODE command. If dry run was not executed, the read default is 0x00000000.

**Table 6.43. READ\_DR\_USERCODE Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x06  | 0x00  | 0x00  |

**Table 6.44. READ\_DR\_USERCODE Return Data**

| Return Data       |
|-------------------|
| DR_USERCODE[31:0] |

#### 6.10.4.18. READ\_STATUS0

The READ\_STATUS0 command reads out the internal status bits from Status Register 0. Refer to the [Status Register 0](#) section for the bit definitions.

**Table 6.45. READ\_STATUS0 Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x07  | 0x00  | 0x00  |

**Table 6.46. READ\_STATUS0 Return Data**

| Return Data   |
|---------------|
| STATUS0[31:0] |

#### 6.10.4.19. READ\_STATUS1

The READ\_STATUS1 command reads out the internal status bits from Status Register 1. Refer to the [Status Register 1](#) section for the bit definitions.

**Table 6.47. READ\_STATUS1 Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x08  | 0x00  | 0x00  |

**Table 6.48. READ\_STATUS1 Return Data**

| Return Data   |
|---------------|
| STATUS1[31:0] |

#### 6.10.4.20. READ\_STATUS2

The READ\_STATUS2 command reads out the internal status bits from Status Register 2. Refer to the [Status Register 2](#) section for the bit definitions.

**Table 6.49. READ\_STATUS2 Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x09  | 0x00  | 0x00  |

**Table 6.50. READ\_STATUS2 Return Data**

| Return Data   |
|---------------|
| STATUS2[31:0] |

#### 6.10.4.21. CHECK\_BUSY

The CHECK\_BUSY command reads out the busy flag from Status Register 0 to check the command execution status.

**Table 6.51. CHECK\_BUSY Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x0B  | 0x00  | 0x00  |

**Table 6.52. CHECK\_BUSY Return Data**

| Return Data                             |
|-----------------------------------------|
| Bit 31: 0                               |
| ...                                     |
| Bit 1: 0                                |
| Bit 0: Busy flag from Status Register 0 |

#### 6.10.4.22. PROG\_CNTRL0

The PROG\_CNTRL0 command writes to Control Register 0. The first 32 bits are data to write to Control Register 0, and the remaining 32 bits are mask bits. Bits with mask set to 1 retain their current values. This instruction does not check for any errors.

$\text{New\_CR0} = (\text{Old\_CR0} \& \text{MASK}) \mid (\text{DATA} \& \sim \text{MASK})$

Example to change CR0[0] to 0:

Current Register Value = 0x000000AB

Data = 0x00000000

Mask = 0xFFFFFFFF

New Register Value = 0x000000AA

**Table 6.53. PROG\_CNTRL0 Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data                                                                                                      |
|-------|-------|-------|-------|-----------------------------------------------------------------------------------------------------------|
| 0x80  | 0x08  | 0x03  | 0x00  | DATA[31:24], DATA[23:16], DATA[15:8],<br>DATA[7:0],<br>MASK[31:24], MASK[23:16], MASK[15:8],<br>MASK[7:0] |

#### 6.10.4.23. READ\_CNTRL0

The READ\_CNTRL0 command reads out Control Register 0. Refer to the [Control Register 0 \(CR0\)](#) section for the bit definitions.

**Table 6.54. READ\_CNTRL0 Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x84  | 0x00  | 0x04  | 0x00  |

**Table 6.55. READ\_CNTRL0 Return Data**

| Return Data |
|-------------|
| CR0[31:0]   |

#### 6.10.4.24. PROG\_CNTRL1

The PROG\_CNTRL1 command writes to Control Register 1. The first 32 bits are data to write to Control Register 1, and the remaining 32 bits are mask bits. MASK bits set to 1 indicate bits to retain their previous values; MASK bits set to 0 indicate bits to be updated based on DATA[31:0]. Refer to the [Control Register 1 \(CR1\)](#) section for the bit definitions.

New\_CR1 = (Old\_CR1 & MASK) | (DATA & ~MASK)

**Table 6.56. PROG\_CNTRL1 Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data                                                                                                      |
|-------|-------|-------|-------|-----------------------------------------------------------------------------------------------------------|
| 0x80  | 0x08  | 0x05  | 0x00  | DATA[31:24], DATA[23:16], DATA[15:8],<br>DATA[7:0],<br>MASK[31:24], MASK[23:16], MASK[15:8],<br>MASK[7:0] |

#### 6.10.4.25. READ\_CNTRL1

The READ\_CNTRL1 command reads out Control Register 1. Refer to the [Control Register 1 \(CR1\)](#) section for the bit definitions.

**Table 6.57. READ\_CNTRL1 Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x84  | 0x00  | 0x06  | 0x00  |

**Table 6.58. READ\_CNTRL1 Return Data**

| Return Data |
|-------------|
| CR1[31:0]   |

#### 6.10.4.26. READ\_UMR

The READ\_UMR command reads out the user mode register. The bits in the user mode register control the behavior of the device after entering user function mode. Refer to the [User Mode Register](#) section for the bit definitions.

**Table 6.59. READ\_UMR Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x84  | 0x00  | 0x08  | 0x00  |

**Table 6.60. READ\_UMR Return Data**

| Return Data |
|-------------|
| UMR[127:0]  |

#### 6.10.4.27. DEVICE\_CTRL

The DEVICE\_CTRL command performs real-time control of actions on the device. This allows external controller devices to control certain operations on the device. To provide a convenient way to test the dual boot setup, when a DEVICE\_CTRL command is executed with OPR set to 0x7F, an internal flag is set to bypass the primary boot and directly execute the secondary boot. This internal flag can be cleared by sending a DEVICE\_CTRL command with OPR != 0x7F.

**Table 6.61. DEVICE\_CTRL Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x01  | OPR   |

**Table 6.62. DEVICE\_CTRL Operand**

| Operand | Field | Description                                                                                                                                                                                                                                      |
|---------|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OPR     | [7:4] | Reserved                                                                                                                                                                                                                                         |
|         | [3]   | CFG_RST: Cause a reset on configuration logic.                                                                                                                                                                                                   |
|         | [2]   | Reserved                                                                                                                                                                                                                                         |
|         | [1]   | CRC_CHECK: Launch a one-time check of SED. If SED is already running (including continuous SED mode) then ignore this command. This command allows a READ_CRC_SED command to be issued later in order to determine if an SED error has occurred. |
|         | [0]   | SYS_RST: Cause a global set/reset to occur on the device.                                                                                                                                                                                        |

#### 6.10.4.28. DRY\_RUN\_CTRL

The DRY\_RUN\_CTRL command launches dry-run boot. Dry-run boot loads the bitstream and checks the CRC of the bitstream without writing the configuration data. This is done in the background during normal device operation. The user code is programmed to the dry-run user code.

**Table 6.63. DRY\_RUN\_CTRL Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x09  | OPR   |

**Table 6.64. DRY\_RUN\_CTRL Operand**

| Operand | Field | Description                                                                                                                                    |
|---------|-------|------------------------------------------------------------------------------------------------------------------------------------------------|
| OPR     | [7:2] | Reserved                                                                                                                                       |
|         | [1:0] | Dry-run mode<br>00 – No dry-run<br>01 – Dry-run for primary bitstream<br>10 – Dry-run for secondary bitstream<br>11 – Target interface dry-run |

#### 6.10.4.29. REFRESH

The REFRESH command is equivalent to pulsing the PROGRAMN pin.

**Table 6.65. REFRESH Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x0A  | 0x00  |

#### 6.10.4.30. SSPI\_MODE

The SSPI\_MODE command changes the target SPI operating mode. The target SPI operating mode remains in effect until PROGRAMN pulsing, REFRESH command execution, device power cycle, or another SSPI\_MODE command is sent. The device starts in x1 operating mode upon power-up or after PROGRAMN pulsing or REFRESH command execution.

**Note:** Other target SPI parameters (such as phase and polarity) are controlled by Control Register 1.

**Table 6.66. SSPI\_MODE Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x0F  | OPR   |

**Table 6.67. SSPI\_MODE Operand**

| Operand | Field | Description                                                                                         |
|---------|-------|-----------------------------------------------------------------------------------------------------|
| OPR     | [7:2] | Reserved                                                                                            |
|         | [1:0] | Target SPI mode<br>00 – x1 (SPI)<br>01 – x2 (DSPI)<br>10 – x4 (QSPI)<br>11 – x8 DDR (xSPI 8D-8D-8D) |

#### 6.10.4.31. MSPI\_BRIDGE

The MSPI\_BRIDGE command enables the bridge to the controller SPI port and sends DATA\_BYTES through the controller SPI port. Data returned by the controller SPI port is forwarded to the data output of the requesting target interface (SSPI, LMMI). All DATA\_BYTES are sent to the controller SPI port without any parsing. The controller SPI chip select is active for the duration of the MSPI\_BRIDGE command and is de-asserted at the end of the command.

The MSPI\_BRIDGE command is terminated in different ways for each requesting interface as follows:

- On the target SPI port, the command is terminated when SCSN is de-asserted.
- On the LMMI interface, the command is terminated when LMMI\_REQUEST is de-asserted.

The MSPI\_BRIDGE command is not supported on the JTAG interface. The JTAG interface implements a separate mechanism for controller SPI bridging that uses a JTAG instruction (PROG\_SPI).

**Table 6.68. MSPI\_BRIDGE Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data             |
|-------|-------|-------|-------|------------------|
| 0x16  | 0x00  | 0x00  | 0x00  | DATA_BYTES[N]... |

#### 6.10.4.32. MSPI\_BRIDGE\_CLOCK

The MSPI\_BRIDGE\_CLOCK command sets the controller SPI clock source and clock frequency for MSPI\_BRIDGE commands.

**Table 6.69. MSPI\_BRIDGE\_CLOCK Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x1D  | OPR   |

**Table 6.70. MSPI\_BRIDGE\_CLOCK Operand**

| Operand | Field | Description                                                                                |
|---------|-------|--------------------------------------------------------------------------------------------|
| OPR     | [7:2] | Clock divider ( $N_{div}$ , setup by Radiant)                                              |
|         | [1:0] | Clock source ( $F_{osc}$ )<br>00 – 450 MHz<br>01 – 360 MHz<br>10 – 300 MHz<br>11 – 240 MHz |

#### 6.10.4.33. PORT\_REQUEST

The PORT\_REQUEST command requests or releases exclusive access for the port it is sent to. This command is supported by all target interfaces (target SPI, JTAG, LMMI). When a port is granted exclusive access, all input data from other target interfaces is ignored. If a port does not have exclusive access, data that is received over multiple target interfaces simultaneously may conflict, resulting in unexpected device behavior.

**Table 6.71. PORT\_REQUEST Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x7A  | 0x00  | 0x00  | OP3   |

**Table 6.72. PORT\_REQUEST Operand**

| Operand | Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OP3     | [7:2] | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|         | [1:0] | Request type<br>00 – Nowait request for exclusive access<br>Request returns quickly with success/failure status<br>01 – Wait request for exclusive access<br>Request stays active until either a) exclusive access is granted to this port, b) exclusive access is granted to a different port, or c) a timeout is reached<br>10 – Force request for exclusive access<br>Always succeeds, even if another port previously had exclusive access<br>11 – Release exclusive access<br>All ports are available/active |

#### 6.10.4.34. PORT\_STATUS\_READ

The PORT\_STATUS\_READ command reads out the port status for the specified target interface.

**Table 6.73. PORT\_STATUS\_READ Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x7B  | 0x00  | 0x00  | 0x00  |

**Table 6.74. PORT\_STATUS\_READ Return Data**

| Return Data |             |                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Field       | Name        | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
| [31:4]      | reserved    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| [3]         | cfg_mode    | Configuration mode<br>0 – command mode<br>1 – bitstream mode                                                                                                                                                                                                                                                                                                                                                                              |
| [2]         | busy        | PORT_REQUEST command is being processed                                                                                                                                                                                                                                                                                                                                                                                                   |
| [1:0]       | port_active | This bitfield is only valid when busy == 0<br>00 – AVAILABLE: This port is active but non-exclusive. Simultaneous access from multiple ports may cause conflicts and unexpected device behavior.<br>01 – EXCLUSIVE: This port is active and has exclusive control<br>10 – INACTIVE: This port is inactive and a different port is active<br>11 – DISABLED: This port is permanently inactive (for example, if a hard_lock OTP bit is set) |

#### 6.10.4.35. READ\_SED\_CRC

The READ\_SED\_CRC command reads out the expected 32-bit CRC checksum for the full-chip configuration memory from internal registers previously programmed by the bitstream.

**Table 6.75. READ\_SED\_CRC Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x0C  | 0x00  | 0x00  |

**Table 6.76. READ\_SED\_CRC Return Data**

| Return Data   |
|---------------|
| SED_CRC[31:0] |

#### 6.10.4.36. CALC\_SED\_CRC

The CALC\_SED\_CRC command runs an SED scan of the full chip configuration memory and calculates the 32-bit CRC checksum. While the CRC calculation is in progress, the busy flag bit in Status Register 0 (STATUS0) is set. Once the busy flag bit is 0, the CRC result can be read using the READ\_SED\_CRC\_CALC command.

**Table 6.77. CALC\_SED\_CRC Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x4F  | 0x00  |

#### 6.10.4.37. READ\_SED\_CRC\_CALC

The READ\_SED\_CRC\_CALC command reads out the calculated 32-bit CRC checksum for the full chip configuration memory generated by CALC\_SED\_CRC.

**Table 6.78. READ\_SED\_CRC\_CALC Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x01  | 0x0D  | 0x00  | 0x00  |

**Table 6.79. READ\_SED\_CRC\_CALC Return Data**

| Return Data        |
|--------------------|
| SED_CRC_CALC[31:0] |

#### 6.10.4.38. READ\_OTP

The READ\_OTP command reads out the OTP row specified in OPR and returns the 32-bit data value.

**Table 6.80. READ\_OTP Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x84  | 0x00  | 0x4E  | OPR   |

**Table 6.81. READ\_OTP Operand**

| Operand | Field | Description    |
|---------|-------|----------------|
| OPR     | [7:0] | OTP row number |

**Table 6.82. READ\_OTP Return Data**

| Return Data    |
|----------------|
| OTP_DATA[31:0] |

#### 6.10.4.39. PROG\_OTP

The PROG\_OTP command writes the 32-bit DATA value to the user OTP row specified in OPR. The DATA value is bitwise OR-ed with the existing OTP row contents, so OTP bits can only be set to 1, never cleared to 0.

**Table 6.83. PROG\_OTP Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data                                            |
|-------|-------|-------|-------|-------------------------------------------------|
| 0x80  | 0x04  | 0x4C  | OPR   | DATA[31:24], DATA[23:16], DATA[15:8], DATA[7:0] |

**Table 6.84. PROG\_OTP Operand**

| Operand | Field | Description         |
|---------|-------|---------------------|
| OPR     | [7:0] | User OTP row number |

#### 6.10.4.40. PROG\_OTP\_SHADOW

The PROG\_OTP\_SHADOW command reads the value at the user OTP row specified in OPR, masks the value with MASK[31:0], bitwise ORs the result with DATA[31:0], and writes the resulting value to the shadow register corresponding to the user OTP row specified in OPR. MASK bits set to 1 indicate bits to retain their previous values; MASK bits set to 0 indicate bits to be updated based on DATA[31:0]. OTP contents are *not* updated by this command; only the shadow register is updated. The shadow register is only updated if the row lock for the specified user OTP row is 0.

```
If (row_lock(ROW) == 0)
    shadow_register(ROW) = (otp_value(ROW) & MASK) | (DATA & ~MASK)
```

**Table 6.85. PROG\_OTP\_SHADOW Command**

| Byte0 | Byte1 | Byte2 | Byte3 | Data                                                                                                      |
|-------|-------|-------|-------|-----------------------------------------------------------------------------------------------------------|
| 0x80  | 0x08  | 0x55  | OPR   | DATA[31:24], DATA[23:16], DATA[15:8],<br>DATA[7:0],<br>MASK[31:24], MASK[23:16], MASK[15:8],<br>MASK[7:0] |

**Table 6.86. PROG\_OTP\_SHADOW Operand**

| Operand | Field | Description                     |
|---------|-------|---------------------------------|
| OPR     | [7:0] | User Shadow Register row number |

#### 6.10.4.41. LOCK\_OTP\_ROW

The LOCK\_OTP\_ROW command locks the user OTP row specified in OPR. Once a row is locked, that OTP row and corresponding shadow register can no longer be written.

**Table 6.87. LOCK\_OTP\_ROW Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x4D  | OPR   |

**Table 6.88. LOCK\_OTP\_ROW Operand**

| Operand | Field | Description         |
|---------|-------|---------------------|
| OPR     | [7:0] | User OTP row number |

#### 6.10.4.42. LOCK\_USER\_OTP\_BLOCK

The LOCK\_USER\_OTP\_BLOCK command locks the block of OTP rows containing resource locks, port locks, and security feature settings, preventing further modification. The command locks each of the OTP rows in the block and stores a CRC of the block contents in OTP for integrity checking during device power-up.

**Table 6.89. LOCK\_USER\_OTP\_BLOCK Command**

| Byte0 | Byte1 | Byte2 | Byte3 |
|-------|-------|-------|-------|
| 0x80  | 0x00  | 0x54  | 0x00  |

## 6.11. Self-Download Mode

**Note:** Only Mach-N2 devices support the self-download mode.

The advantages of the self-download configuration mode include:

- **Speed:** The Mach-N2 device is ready to run in a few milliseconds depending on the density of the device.
- **Security:** The configuration data is never seen outside the device during the load to SRAM. Internal memory can be prevented from being read.
- **Reduced cost:** External non-volatile memory specifically reserved for device programming is not required.
- **Reduced board space:** Elimination of external non-volatile memory allows the board to be smaller.

The Mach-N2 device retrieves the configuration data from the internal flash memory in self-download mode. SDM is triggered when power is applied, a REFRESH command is received, or by asserting the PROGRAMN pin.

## 6.12. Flash Memory Space Partition

**Note:** Only Mach-N2 devices contain internal (on-chip) flash memory.

Each Mach-N2 device contains a certain quantity of internal memory. The amount of memory depends on the device density. Refer to [Flash Memory Space Partition](#) for the number of internal memory blocks available for selected Mach-N2 devices.



## 7.2. sysCONFIG Options

Table 7.1 shows the software selectable sysCONFIG options with default and all available settings. For more information on the options and settings, refer to the individual sysCONFIG option sections.

**Table 7.1. sysCONFIG Options**

| Option Name                   | Default Setting       | All Settings                                                                                   |
|-------------------------------|-----------------------|------------------------------------------------------------------------------------------------|
| SLAVE_SPI_PORT                | DISABLE               | DISABLE, SERIAL, DUAL, QUAD, XSPI <sup>1</sup>                                                 |
| MASTER_SPI_PORT               | DISABLE               | DISABLE, SERIAL, DUAL, QUAD, XSPI, XSPI_DIFF_CLK <sup>1</sup>                                  |
| MSPI_RESET_PORT               | DISABLE               | DISABLE, ENABLE                                                                                |
| MCCLK_FREQ                    | 3.1                   | 3.1, 7.1, 14.3, 28.6, 57.1, 66.7, 80.0, 100.0, 106.7, 133.3, 160.0                             |
| COMPRESS_CONFIG               | OFF                   | OFF, ON <sup>1</sup>                                                                           |
| BOOT_SEL                      | DUAL                  | DUAL, SINGLE                                                                                   |
| MSPI_RESET                    | DISABLE               | DISABLE, ENABLE <sup>1</sup>                                                                   |
| MULTI_BOOT_MODE               | DISABLE               | DISABLE, ENABLE <sup>1</sup>                                                                   |
| MULTI_BOOT_SEL                | STATIC                | STATIC, DYNAMIC <sup>1</sup>                                                                   |
| CONFIGIO_VOLTAGE_BANK1        | NOT_SPECIFIED         | NOT_SPECIFIED, 2.5, 1.2, 1.8, 3.3                                                              |
| CONFIGIO_VOLTAGE_BANK2        | NOT_SPECIFIED         | NOT_SPECIFIED, 2.5, 1.2, 1.8, 3.3                                                              |
| MSPI_SIGNATURE_TIMER          | 200MS                 | 200MS, 100MS, 50MS, 40MS, 20MS, 1MS, 500US, 100US, INFINITE                                    |
| MSPI_TX_EDGE                  | FALLING               | FALLING, RISING                                                                                |
| MSPI_RX_EDGE                  | RISING                | RISING, FALLING                                                                                |
| MSPI_CPHA                     | FIRST_EDGE            | FIRST_EDGE, SECOND_EDGE                                                                        |
| MSPI_CPOL                     | IDLE_CLOCK_LOW        | IDLE_CLOCK_LOW, IDLE_CLOCK_HIGH <sup>1</sup>                                                   |
| SSPI_TX_EDGE                  | FALLING               | FALLING, RISING                                                                                |
| SSPI_RX_EDGE                  | RISING                | RISING, FALLING                                                                                |
| SSPI_CPHA                     | FIRST_EDGE            | FIRST_EDGE, SECOND_EDGE                                                                        |
| SSPI_CPOL                     | IDLE_CLOCK_LOW        | IDLE_CLOCK_LOW, IDLE_CLOCK_HIGH <sup>1</sup>                                                   |
| SSPI_SHIFT_ORDER              | MSB_FIRST             | MSB_FIRST, LSB_FIRST <sup>1</sup>                                                              |
| SSPI_DAISY_CHAIN_MODE         | DISABLE               | DISABLE, SCM <sup>1</sup> , AUTO <sup>1</sup>                                                  |
| ERASE_EBR_ON_REFRESH          | DISABLE               | DISABLE, ENABLE <sup>1</sup>                                                                   |
| SIGNATURE_CHECK               | ENABLE_LSCC_SIGNATURE | ENABLE_LSCC_SIGNATURE, DISABLE, ENABLE_SFDP_SIGNATURE                                          |
| MSPI_ADDRESS_32BIT            | DISABLE               | DISABLE, ENABLE                                                                                |
| MSPI_COMMAND_32BIT            | DISABLE               | DISABLE, ENABLE                                                                                |
| SSPI_IDLE_TIMER               | DISABLE               | DISABLE, 200S, 100S, 50S, 25S, 10S, 5S, 1S, 750MS, 500MS, 250MS, 100MS, 75MS, 50MS, 25MS, 10MS |
| MSPI_PREAMBLE_DETECTION_TIMER | 200MS                 | 200MS, 100MS, 50MS, 40MS, 20MS, 1MS, 500US, 100US                                              |
| BACKGROUND_RECONFIG           | OFF                   | OFF, SRAM_ONLY <sup>1</sup> , ON <sup>1</sup>                                                  |
| DAISY_CHAIN                   | DISABLE               | DISABLE, BYPASS <sup>1</sup> , FLOW_THROUGH <sup>1</sup>                                       |
| DAISY_CHAIN_WAIT_DONE         | DISABLE               | DISABLE, ENABLE <sup>1</sup>                                                                   |
| TRANSFR                       | OFF                   | OFF, ON                                                                                        |
| UserCode Format               | Binary                | Binary, Hex, ASCII, Auto                                                                       |
| UserCode                      | 32'b0                 | 32-bit user code (user electronic signature)                                                   |
| MultiBoot Offset              | 48'b0                 | 48-bit address for external or internal flash memory                                           |
| BitstreamRevision Format      | Binary                | Binary, Hex, ASCII, Timestamp                                                                  |
| BitstreamRevision             | 32'b0                 | 32-bit bitstream revision                                                                      |

**Note:**

1. Setting is not available for ES1 silicon.

### 7.2.1. SLAVE\_SPI\_PORT

The SLAVE\_SPI\_PORT option controls the behavior of the target SPI configuration port after the device enters user function mode. SLAVE\_SPI\_PORT can be enabled at the same time as MASTER\_SPI\_PORT because the target SPI port and controller SPI port are in different I/O banks.

**Table 7.2. SLAVE\_SPI\_PORT Option**

| Option Name    | Setting(s)        | Description                                                                                                                                                                                                                                                                                                            |
|----------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLAVE_SPI_PORT | DISABLE (default) | Disconnect the SPI port pins from the configuration logic.                                                                                                                                                                                                                                                             |
|                | SERIAL            | Persist the standard serial SPI port I/O pins (SCLKP, SCSN, SMOSI, SMISO) in serial mode when the device is in user function mode. When the pins are persisted, an external SPI controller can interact with the configuration logic. This preference also prevents over-assigning I/O functions to these pins.        |
|                | DUAL              | Persist the SPI port I/O pins (SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1) in dual mode when the device is in user function mode. When the pins are persisted, an external SPI controller can interact with the configuration logic. This preference also prevents over-assigning I/O functions to these pins.                |
|                | QUAD              | Persist the SPI port I/O pins (SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1, SDQ[3:2]) in quad mode when the device is in user function mode. When the pins are persisted, an external SPI controller can interact with the configuration logic. This preference also prevents over-assigning I/O functions to these pins.      |
|                | XSPI              | Persist the SPI port I/O pins (SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1, SDQ[7:2], SDS) in xSPI mode when the device is in user function mode. When the pins are persisted, an external SPI controller can interact with the configuration logic. This preference also prevents over-assigning I/O functions to these pins. |

### 7.2.2. MASTER\_SPI\_PORT

The MASTER\_SPI\_PORT option controls the behavior of the controller SPI configuration port after the device enters user function mode.

**Table 7.3. MASTER\_SPI\_PORT Option**

| Option Name     | Setting(s)        | Description                                                                                                                                                                                                    |
|-----------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MASTER_SPI_PORT | DISABLE (default) | Disconnect the controller SPI port pins from the configuration logic. The controller SPI pins (MCLKP, MCLKN, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[7:2]) can be used as general purpose I/O pins.                  |
|                 | SERIAL            | Persist the SPI port I/O pins (MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1) in serial mode when the device is in user function mode. This preference also prevents over-assigning I/O functions to these pins.         |
|                 | DUAL              | Persist the SPI port I/O pins (MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1) in dual mode when the device is in user function mode. This preference also prevents over-assigning I/O functions to these pins.           |
|                 | QUAD              | Persist the SPI port I/O pins (MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[3:2]) in quad mode when the device is in user function mode. This preference also prevents over-assigning I/O functions to these pins. |
|                 | XSPI <sup>1</sup> | Persist the SPI port I/O pins (MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ [7:2], MDS) in xSPI mode when the device is in user function mode.                                                                     |
|                 | XSPI_DIFF_CLK     | Persist the SPI port I/O pins with differential clock (MCLKP, MCLKN, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[7:2], MDS) in xSPI mode when the device is in user function mode.                                       |

**Note:**

1. When MASTER\_SPI\_PORT = XSPI, you must leave the MSPI\_RX\_EDGE option setting at default, which is MSPI\_RX\_EDGE = RISING.

### 7.2.3. MSPI\_RESET\_PORT

The MSPI\_RESET\_PORT option specifies whether the MSPI\_RESET port will be available to be used for configuration purposes in user function mode.

**Table 7.4. MSPI\_RESET\_PORT Option**

| Option Name     | Setting(s)        | Description                                                                                                           |
|-----------------|-------------------|-----------------------------------------------------------------------------------------------------------------------|
| MSPI_RESET_PORT | DISABLE (default) | MRSTN pin will not be available for configuration in user function mode. It can be used as a general purpose I/O pin. |
|                 | ENABLE            | MRSTN pin will be available for configuration in user function mode.                                                  |

### 7.2.4. MCCLK\_FREQ

The MCCLK\_FREQ option controls the MCLKP frequency used to retrieve data from an external SPI storage device or internal flash memory when using the single or dual boot configuration mode.

**Table 7.5. MCCLK\_FREQ Option**

| Option Name | Setting(s)                                                    | Description                                                                                                                                                                                                                                                                                                                                                                      |
|-------------|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCCLK_FREQ  | 3.1 (default)                                                 | Default clock frequency used by the FPGA device (nominal 3.1 MHz $\pm$ 15%) to begin retrieving data from the external SPI storage device or internal flash memory. The MCCLK_FREQ value is stored in the bitstream.                                                                                                                                                             |
|             | 7.1, 14.3, 28.6, 57.1, 66.7, 80.0, 100.0, 106.7, 133.3, 160.0 | Set the MCCLK_FREQ value to the selected clock frequency. When the FPGA device reads the MCCLK_FREQ value in the bitstream, the device switches to the new clock frequency and then loads the bitstream using that frequency. Possible MCCLK_FREQ values range from 3.1 MHz to 160 MHz. The maximum frequency may be limited by the non-volatile memory device or system design. |

### 7.2.5. COMPRESS\_CONFIG

The COMPRESS\_CONFIG option alters the way files are generated with compressed FPGA data frames.

**Table 7.6. COMPRESS\_CONFIG Option**

| Option Name     | Setting(s)    | Description                                              |
|-----------------|---------------|----------------------------------------------------------|
| COMPRESS_CONFIG | OFF (default) | Turn off FPGA data frame compression in generated files. |
|                 | ON            | Turn on FPGA data frame compression in generated files.  |

### 7.2.6. BOOT\_SEL

The BOOT\_SEL option selects the device booting mode.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.7. BOOT\_SEL Option**

| Option Name | Setting(s)     | Description              |
|-------------|----------------|--------------------------|
| BOOT_SEL    | DUAL (default) | Select dual boot mode.   |
|             | SINGLE         | Select single boot mode. |

### 7.2.7. MSPI\_RESET

The MSPI\_RESET option enables the hardware reset signal from the configuration controller SPI to the flash device.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.8. MSPI\_RESET Option**

| Option Name | Setting(s)        | Description                                                                                  |
|-------------|-------------------|----------------------------------------------------------------------------------------------|
| MSPI_RESET  | DISABLE (default) | Disable the hardware reset signal from the configuration controller SPI to the flash device. |
|             | ENABLE            | Enable the hardware reset signal from the configuration controller SPI to the flash device.  |

### 7.2.8. MULTI\_BOOT\_MODE

The MULTI\_BOOT\_MODE option enables the multiple boot functionality of the device.

**Table 7.9. MULTI\_BOOT\_MODE Option**

| Option Name     | Setting(s)        | Description              |
|-----------------|-------------------|--------------------------|
| MULTI_BOOT_MODE | DISABLE (default) | Disable multi-boot mode. |
|                 | ENABLE            | Enable multi-boot mode.  |

### 7.2.9. MULTI\_BOOT\_SEL

The MULTI\_BOOT\_SEL option selects the booting address for the multiple boot event when MULTI\_BOOT\_MODE is set to ENABLE.

**Table 7.10. MULTI\_BOOT\_SEL Option**

| Option Name    | Setting(s)       | Description                                                                                                                          |
|----------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| MULTI_BOOT_SEL | STATIC (default) | Use the value stored in the MultiBoot Offset register as the booting address. Refer to the <a href="#">MultiBoot Offset</a> section. |
|                | DYNAMIC          | Use the multi-boot address register programmed through the CONFIG_LMMI interface.                                                    |

## 7.2.10. CONFIGIO\_VOLTAGE\_BANK1

The CONFIGIO\_VOLTAGE\_BANK1 option specifies the  $V_{CCIO}$  level for I/O bank 1. Setting this option informs the software of the voltage required at I/O bank 1 to meet the user's sysCONFIG requirements. The Radiant software can then generate DRC errors based on this setting and I/O type of the I/O pins that reside in I/O bank 1.

**Table 7.11. CONFIGIO\_VOLTAGE\_BANK1 Option**

| Option Name            | Setting(s)              | Description                                                                |
|------------------------|-------------------------|----------------------------------------------------------------------------|
| CONFIGIO_VOLTAGE_BANK1 | NOT_SPECIFIED (default) | $V_{CCIO}$ level for I/O bank 1 is not specified by default.               |
|                        | 2.5, 1.2, 1.8, 3.3      | Specify the $V_{CCIO}$ level for I/O bank 1. $V_{CCIO}$ level is in volts. |

## 7.2.11. CONFIGIO\_VOLTAGE\_BANK2

The CONFIGIO\_VOLTAGE\_BANK2 option specifies the  $V_{CCIO}$  level for I/O bank 2. Setting this option informs the software of the voltage required at I/O bank 2 to meet the user's sysCONFIG requirements. The Radiant software can then generate DRC errors based on this setting and the I/O type of the I/O pins that reside in I/O bank 2.

**Table 7.12. CONFIGIO\_VOLTAGE\_BANK2 Option**

| Option Name            | Setting(s)              | Description                                                                |
|------------------------|-------------------------|----------------------------------------------------------------------------|
| CONFIGIO_VOLTAGE_BANK2 | NOT_SPECIFIED (default) | $V_{CCIO}$ level for I/O bank 2 is not specified by default.               |
|                        | 2.5, 1.2, 1.8, 3.3      | Specify the $V_{CCIO}$ level for I/O bank 2. $V_{CCIO}$ level is in volts. |

## 7.2.12. MSPI\_SIGNATURE\_TIMER

The MSPI\_SIGNATURE\_TIMER option sets the maximum time to get a valid LSCC/SFDP signature from flash.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.13. MSPI\_SIGNATURE\_TIMER Option**

| Option Name          | Setting(s)                                 | Description                                                                                                                                                                                                                                                                                     |
|----------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSPI_SIGNATURE_TIMER | 200MS (default)                            | Default timer value. The unit of the timer value is indicated in the setting name, for example, <i>MS</i> for milliseconds. If no signature is detected before timeout, the configuration logic stops and signals a signature detection failure.                                                |
|                      | 100MS, 50MS, 40MS, 20MS, 1MS, 500US, 100US | Set timer value to the selected value. The unit of the timer value is indicated in the setting name for example <i>MS</i> for milliseconds and <i>US</i> for microseconds. If no signature is detected before timeout, the configuration logic stops and signals a signature detection failure. |
|                      | INFINITE                                   | No timer value. There is no time limit for getting a valid signature from flash.                                                                                                                                                                                                                |

### 7.2.13. MSPI\_TX\_EDGE

The MSPI\_TX\_EDGE option selects the controller SPI data transmitting clock edge.

**Note:** This setting is stored in the device non-volatile memory (OTP).

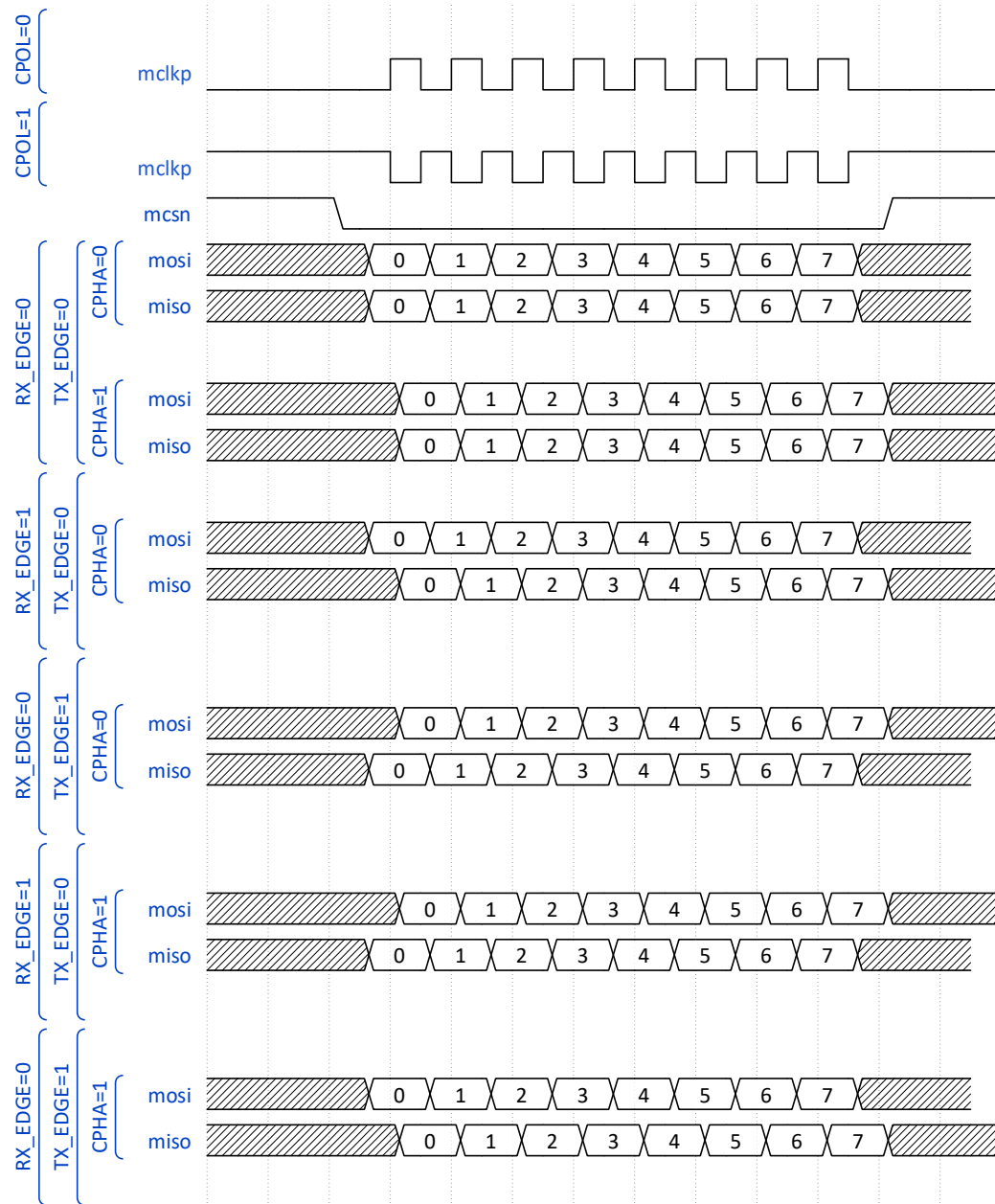
**Table 7.14. MSPI\_TX\_EDGE Option**

| Option Name  | Setting(s)        | Description                                                                                                                                                                                                                                        |
|--------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSPI_TX_EDGE | FALLING (default) | Set the controller SPI to transmit data on the inverted MCLKP edge. For example, if MSPI_CPOL = IDLE_CLOCK_LOW and MSPI_CPHA = FIRST_EDGE, the MCLKP edge is a rising leading edge. The inverted MCLKP edge is the falling trailing (second) edge. |
|              | RISING            | Set the controller SPI to transmit data on the MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA. For example, if MSPI_CPOL = IDLE_CLOCK_LOW and MSPI_CPHA = FIRST_EDGE, the MCLKP edge is a rising leading (first) edge.                           |

Table 7.15 lists the different combinations of CPOL, CPHA, and TX/RX\_EDGE settings and corresponding Tx and Rx operations. Figure 7.2 shows waveform examples for the different combinations of CPOL, CPHA, and TX/RX\_EDGE settings.

**Table 7.15. CPOL, CPHA, and TX/RX\_EDGE Settings versus Tx and Rx Operations**

| CPOL            | CPHA        | TX_EDGE | RX_EDGE | Tx Operation                    | Rx Operation                   |
|-----------------|-------------|---------|---------|---------------------------------|--------------------------------|
| IDLE_CLOCK_LOW  | FIRST_EDGE  | RISING  | RISING  | Transmit on rising first edge   | Receive on rising first edge   |
|                 |             | RISING  | FALLING | Transmit on rising first edge   | Receive on falling second edge |
|                 |             | FALLING | RISING  | Transmit on falling second edge | Receive on rising first edge   |
|                 | SECOND_EDGE | RISING  | RISING  | Transmit on falling second edge | Receive on falling second edge |
|                 |             | RISING  | FALLING | Transmit on falling second edge | Receive on rising first edge   |
|                 |             | FALLING | RISING  | Transmit on rising first edge   | Receive on falling second edge |
| IDLE_CLOCK_HIGH | FIRST_EDGE  | RISING  | RISING  | Transmit on falling first edge  | Receive on falling first edge  |
|                 |             | RISING  | FALLING | Transmit on falling first edge  | Receive on rising second edge  |
|                 |             | FALLING | RISING  | Transmit on rising second edge  | Receive on falling first edge  |
|                 | SECOND_EDGE | RISING  | RISING  | Transmit on rising second edge  | Receive on rising second edge  |
|                 |             | RISING  | FALLING | Transmit on rising second edge  | Receive on falling first edge  |
|                 |             | FALLING | RISING  | Transmit on falling first edge  | Receive on rising second edge  |



**Notes:**

1. For CPOL, 0 and 1 are equivalent to IDLE\_CLOCK\_LOW and IDLE\_CLOCK\_HIGH, respectively.
2. For CPHA, 0 and 1 are equivalent to FIRST\_EDGE and SECOND\_EDGE, respectively.
3. For TX/RX\_EDGE, 0 and 1 are equivalent to RISING and FALLING, respectively.

**Figure 7.2. CPOL, CPHA, and TX/RX\_EDGE Settings versus Waveform Examples**

## 7.2.14. MSPI\_RX\_EDGE

The MSPI\_RX\_EDGE option selects the controller SPI data receiving clock edge. Refer to [Table 7.15](#) and [Figure 7.2](#) for the different combinations of CPOL, CPHA, and TX/RX\_EDGE settings and corresponding TX/RX operations and waveform examples, respectively.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.16. MSPI\_RX\_EDGE Option**

| Option Name  | Setting(s)                    | Description                                                                                                                                                                                                                                       |
|--------------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSPI_RX_EDGE | RISING (default) <sup>1</sup> | Set the controller SPI to receive data on the MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA. For example, if MSPI_CPOL = IDLE_CLOCK_LOW and MSPI_CPHA = FIRST_EDGE, the MCLKP edge is a rising leading (first) edge.                           |
|              | FALLING                       | Set the controller SPI to receive data on the inverted MCLKP edge. For example, if MSPI_CPOL = IDLE_CLOCK_LOW and MSPI_CPHA = FIRST_EDGE, the MCLKP edge is a rising leading edge. The inverted MCLKP edge is the falling trailing (second) edge. |

**Note:**

1. When MASTER\_SPI\_PORT = XSPI, you must use the default MSPI\_RX\_EDGE option setting, which is MSPI\_RX\_EDGE = RISING.

## 7.2.15. MSPI\_CPHA

The MSPI\_CPHA option selects the data receiving and transmitting edges in relation to the controller SPI clock phase.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.17. MSPI\_CPHA Option**

| Option Name | Setting(s)           | Description                                                                                   |
|-------------|----------------------|-----------------------------------------------------------------------------------------------|
| MSPI_CPHA   | FIRST_EDGE (default) | Set the controller SPI transceiver to transmit or receive data on the leading edge of MCLKP.  |
|             | SECOND_EDGE          | Set the controller SPI transceiver to transmit or receive data on the trailing edge of MCLKP. |

## 7.2.16. MSPI\_CPOL

The MSPI\_CPOL option selects the controller SPI clock polarity (inverted or non-inverted), which determines the clock idle state when the port is not active.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.18. MSPI\_CPOL Option**

| Option Name | Setting(s)               | Description                 |
|-------------|--------------------------|-----------------------------|
| MSPI_CPOL   | IDLE_CLOCK_LOW (default) | MCLKP stays low when idle.  |
|             | IDLE_CLOCK_HIGH          | MCLKP stays high when idle. |

### 7.2.17. SSPI\_TX\_EDGE

The SSPI\_TX\_EDGE option selects the target SPI data transmitting clock edge. Refer to [Table 7.15](#) and [Figure 7.2](#) for the different combinations of CPOL, CPHA, and TX/RX\_EDGE settings and corresponding TX/RX operations and waveform examples, respectively.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.19. SSPI\_TX\_EDGE Option**

| Option Name  | Setting(s)        | Description                                                                                                                                                                                                                                    |
|--------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSPI_TX_EDGE | FALLING (default) | Set the target SPI to transmit data on the inverted SCLKP edge. For example, if SSPI_CPOL = IDLE_CLOCK_LOW and SSPI_CPHA = FIRST_EDGE, the SCLKP edge is a rising leading edge. The inverted SCLKP edge is the falling trailing (second) edge. |
|              | RISING            | Set the target SPI to transmit data on the SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA. For example, if SSPI_CPOL = IDLE_CLOCK_LOW and SSPI_CPHA = FIRST_EDGE, the SCLKP edge is a rising leading (first) edge.                           |

### 7.2.18. SSPI\_RX\_EDGE

The SSPI\_RX\_EDGE option selects the target SPI data receiving clock edge. Refer to [Table 7.15](#) and [Figure 7.2](#) for the different combinations of CPOL, CPHA, and TX/RX\_EDGE settings and corresponding TX/RX operations and waveform examples, respectively.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.20. SSPI\_RX\_EDGE Option**

| Option Name  | Setting(s)       | Description                                                                                                                                                                                                                                   |
|--------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSPI_RX_EDGE | RISING (default) | Set the target SPI to receive data on the SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA. For example, if SSPI_CPOL = IDLE_CLOCK_LOW and SSPI_CPHA = FIRST_EDGE, the SCLKP edge is a rising leading (first) edge.                           |
|              | FALLING          | Set the target SPI to receive data on the inverted SCLKP edge. For example, if SSPI_CPOL = IDLE_CLOCK_LOW and SSPI_CPHA = FIRST_EDGE, the SCLKP edge is a rising leading edge. The inverted SCLKP edge is the falling trailing (second) edge. |

### 7.2.19. SSPI\_CPHA

The SSPI\_CPHA option selects the data receiving and transmitting edges in relation to the target SPI clock phase.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.21. SSPI\_CPHA Option**

| Option Name | Setting(s)           | Description                                                                               |
|-------------|----------------------|-------------------------------------------------------------------------------------------|
| SSPI_CPHA   | FIRST_EDGE (default) | Set the target SPI transceiver to transmit or receive data on the leading edge of SCLKP.  |
|             | SECOND_EDGE          | Set the target SPI transceiver to transmit or receive data on the trailing edge of SCLKP. |

## 7.2.20. SSPI\_CPOL

The SSPI\_CPOL option selects the target SPI clock polarity (inverted or non-inverted), which determines the clock idle state when the port is not active.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.22. SSPI\_CPOL Option**

| Option Name | Setting(s)               | Description                 |
|-------------|--------------------------|-----------------------------|
| SSPI_CPOL   | IDLE_CLOCK_LOW (default) | SCLKP stays low when idle.  |
|             | IDLE_CLOCK_HIGH          | SCLKP stays high when idle. |

## 7.2.21. SSPI\_SHIFT\_ORDER

The SSPI\_SHIFT\_ORDER option selects the target SPI transceiver shifting direction.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.23. SSPI\_SHIFT\_ORDER Option**

| Option Name      | Setting(s)          | Description                               |
|------------------|---------------------|-------------------------------------------|
| SSPI_SHIFT_ORDER | LSB_FIRST (default) | Least significant bit (LSB) shifts first. |
|                  | MSB_FIRST           | Most significant bit (MSB) shifts first.  |

## 7.2.22. SSPI\_DAISY\_CHAIN\_MODE

The SSPI\_DAISY\_CHAIN\_MODE option enables or disables the device as a downstream device in a daisy chain.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.24. SSPI\_DAISY\_CHAIN\_MODE Option**

| Option Name           | Setting(s)        | Description                                                                                                |
|-----------------------|-------------------|------------------------------------------------------------------------------------------------------------|
| SSPI_DAISY_CHAIN_MODE | DISABLE (default) | Disable the device as a downstream device in a daisy chain.                                                |
|                       | SCM               | Enable the device in serial configuration mode as a downstream device in a daisy chain.                    |
|                       | AUTO              | Automatically set the device as a downstream device in a daisy chain following the SLAVE_SPI_PORT setting. |

## 7.2.23. ERASE\_EBR\_ON\_REFRESH

The ERASE\_EBR\_ON\_REFRESH option enables or disables the erasure of EBR contents upon a refresh event triggered by PROGRAMN pin pulsing or REFRESH command execution.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.25. ERASE\_EBR\_ON\_REFRESH Option**

| Option Name          | Setting(s)        | Description                                                                                 |
|----------------------|-------------------|---------------------------------------------------------------------------------------------|
| ERASE_EBR_ON_REFRESH | DISABLE (default) | Disable the erasure of EBR contents upon PROGRAMN pin pulsing or REFRESH command execution. |
|                      | ENABLE            | Enable the erasure of EBR contents upon PROGRAMN pin pulsing or REFRESH command execution.  |

## 7.2.24. SIGNATURE\_CHECK

The SIGNATURE\_CHECK option controls the signature checking mode for the flash device before loading the bitstream.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.26. SIGNATURE\_CHECK Option**

| Option Name     | Setting(s)                      | Description                                   |
|-----------------|---------------------------------|-----------------------------------------------|
| SIGNATURE_CHECK | ENABLE_LSCC_SIGNATURE (default) | Enable signature checking for LSCC signature. |
|                 | DISABLE                         | Disable signature checking.                   |
|                 | ENABLE_SFDP_SIGNATURE           | Enable signature checking for SFDP signature. |

## 7.2.25. MSPI\_ADDRESS\_32BIT

The MSPI\_ADDRESS\_32BIT option sets the address format for the SPI flash memory.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.27. MSPI\_ADDRESS\_32BIT Option**

| Option Name        | Setting(s)        | Description                        |
|--------------------|-------------------|------------------------------------|
| MSPI_ADDRESS_32BIT | DISABLE (default) | Use 24-bit controller SPI address. |
|                    | ENABLE            | Use 32-bit controller SPI address. |

## 7.2.26. MSPI\_COMMAND\_32BIT

The MSPI\_COMMAND\_32BIT option sets the command format for SPI flash memory.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.28. MSPI\_COMMAND\_32BIT Option**

| Option Name        | Setting(s)        | Description                         |
|--------------------|-------------------|-------------------------------------|
| MSPI_COMMAND_32BIT | DISABLE (default) | Use 24-bit controller SPI commands. |
|                    | ENABLE            | Use 32-bit controller SPI commands. |

## 7.2.27. SSPI\_IDLE\_TIMER

The SSPI\_IDLE\_TIMER option sets the target idle timer to prevent system lock when performing segmented bitstream burst.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.29. SSPI\_IDLE\_TIMER Option**

| Option Name     | Setting(s)                                                                            | Description                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSPI_IDLE_TIMER | DEFAULT (default)                                                                     | Disable the timer.                                                                                                                                                                                                                                                                                                                                             |
|                 | 200S, 100S, 50S, 25S, 10S, 5S, 1S, 750MS, 500MS, 250MS, 100MS, 75MS, 50MS, 25MS, 10MS | Set timer value to the selected value. The unit of the timer value is indicated in the setting name, for example, S for seconds and MS for milliseconds. If a bitstream is sent through the SSPI or JTAG interface and the idle time on the port exceeds the timeout, the configuration logic stops and indicates SSPI Timeout in Status Register 1 (STATUS1). |

## 7.2.28. MSPI\_PREAMBLE\_DETECTION\_TIMER

The MSPI\_PREAMBLE\_DETECTION\_TIMER option sets the timer for preamble detection.

**Note:** This setting is stored in the device non-volatile memory (OTP).

**Table 7.30. MSPI\_PREAMBLE\_DETECTION\_TIMER Option**

| Option Name                   | Setting(s)                                 | Description                                                                                                                                                                                                                                                                                                        |
|-------------------------------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSPI_PREAMBLE_DETECTION_TIMER | 200MS (default)                            | Default timer value. The unit of the timer value is indicated in the setting name, for example, <i>MS</i> for milliseconds. If no preamble is detected before timeout, the configuration logic stops and indicates preamble timeout in Status Register 0 (STATUS0).                                                |
|                               | 100MS, 50MS, 40MS, 20MS, 1MS, 500US, 100US | Set timer value to the selected value. The unit of the timer value is indicated in the setting name for example <i>MS</i> for milliseconds and <i>US</i> for microseconds. If no preamble is detected before timeout, the configuration logic stops and indicates preamble timeout in Status Register 0 (STATUS0). |

## 7.2.29. BACKGROUND\_RECONFIG

The BACKGROUND\_RECONFIG option controls the behavior regarding transparent access mode.

**Table 7.31. BACKGROUND\_RECONFIG Option**

| Option Name         | Setting(s)    | Description                                                                                                                                                                                         |
|---------------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BACKGROUND_RECONFIG | OFF (default) | Prevent configuration memory and INIT access after executing PROG_ENABLE in user function mode. All other commands denoted with PROG_QUALIFY = Y in <a href="#">Table 6.12</a> are still available. |
|                     | ON            | Cause the device to go into transparent access mode with configuration memory and INIT access after executing PROG_ENABLE in user function mode.                                                    |
|                     | SRAM_ONLY     | Cause the device to go into transparent access mode with configuration memory only access after executing PROG_ENABLE in user function mode.                                                        |

## 7.2.30. DAISY\_CHAIN

The DAISY\_CHAIN option selects the sysCONFIG daisy chain mode.

**Table 7.32. DAISY\_CHAIN Option**

| Option Name | Setting(s)        | Description                                            |
|-------------|-------------------|--------------------------------------------------------|
| DAISY_CHAIN | DISABLE (default) | Disable the sysCONFIG daisy chain.                     |
|             | BYPASS            | Enable the sysCONFIG daisy chain in bypass mode.       |
|             | FLOW_THROUGH      | Enable the sysCONFIG daisy chain in flow-through mode. |

### 7.2.31. DAISY\_CHAIN\_WAIT\_DONE

The DAISY\_CHAIN\_WAIT\_DONE option selects the wake-up mode for the device in a sysCONFIG daisy chain.

**Table 7.33. DAISY\_CHAIN\_WAIT\_DONE Option**

| Option Name           | Setting(s)        | Description                                                                                                                       |
|-----------------------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| DAISY_CHAIN_WAIT_DONE | DISABLE (default) | Device enters user function mode immediately after successful configuration without waiting for the external DONE pin to go high. |
|                       | ENABLE            | Device waits for the external DONE pin to go high before entering user function mode.                                             |

### 7.2.32. TRANSFR

The TRANSFR option enables or disables the TRANSFR feature for latching and freezing I/O pins during reconfiguration.

**Table 7.34. TRANSFR Option**

| Option Name | Setting(s)    | Description      |
|-------------|---------------|------------------|
| TRANSFR     | OFF (default) | Disable TRANSFR. |
|             | ON            | Enable TRANSFR.  |

### 7.2.33. UserCode Format

The UserCode Format option selects the format for the data field used to assign a value to UserCode.

**Table 7.35. UserCode Format Option**

| Option Name     | Setting(s)       | Description                                                                                                                                                                         |
|-----------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UserCode Format | Binary (default) | Set user code using 32 1 or 0 characters.                                                                                                                                           |
|                 | Hex              | Set user code using eight hexadecimal digits (0–9, A–F).                                                                                                                            |
|                 | ASCII            | Set user code using up to four ASCII characters.                                                                                                                                    |
|                 | Auto             | Software automatically creates user code. The upper 16 bits constitute the Unique ID and the lower 16 bits are sequentially increased automatically for every bitstream generation. |

### 7.2.34. UserCode

The FPGA device contains a 32-bit register for storing a user-defined value. This register can be initialized with any 32-bit value specified through UserCode.

**Table 7.36. UserCode Option**

| Option Name | Setting(s)      | Description                                                                                                                                                                                                                                                                   |
|-------------|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UserCode    | 32'b0 (default) | 32-bit user-defined value to be stored in a 32-bit register in the FPGA device. Suggested uses include the configuration data version number, a manufacturing ID code, date of assembly, or the JEDEC file checksum. The data format is set using the UserCode Format option. |

### 7.2.35. MultiBoot Offset

The FPGA device contains a 48-bit register for storing the boot address (for multi-boot operations) in external or internal flash memory. This register can be initialized with any 48-bit value specified through MultiBoot Offset.

**Table 7.37. MultiBoot Offset Option**

| Option Name      | Setting(s)      | Description                                                                                                |
|------------------|-----------------|------------------------------------------------------------------------------------------------------------|
| MultiBoot Offset | 48'b0 (default) | 48-bit address for external or internal flash memory to be stored in a 48-bit register in the FPGA device. |

### 7.2.36. BitstreamRevision Format

The BitstreamRevision Format option selects the format for the data field used to assign a value to BitstreamRevision.

**Table 7.38. BitstreamRevision Format Option**

| Option Name              | Setting(s)       | Description                                                       |
|--------------------------|------------------|-------------------------------------------------------------------|
| BitstreamRevision Format | Binary (default) | Set bitstream revision using 32 1 or 0 characters.                |
|                          | Hex              | Set bitstream revision using eight hexadecimal digits (0–9, A–F). |
|                          | ASCII            | Set bitstream revision using up to four ASCII characters.         |
|                          | Timestamp        | Timestamp automatically sets the bitstream revision.              |

### 7.2.37. BitstreamRevision

The FPGA device contains a 32-bit register for storing the bitstream revision information. This register can be initialized with any 32-bit value specified through BitstreamRevision.

**Table 7.39. BitstreamRevision Option**

| Option            | Value           | Description                                                                                                                                                   |
|-------------------|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BitstreamRevision | 32'b0 (default) | 32-bit bitstream revision information to be stored in a 32-bit register in the FPGA device. The data format is set using the BitstreamRevision Format option. |

## 8. Daisy Chaining

Typically, there is one configuration bitstream per FPGA in a system. Today's systems often have several FPGA devices. If all the FPGAs in the application use the same device and bitstream, only a single bitstream is required. Using a ganged configuration loads multiple, similar FPGAs with the same bitstream at the same time.

However, to save PCB space and use external or internal storage devices more efficiently, several different FPGA bitstreams from various devices and designs can share a single configuration mechanism by using a daisy chain method. The Nexus 2 device supports flow-through, bypass, and SCM modes when the leading device is in the controller SPI serial, dual, or quad mode. There is no daisy chaining support when the leading device is in xSPI mode.

The Nexus 2 device supports flow-through mode, bypass mode, and SCM mode with external host. This section provides guidance on enabling configuration daisy chaining for Nexus 2 devices using the Lattice Radiant Device Constraint Editor. Each supported daisy-chain mode requires a specific combination of sysCONFIG settings. Depending on the number of devices in the chain and the selected mode, two to three different bitstreams must be generated using different constraint settings. These bitstreams must then be merged into a single daisy-chain configuration file (for example, .mcs file) using the Radiant Deployment Tool.

The following is the general daisy-chain configuration flow:

1. Determine the desired daisy-chain mode (flow-through mode, bypass mode, or SCM mode with external host).
2. Identify the role of each device in the chain as follows:
  - First FPGA
  - Intermediate FPGA (between first and last)
  - Last FPGA
3. For each role, configure the required sysCONFIG options for the device through the Device Constraint Editor.
4. Generate a bitstream for each role.
  - For a chain with intermediate devices, generate three different bitstreams: first, intermediate, and last.
  - For a two-device chain, generate two different bitstreams: first and last.
5. Use the Radiant Deployment Tool to convert and merge the generated bitstreams into a single daisy-chain configuration file.

### 8.1. Flow-Through Mode

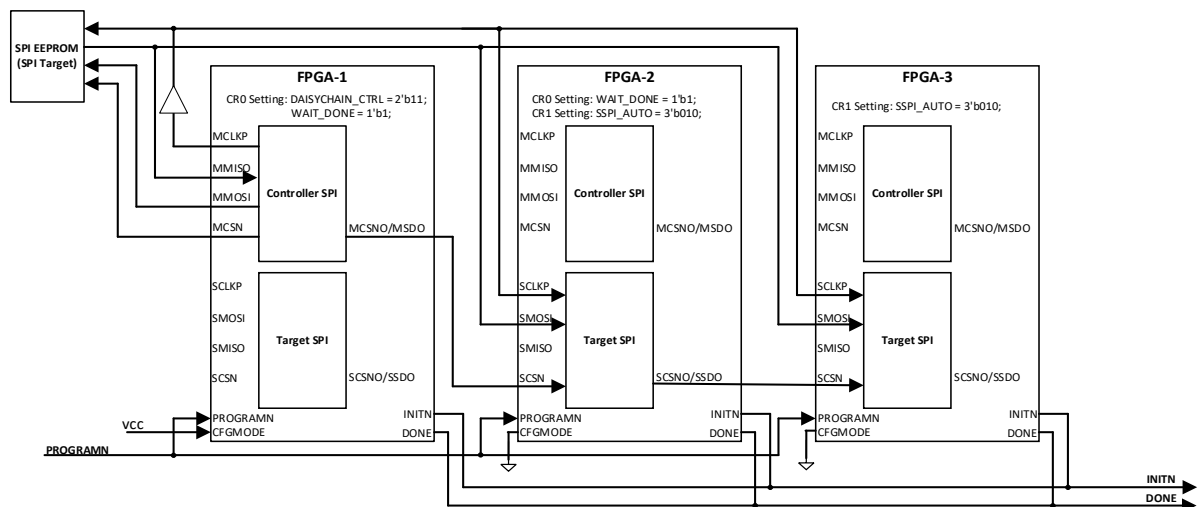
To configure flow-through mode daisy chaining, the data lines of all the devices are connected to the same data line, and the SCSN (SPI Chip-Select input) of a downstream FPGA device is connected to the MCSNO/MSDO pin (SPI Chip-Select output) of the FPGA device before it. In this daisy chain configuration, the devices are configured sequentially by placing the completed device in one of the flow-through states (Config Flow-Through or End Flow-Through state). This sets the MCSNO/MSDO pin to low, selecting the next device in the chain. Then the next device (target device) sets SCSNO/SSDO pin low to the last device.

In flow-through mode daisy chaining, the lead device is set to a controller SPI configuration mode while all target devices in the chain are set to the auto SSPI mode. Configuration data flows directly through the controller and target SPI ports to downstream devices. Each device fully participates in the SPI-based daisy chain.

An example of the Lattice Nexus 2 devices in a configuration daisy chain with flow-through option is shown in [Figure 8.1](#). Without the buffer on the MCLKP line, the maximum supported MCLK frequency is 28.6 MHz. Adding this optional buffer can increase the maximum frequency, subject to limiting factors such as buffer propagation delay and the number of daisy-chain loads. Note that downstream devices in daisy chain mode only operate in x1 SPI mode.

### Table 8.1. Required sysCONFIG Settings for Flow-Through Mode

| FPGA Role         | sysCONFIG Options                                         |                                            |                                              |
|-------------------|-----------------------------------------------------------|--------------------------------------------|----------------------------------------------|
|                   | DAISY_CHAIN                                               | DAISY_CHAIN_WAIT_DONE                      | SSPI_DAI SY_CHAIN_MODE                       |
| First FPGA        | FLOW_THROUGH<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b11) | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | DISABLE<br>(CR1 Setting: SSPI_AUTO = 3'b000) |
| Intermediate FPGA | DISABLE<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b00)      | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | AUTO<br>(CR1 Setting: SSPI_AUTO = 3'b010)    |
| Last FPGA         | DISABLE<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b00)      | DISABLE<br>(CR0 Setting: WAIT_DONE = 1'b0) | AUTO<br>(CR1 Setting: SSPI_AUTO = 3'b010)    |



**Figure 8.1. Lattice Nexus 2 in Configuration Daisy Chain in Flow-Through Mode**

## 8.2. Bypass Mode

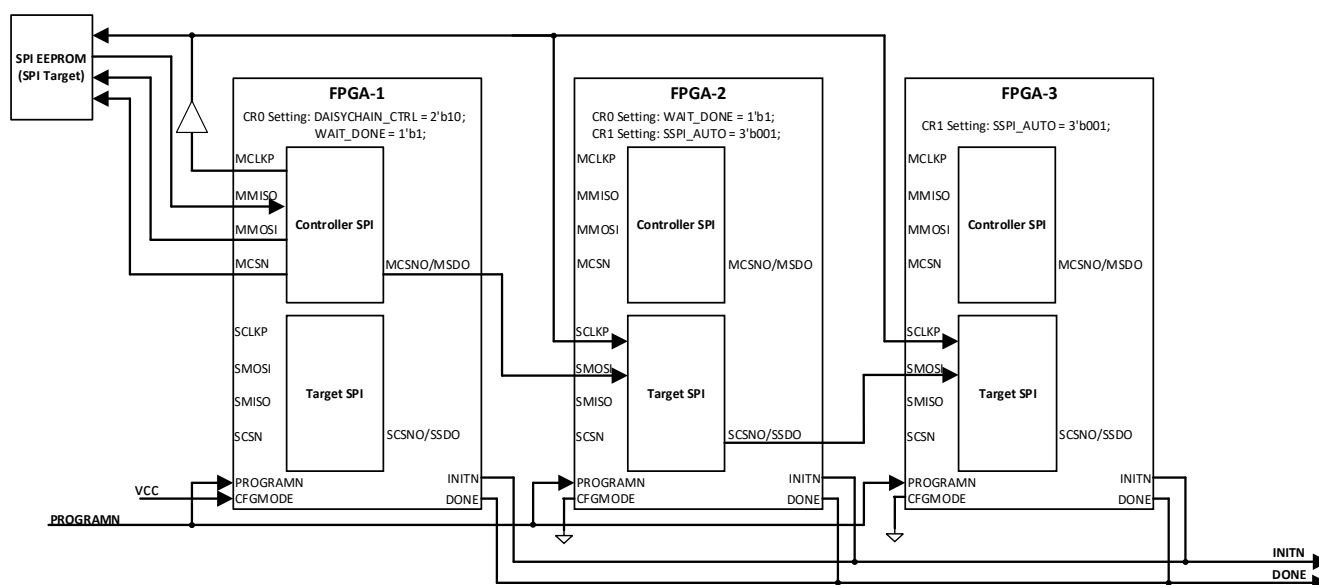
In bypass mode configuration, the upstream device is in controller SPI boot mode and reads data from flash. Data is passed to the downstream device through the MCSNO/MSDO pin to the SI pin of the downstream device through the target SPI port. Downstream devices that are not the last device in the chain forward data through the SCSNO/SSDO pin to the SI pin of the downstream device.

An example of the Lattice Nexus 2 devices in a configuration daisy chain with bypass option is shown in [Figure 8.2](#). Without the buffer on the MCLKP line, the maximum supported MCLK frequency is 28.6 MHz. Adding this optional buffer can increase the maximum frequency, subject to limiting factors such as buffer propagation delay and the number of daisy-chain loads. Note that downstream devices in daisy chain mode only operate in x1 SPI mode.

[Table 8.2](#) shows the required sysCONFIG settings for the bypass mode.

**Table 8.2. Required sysCONFIG Settings for Bypass Mode**

| FPGA Role         | sysCONFIG Options                                    |                                            |                                              |
|-------------------|------------------------------------------------------|--------------------------------------------|----------------------------------------------|
|                   | DAISY_CHAIN                                          | DAISY_CHAIN_WAIT_DONE                      | SSPI_DAISY_CHAIN_MODE                        |
| First FPGA        | BYPASS<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b10)  | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | DISABLE<br>(CR1 Setting: SSPI_AUTO = 3'b000) |
| Intermediate FPGA | DISABLE<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b00) | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | SCM<br>(CR1 Setting: SSPI_AUTO = 3'b001)     |
| Last FPGA         | DISABLE<br>(CR0 Setting:<br>DAISYCHAIN_CTRL = 2'b00) | DISABLE<br>(CR0 Setting: WAIT_DONE = 1'b0) | SCM<br>(CR1 Setting: SSPI_AUTO = 3'b001)     |



**Figure 8.2. Lattice Nexus 2 in Configuration Daisy Chain in Bypass Mode**

### 8.3. SCM Mode with External Host

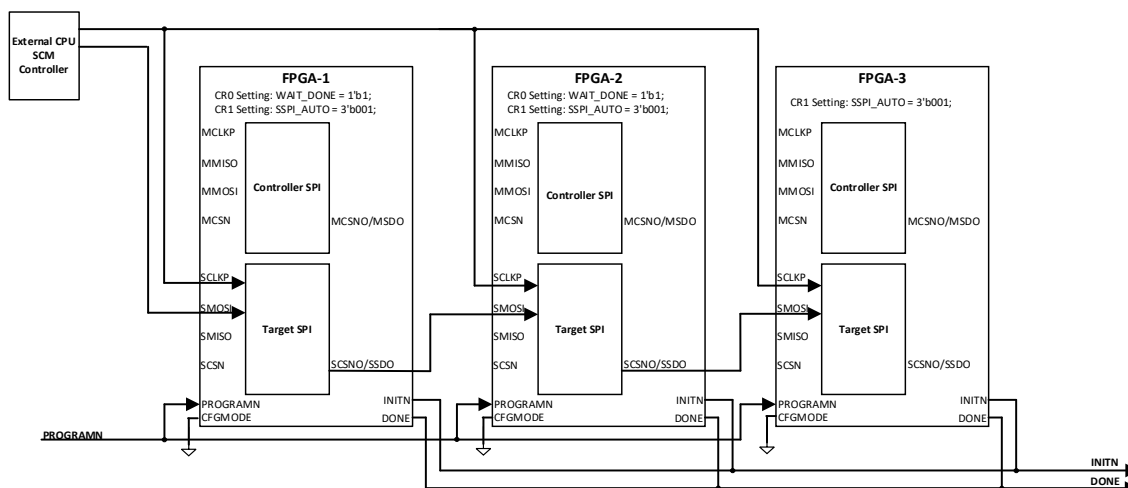
SCM mode can be used to configure multiple devices in a chain with an external host. Figure 8.3 shows the connectivity of a system using SCM mode with an external host.

In SCM mode with external host daisy chaining, all devices are configured using the legacy serial configuration mode. The external host controls the entire configuration sequence.

Table 8.3 shows the required sysCONFIG settings for the SCM mode with external host.

**Table 8.3. Required sysCONFIG Settings for SCM Mode with External Host**

| FPGA Role         | sysCONFIG Options                                 |                                            |                                          |
|-------------------|---------------------------------------------------|--------------------------------------------|------------------------------------------|
|                   | DAISY_CHAIN                                       | DAISY_CHAIN_WAIT_DONE                      | SSPI_DAISSY_CHAIN_MODE                   |
| First FPGA        | DISABLE<br>(CR0 Setting: DAISYCHAIN_CTRL = 2'b00) | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | SCM<br>(CR1 Setting: SSPI_AUTO = 3'b001) |
| Intermediate FPGA | DISABLE<br>(CR0 Setting: DAISYCHAIN_CTRL = 2'b00) | ENABLE<br>(CR0 Setting: WAIT_DONE = 1'b1)  | SCM<br>(CR1 Setting: SSPI_AUTO = 3'b001) |
| Last FPGA         | DISABLE<br>(CR0 Setting: DAISYCHAIN_CTRL = 2'b00) | DISABLE<br>(CR0 Setting: WAIT_DONE = 1'b0) | SCM<br>(CR1 Setting: SSPI_AUTO = 3'b001) |

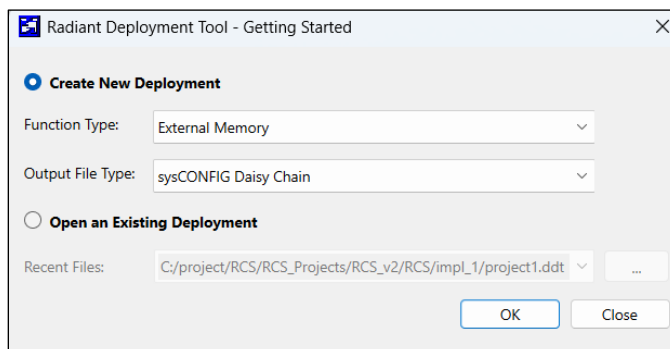


**Figure 8.3. Lattice Nexus 2 in Configuration Daisy Chain in SCM Mode with External Host**

## 8.4. Merging Daisy-Chain Bitstreams

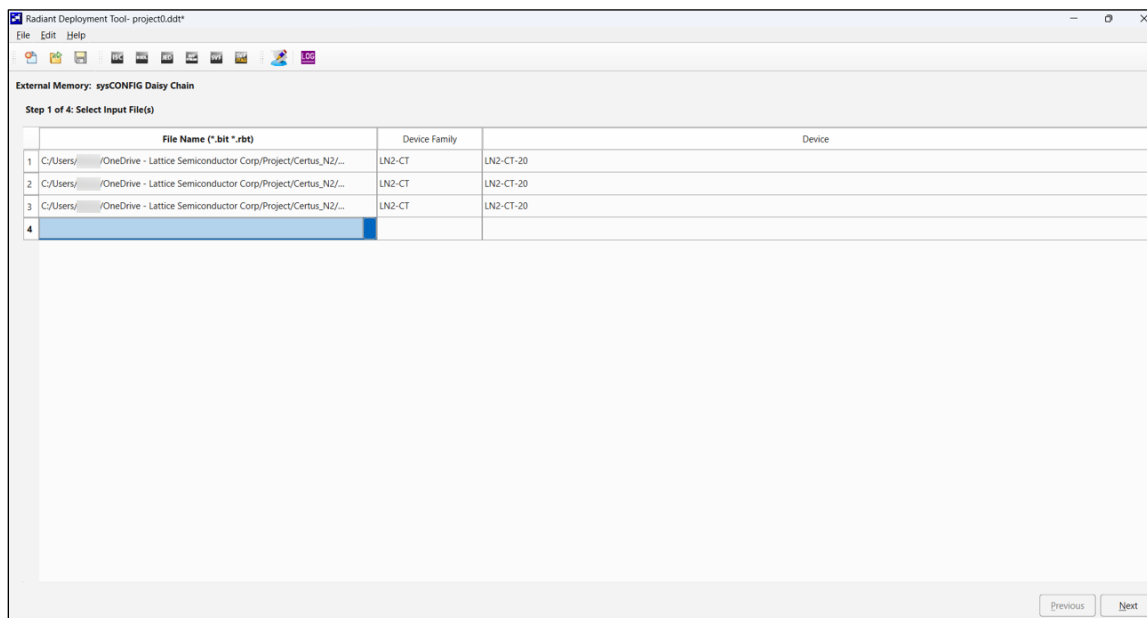
To merge the required generated bitstreams for a daisy chain into a single daisy chain configuration file, perform the following:

1. In the Lattice Radiant Programmer, select **Tools > Deployment Tool** to launch the Lattice Radiant Deployment Tool. The **Getting Started** window appears.



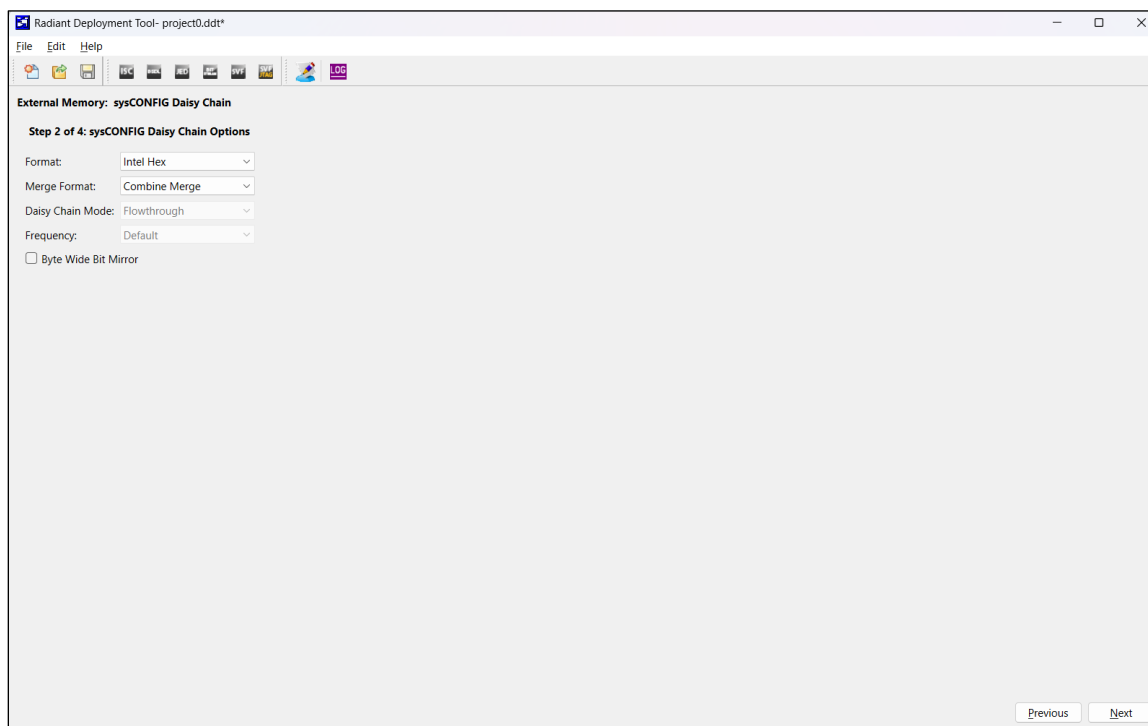
**Figure 8.4. Radiant Deployment Tool – Getting Started (sysCONFIG Daisy Chain)**

2. Select the following options:
  - Function Type: External Memory
  - Output File Type: sysCONFIG Daisy Chain
3. Click **OK**.
4. In the Radiant Deployment Tool *Step 1 of 4: Select Input File(s)*:
  - a. Add two or more bitstream files depending on the number of devices in the daisy chain. To add a file, double-click the empty cell in the **File Name** column, and then navigate to and select the file.
  - b. Click **Next**.



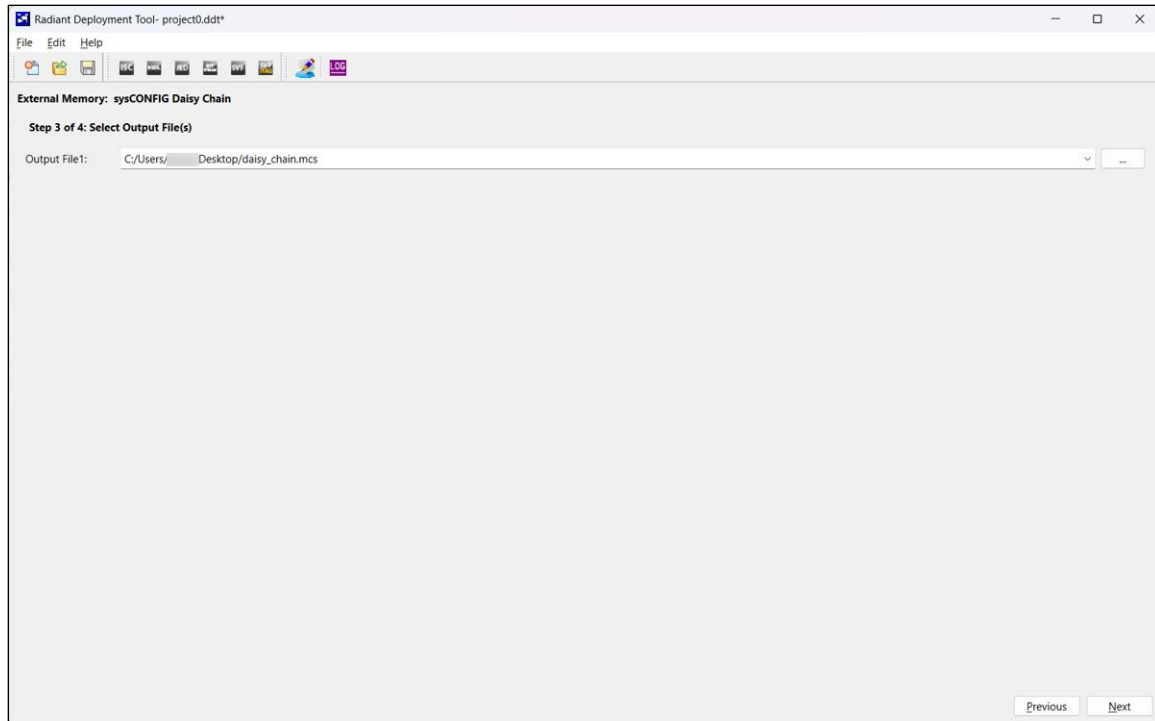
**Figure 8.5. Radiant Deployment Tool – Step 1 of 4: Select Input File(s) (sysCONFIG Daisy Chain)**

5. In the Radiant Deployment Tool *Step 2 of 4: sysCONFIG Daisy Chain Options*:
  - a. Select the following options:
    - Format: Intel Hex
    - Merge Format: Combine Merge
  - b. Click **Next**.



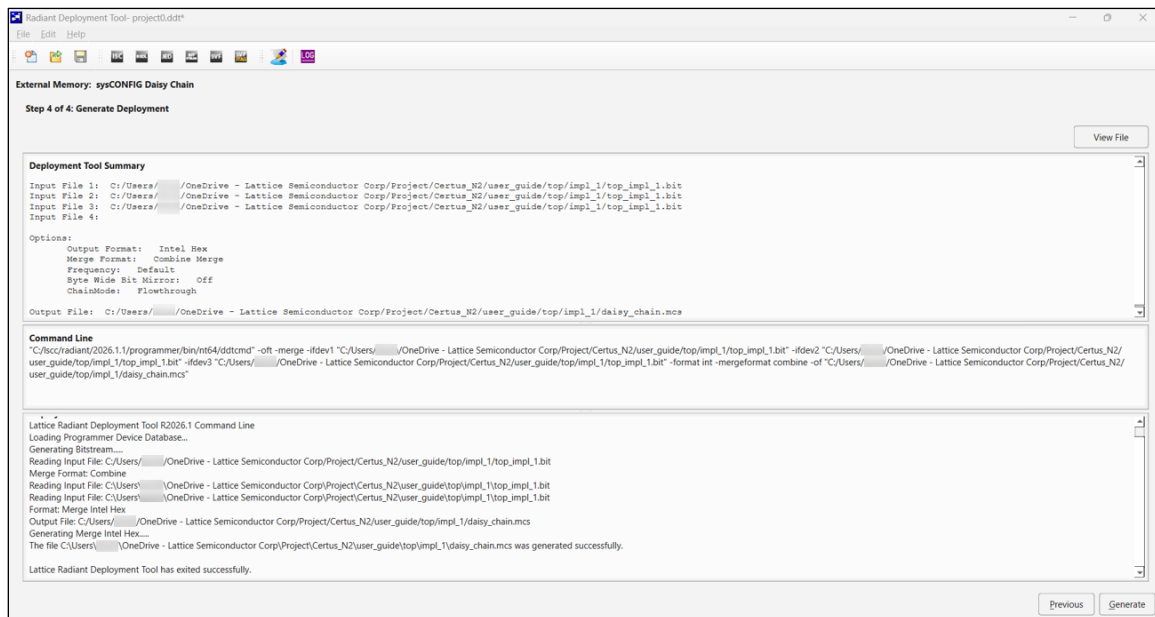
**Figure 8.6. Radiant Deployment Tool – Step 2 of 4: sysCONFIG Daisy Chain Options**

6. In the Radiant Deployment Tool *Step 3 of 4: Select Output File(s)*:
  - c. Click the ... (Browse) button associated with the **Output File 1** field.
  - d. Choose a destination directory.
  - e. Enter a filename (for example, *daisy\_chain.mcs*).
  - f. Ensure the file type is set to MCS (\*.mcs).
  - g. Click **Save**.
  - h. Click **Next**.



**Figure 8.7. Radiant Deployment Tool – Step 3 of 4: Select Output File(s) (sysCONFIG Daisy Chain)**

7. In the Radiant Deployment Tool *Step 4 of 4: Generate Deployment*, click **Generate**.



**Figure 8.8. Radiant Deployment Tool – Step 4 of 4: Generate Deployment (sysCONFIG Daisy Chain)**

## Appendix A. Nexus 2 Target SPI Programming Guide

The target SPI port of the Lattice Nexus 2 device can be used for device configuration. If these pins are not used by user logic, they are tri-stated with a weak pull-up. The target SPI port must be enabled in order to support device configuration from an external host or download cable using SPI protocol. This is done by setting the SLAVE\_SPI\_PORT preference to ENABLE in the bitstream through the Lattice Radiant Device Constraint Editor. Target SPI mode supports single device configuration.

The Lattice Radiant Programmer supports the target SPI programming mode as one of the device access options. Selecting this option allows you to perform device erase, program, verify, readback, refresh, and more. The connections of target SPI pins to the Lattice programming cable are:

- TDI -> SMOSI
- ISPEN -> SN
- TCK -> SCLKP
- TDO -> SMISO

The target SPI chip-select pin (SCSN) is held low during the command sequences. The Lattice Deployment Tool software can generate an SVF file from a bitstream file (.bit) to show the details of the command sequences for target SPI programming mode.

The *Program, Erase and Verify* flow in Radiant Programmer from any target configuration port only targets the FPGA SRAM array, which does not handle the hard IP and EBR initialization. *Fast Program* option is preferred for full device configuration.

## Appendix B. Nexus 2 Device Information

### B.1. Device ID

**Table B.1. Lattice Nexus 2 Device ID**

| Device Family Code | Product Name                  | Logic Capacity          | 32-bit IDCODE |
|--------------------|-------------------------------|-------------------------|---------------|
| LN2-CT             | LN2-CT-20                     | 220k System Logic Cells | 0x210A2043    |
| LN2-CT             | LN2-CT-20ES                   | 220k System Logic Cells | 0x010A2043    |
| LN2-CT             | LN2-CT-20ES1                  | 220k System Logic Cells | 0xA10A2043    |
| LN2-MH             | LN2-MH-20                     | 220k System Logic Cells | 0x150A2043    |
| LN2-MH             | LN2-MH-20ES1,<br>LN2-MH-70ES1 | 220k System Logic Cells | 0xA90A4043    |

### B.2. Configuration Memory

**Table B.2. Lattice Nexus 2 CRAM Frame Sizes**

| Device Family Code | Product Name | Number of CRAM Levels | Number of CRAM Zones | CRAM Frame Length, Level 0 (bytes) | CRAM Frame Length, Level 1 (bytes) |
|--------------------|--------------|-----------------------|----------------------|------------------------------------|------------------------------------|
| LN2-CT             | LN2-CT-20    | 2                     | 5                    | 62                                 | 70                                 |

## Appendix C. Modifying User OTP Settings

The Nexus 2 device user OTP memory (also known as the feature row) contains design-specific information for Nexus 2 device operation and behaviors. Bits of the user OTP settings are one-time programmable; once a bit is set to 1, it cannot be cleared to 0. All bits default to 0 in an unprogrammed device.

### C.1. Radiant Programmer Support for OTP Memory Access

The Lattice Radiant Programmer provides full support for OTP memory access. In the Radiant Programmer, double-click **Operation** to open the Device Properties window. In the Device Properties window, select **Non Volatile Configuration Memory** for *Target Memory* as shown in Figure C.1. The Radiant Programmer also provides a pseudo programming option, which programs data into shadow registers instead of non-volatile memory. The contents of the shadow registers are retained even after PROGRAMN pulsing or REFRESH command execution. The shadow registers are updated with the corresponding non-volatile memory data only after a device power cycle.

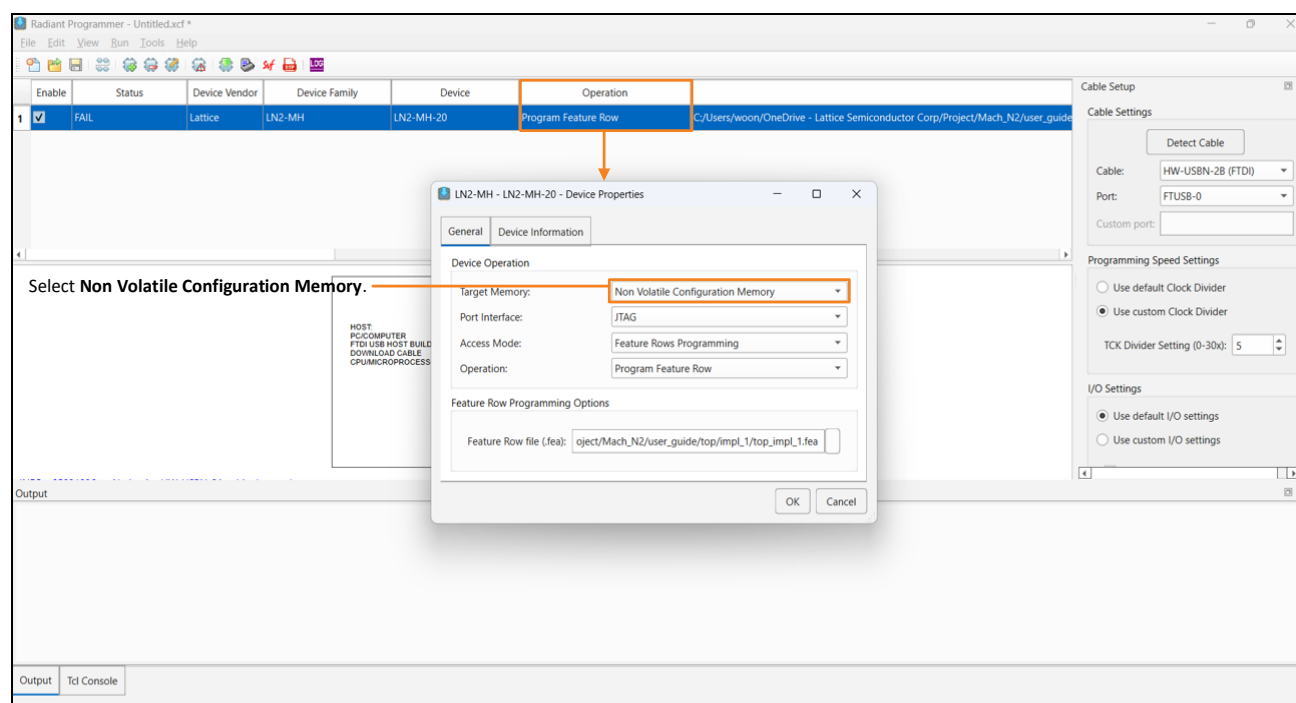
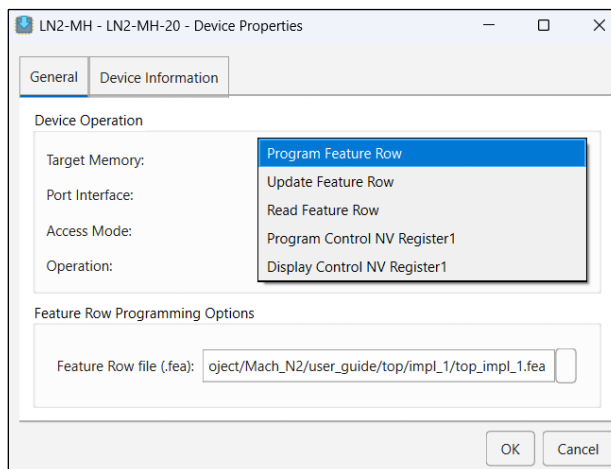


Figure C.1. Non-volatile Configuration Memory Access from Radiant Programmer

## C.2. Feature Rows Programming

After selecting **Non Volatile Configuration Memory** for *Target Memory*, select **Feature Rows Programming** for *Access Mode*. Feature rows programming operations are program/update/read feature row and program/display non-volatile control register 1 as shown in [Figure C.2](#).



**Figure C.2. Feature Rows Programming Operations**

### C.2.1. Feature Row Items

[Table C.1](#) lists the feature row items.

**Table C.1. Feature Row Items**

| Item                            | Default | Description                                                                                                                                                        |
|---------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Boot_SEL                        | 0       | Boot mode select<br>0 – Dual boot<br>1 – Single boot                                                                                                               |
| Cust_IDCODE_EN                  | 0       | Customer IDCODE enable<br>0 – OFF<br>1 – ON                                                                                                                        |
| MSPI_Reset_EN                   | 0       | Enable HW reset signal to flash device                                                                                                                             |
| tI[1:0]                         | 0       | Extra delay cycles for MCSN idle time                                                                                                                              |
| tT[2:0]                         | 0       | Extra delay cycles from MCLKP to MCSN high                                                                                                                         |
| tL[2:0]                         | 0       | Extra delay cycles from MCSN low to MCLKP                                                                                                                          |
| Customer_UID_msb[7:0]           | 0       | MSB of the unique ID code<br>UNIQUE_ID[63:0] = {Customer_UID_msb[7:0],<br>Manufacture_Electronic_Signature[55:0]}                                                  |
| Custom ID Code[31:0]            | 0       | Customer IDCODE                                                                                                                                                    |
| MSPI Signature Timer Count[2:0] | 0       | Signature counter timeout value select<br>000 – 200ms<br>001 – 100ms<br>010 – 50ms<br>011 – 40ms<br>100 – 20ms<br>101 – 1ms<br>110 – 500us<br>111 – 100us          |
| MSPI_CPHA                       | 0       | MSPI clock format select<br>0 – Sampling of data occurs at the leading (first) edge of MCLKP<br>1 – Sampling of data occurs at the trailing (second) edge of MCLKP |

| Item                     | Default | Description                                                                                                                                                                                                                                                |
|--------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MSPI_CPOL                | 0       | MSPI clock polarity select<br>0 – Active-high clocks selected. In idle state, MCLKP is low.<br>1 – Active-low clocks selected. In idle state, MCLKP is high.                                                                                               |
| MSPI_TX_Edge             | 0       | MSPI transmit data edge select<br>0 – Transmit data on MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA<br>1 – Transmit data on inverted MCLKP edge                                                                                                        |
| MSPI_RX_Edge             | 0       | MSPI receive data edge select<br>0 – Receive data on MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA<br>1 – Receive data on inverted MCLKP edge                                                                                                           |
| SSPI_CPHA                | 0       | SSPI clock format select<br>0 – Sampling of data occurs at the leading (first) edge of SCLKP<br>1 – Sampling of data occurs at the trailing (second) edge of SCLKP                                                                                         |
| SSPI_CPOL                | 0       | SSPI clock polarity select<br>0 – Active-high clocks selected. In idle state, SCLKP is low.<br>1 – Active-low clocks selected. In idle state, SCLKP is high.                                                                                               |
| SSPI_TX_Edge             | 0       | SSPI transmit data edge select<br>0 – Transmit data on SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA<br>1 – Transmit data on inverted SCLKP edge                                                                                                        |
| SSPI_RX_Edge             | 0       | SSPI receive data edge select<br>0 - Receive data on SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA<br>1 - Receive data on inverted SCLKP edge                                                                                                           |
| SSPI_LSBF                | 0       | SSPI shifting direction select<br>0 – MSB first shifting for transmitting and receiving data<br>1 – LSB first shifting for transmitting and receiving data                                                                                                 |
| SSPI Auto[2:0]           | 0       | Enable SSPI auto function for downstream daisy chain<br>000 – SPI auto/SCM mode disabled<br>001 – SCM mode<br>010 – SPI auto x1 mode<br>011 – SPI auto x2 mode<br>100 – SPI auto x4 mode<br>101 – SPI auto DDR x8 mode<br>110 – Reserved<br>111 – Reserved |
| EBR Erase Disable        | 0       | Disable the erase of EBR contents on PROGRAMN or refresh                                                                                                                                                                                                   |
| SFDP Enable              | 0       | Enable SFDP signature check for flash devices supporting SFDP                                                                                                                                                                                              |
| Signature Disable        | 0       | Disable flash signature check at power-up                                                                                                                                                                                                                  |
| Signature Infinite Retry | 0       | Do not timeout on signature check                                                                                                                                                                                                                          |
| 32-bit MSPI Address      | 0       | Enable 32-bit MSPI address for MSPI boot                                                                                                                                                                                                                   |
| 32-bit MSPI Command      | 0       | Enable 32-bit MSPI commands for MSPI boot                                                                                                                                                                                                                  |
| Disable IO glitch Filter | 0       | Disable the I/O glitch filter                                                                                                                                                                                                                              |

| Item                                  | Default | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSPI/JTAG Idle Timer count value[3:0] | 0       | <p>Target idle timer count value</p> <p>0000 – Disabled (default); bitstream sent in one continuous stream</p> <p>0001 – 200s</p> <p>0010 – 100s</p> <p>0011 – 50s</p> <p>0100 – 25s</p> <p>0101 – 10s</p> <p>0110 – 5s</p> <p>0111 – 1s</p> <p>1000 – 750ms</p> <p>1001 – 500ms</p> <p>1010 – 250ms</p> <p>1011 – 100ms</p> <p>1100 – 75ms</p> <p>1101 – 50ms</p> <p>1110 – 25ms</p> <p>1111 – 10ms</p> |
| MSPI Preamble Timer count value[2:0]  | 0       | <p>Controller preamble timer count value</p> <p>000 – 200ms</p> <p>001 – 100ms</p> <p>010 – 50ms</p> <p>011 – 40ms</p> <p>100 – 20ms</p> <p>101 – 1ms</p> <p>110 – 500us</p> <p>111 – 100us</p>                                                                                                                                                                                                          |
| Boot_Offset[47:0]                     | 0       | Starting boot address in SPI flash for Controller SPI mode                                                                                                                                                                                                                                                                                                                                               |

## C.2.2. Program Feature Row

To program the feature row, select **Program Feature Row** for *Operation* followed by the feature row file (.fea) generated by the Radiant software based on the user options in the Global page of the Device Constraint Editor. Next, click **OK** followed by the **Program Device** button as shown in Figure C.3.

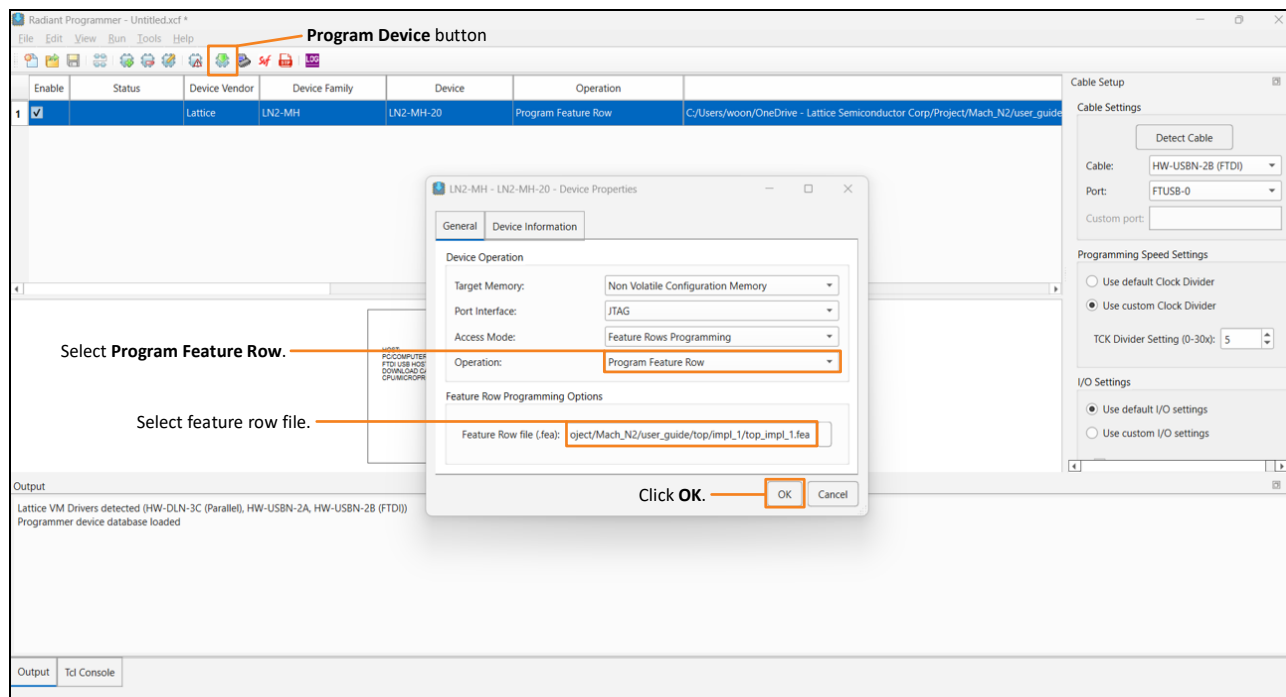


Figure C.3. Programming Feature Row

### C.2.3. Update Feature Row

To update the feature row, select **Update Feature Row** for *Operation* and then click **OK**. Next, click the **Program Device** button. The Feature Row window appears. This window allows feature row items to be modified by setting the **Chip Value** of the associated item and then clicking **Program** as shown in Figure C.4.

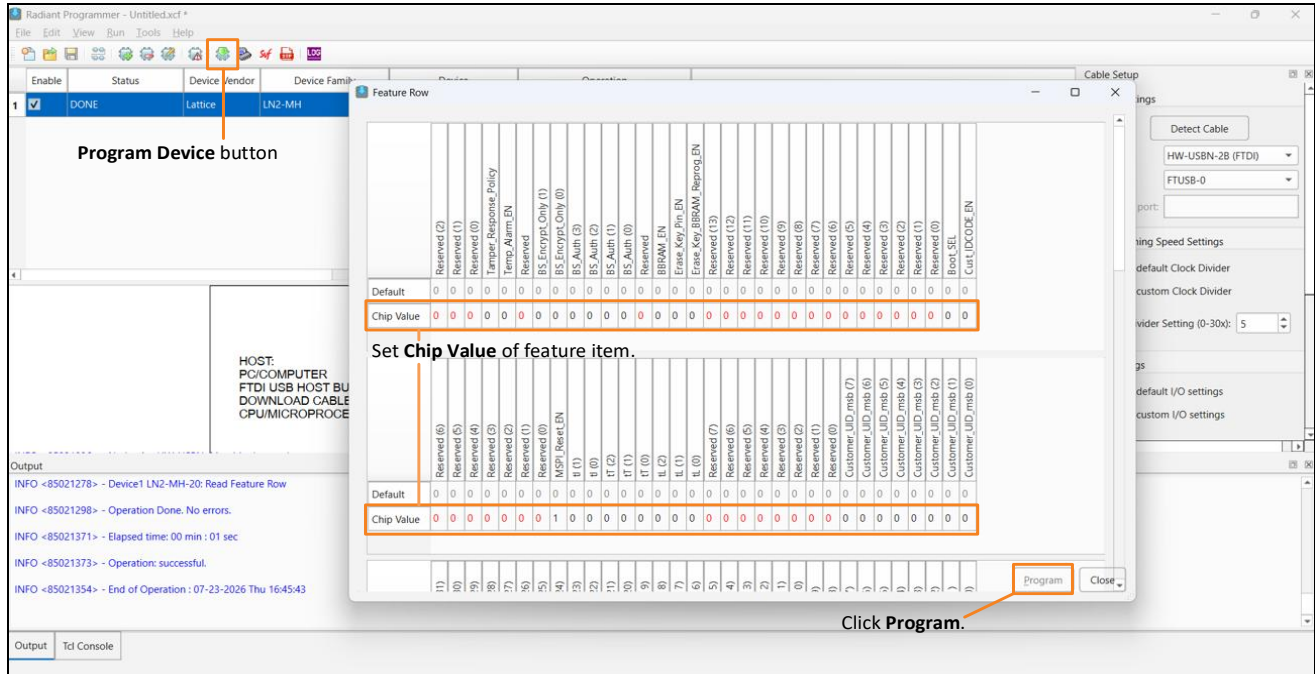


Figure C.4. Updating Feature Row

## C.2.4. Program Control NV Register 1

To program the non-volatile control register 1 (CR1), select **Program Control NV Register 1** for *Operation* and click **OK**. Next, click the **Program Device** button as shown in Figure C.5. The Control Register 1 window appears.

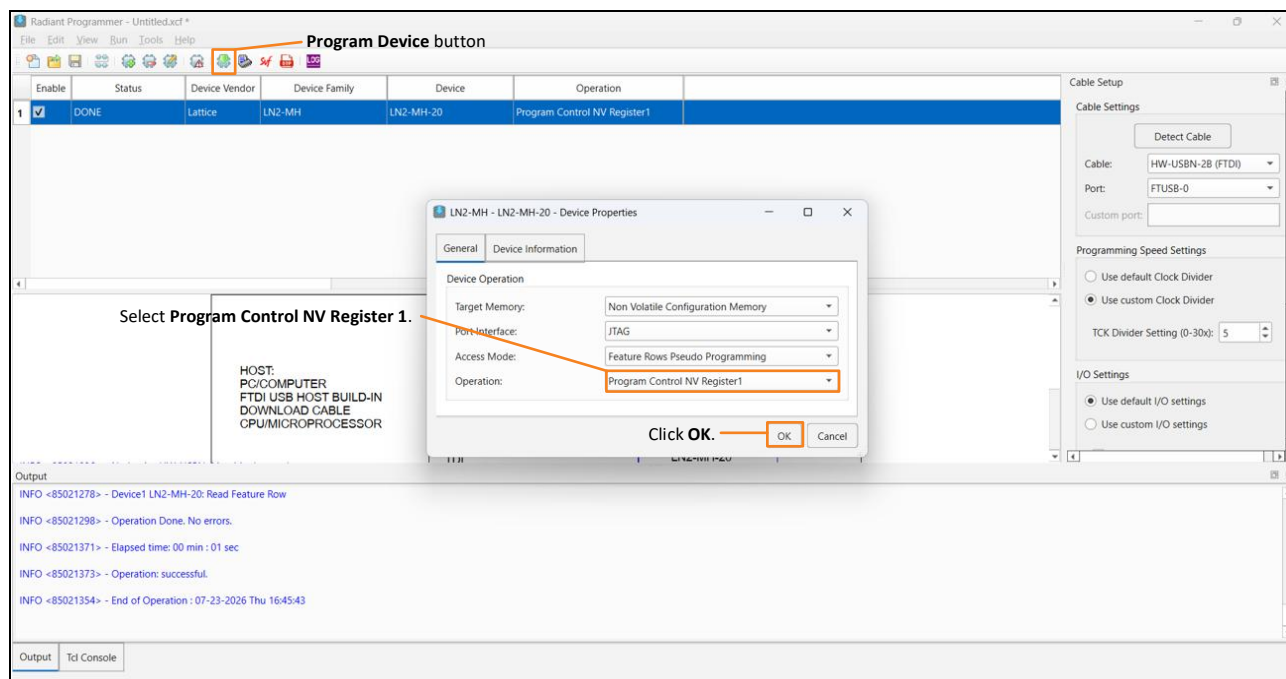


Figure C.5. Programming Non-volatile Control Register 1

The Control Register 1 window allows CR1 bits to be modified by setting the **Chip Value** or maintained by setting the **Mask Value** of the associated bit and then clicking **Program** as shown in Figure C.6. Refer to the [Control Register 1 \(CR1\)](#) section for descriptions of CR1 bits.

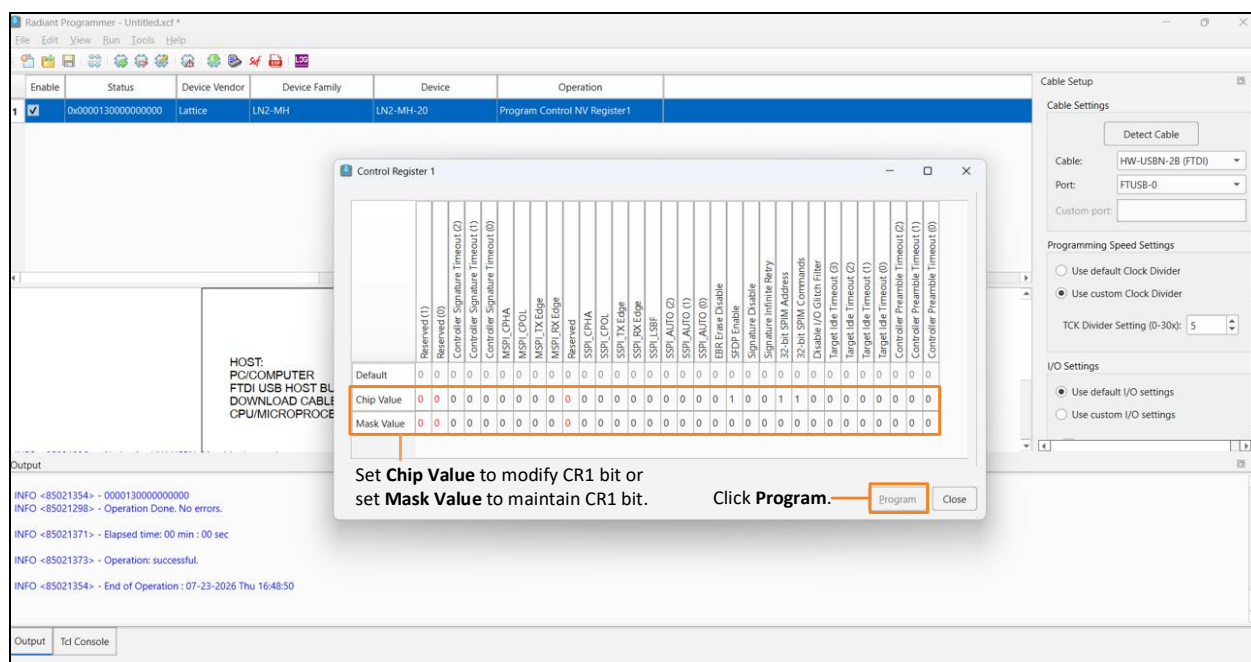


Figure C.6. Control Register 1 Window

### C.3. Advanced Security Keys Programming

For information on programming the encryption key, public key, lock policy, and ports interface lock, refer to the [Lattice Nexus 2 One-Time Programmable Security User Guide \(FPGA-TN-02436\)](#).

**Note:** For device setup finalization, it is strongly recommended that the feature row be locked for write after device setup is complete. This prevents the device setup from being accidentally altered.

## Appendix D. Configuration Access from User Logic

The Nexus 2 device configuration logic provides an LMMI interface to allow user logic residing inside the FPGA fabric to access the device configuration (CFG) functionalities. To achieve this, the CONFIG\_LMMIC primitive must be instantiated in the user design.

### D.1. CONFIG\_LMMIC Primitive

Figure D.1 shows the CONFIG\_LMMIC primitive pin diagram. Table D.1 lists the CONFIG\_LMMIC primitive pins.

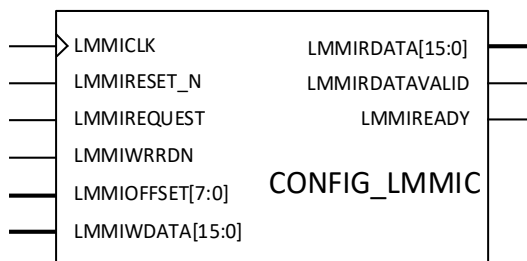


Figure D.1. CONFIG\_LMMIC Primitive Pin Diagram

Table D.1. CONFIG\_LMMIC Primitive Pins<sup>1</sup>

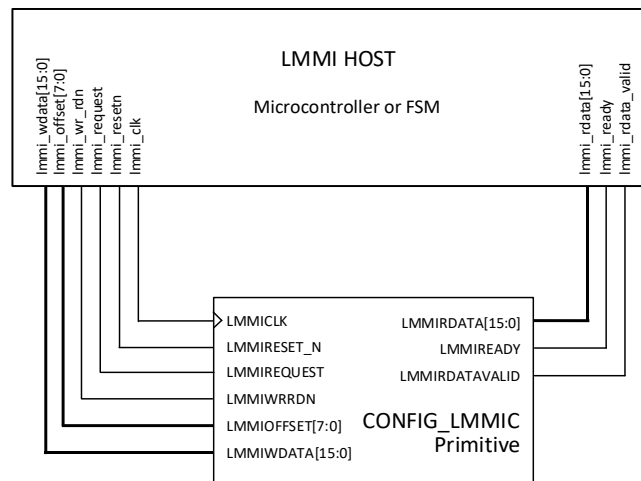
| Pin Name        | Direction | Description                                                         |
|-----------------|-----------|---------------------------------------------------------------------|
| LMMICLK         | In        | Clock for LMMI interface                                            |
| LMMIRESET_N     | In        | LMMI interface reset                                                |
| LMMIREQUEST     | In        | LMMI start transaction                                              |
| LMMIWRRDN       | In        | LMMI write/read control; 1=Write; 0=Read                            |
| LMMIOFFSET[7:0] | In        | LMMI register offset                                                |
| LMMIWDATA[15:0] | In        | LMMI write data                                                     |
| LMMIRDATA[15:0] | Out       | LMMI read data                                                      |
| LMMIRDATAVALID  | Out       | LMMI read transaction is complete and LMMIRDATA contains valid data |
| LMMIREADY       | Out       | LMMI ready signal                                                   |

**Note:**

1. There is no simulation model supported for the CONFIG\_LMMIC primitive.

## D.2. CONFIG\_LMMIC Primitive Connection and Instantiation

In the user design, the CONFIG\_LMMIC primitive should be connected as shown in Figure D.2.



**Figure D.2. CONFIG\_LMMIC Primitive Connection**

The following is a sample register-transfer level (RTL) design for instantiating the CONFIG\_LMMIC primitive:

```

wire      lmmi_clk;
wire      lmmi_resets;
wire      lmmi_request;
wire      lmmi_wr_rdn;
wire [7:0] lmmi_offset;
wire [15:0] lmmi_wdata;
wire [15:0] lmmi_rdata;
wire      lmmi_ready;
wire      lmmi_rdata_valid;

//
*****

Implement LMMI Host Logic here.

//
*****

CONFIG_LMMIC
config_lmmi (// Inputs
    .LMMICLK      (lmmi_clk),           // I -> Clock for LMMI
    .LMMIREQUEST  (lmmi_request),       // I -> Start Transaction
    .LMMIWRRDN    (lmmi_wr_rdn),        // I -> 1 = Write ; 0 = Read
    .LMMIOFFSET   (lmmi_offset),        // I -> Register Offset
    .LMMIWDATA    (lmmi_wdata),         // I -> Write Data
    .LMMIRESET_N  (lmmi_resets),        // I -> Reset the LMMI
    // Outputs
    .LMMIRDATA    (lmmi_rdata),         // O -> Read data
    .LMMIREADY    (lmmi_ready),         // O -> Ready Flag
    .LMMIRDATAVALID (lmmi_rdata_valid)); // O -> Read complete. LMMIRDATA is
valid

```

## D.3. CONFIG\_LMMIC Access Protocol

To access the CONFIG\_LMMIC primitive, the LMMI host logic should drive the LMMI interface by following the LMMI protocol. Refer to the [Lattice Memory Mapped Interface and Lattice Interrupt Interface \(FPGA-UG-02039\)](#) for details on the LMMI specification.

### D.3.1. LMMI Offset Assignment for CFG Access

The LMMI offset assignments for Nexus 2 device CFG access are shown in [Table D.2](#).

**Table D.2. LMMI CFG Offset List**

| LMMI Offset | Access | Target                                    | Default | Target Description                                                                                                                                                                                                                                                           |
|-------------|--------|-------------------------------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8'h00       | RW     | LMMI_CFG_DATA[15:0]                       | 16'h0   | Write, read operation for target configuration command execution.                                                                                                                                                                                                            |
| 8'h01       | RW     | LMMI_CFG_PORT_REQUEST[15:0] <sup>1</sup>  | 16'h0   | Request to enable LMMI access to CFG logic<br>LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000000; => NOWAIT<br>LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000001; => WAIT<br>LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000002; => FORCE<br>LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000003; => RELEASE |
| 8'h02       |        | LMMI_CFG_PORT_REQUEST[31:16] <sup>1</sup> | 16'h0   |                                                                                                                                                                                                                                                                              |
| 8'h03       | RO     | LMMI_CFG_PORT_STATUS[15:0]                | 16'h0   | Status of LMMI interface                                                                                                                                                                                                                                                     |
| 8'h04       |        | LMMI_CFG_PORT_STATUS[31:16]               | 16'h0   |                                                                                                                                                                                                                                                                              |
| 8'h05       | RW     | LMMI_CFG_MSPI_MULTIBOOT_ADDR [15:0]       | 16'h0   | Controller SPI multiple boot address                                                                                                                                                                                                                                         |
| 8'h06       |        | LMMI_CFG_MSPI_MULTIBOOT_ADDR [31:16]      | 16'h0   |                                                                                                                                                                                                                                                                              |
| 8'h07       |        | LMMI_CFG_MSPI_MULTIBOOT_ADDR [47:32]      | 16'h0   |                                                                                                                                                                                                                                                                              |
| 8'h08       | RW     | LMMI_CFG_16BIT_ENABLE                     | 16'h1   | LMMI CFG interface 16-bit data width enable<br>LMMI_CFG_16BIT_ENABLE[0] = 1'b0; => LMMI_CFG_DATA = {8'h00, CFG_DATA[7:0]};<br>LMMI_CFG_16BIT_ENABLE[0] = 1'b1; => LMMI_CFG_DATA = CFG_DATA[15:0]                                                                             |

**Note:**

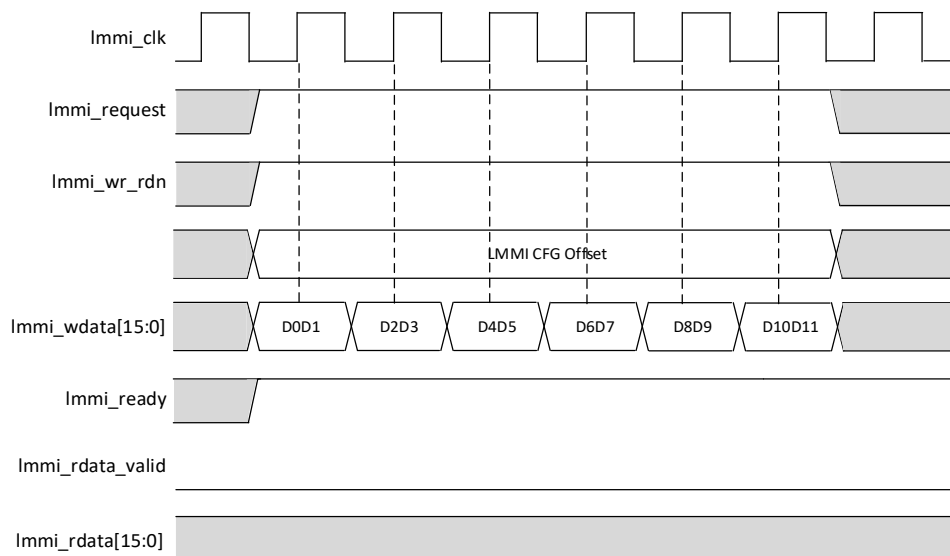
1. To enable LMMI access to CFG logic, you must write LMMI\_CFG\_PORT\_REQUEST[15:0] before LMMI\_CFG\_PORT\_REQUEST[31:16]. Failure to follow this sequence might cause the LMMI port to become inactive, thereby requiring power cycling of the device to recover the LMMI port.

### D.3.2. LMMI CFG Command Execution

When accessing any CFG internal register or any other CFG data, the non-JTAG target command format should be followed. The supported commands are listed in the [CONFIG\\_LMMIC Supported Commands](#) section.

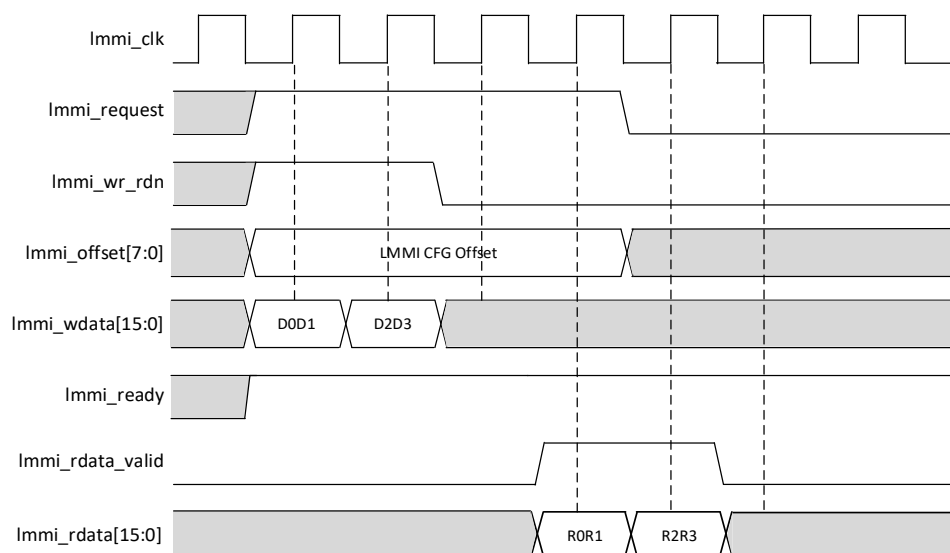
To execute the target configuration command, the LMMI host should perform four back-to-back write transfers for the four command bytes, followed by the optional data bytes. The typical LMMI write and read operations are illustrated in [Figure D.3](#) and [Figure D.4](#).

Using the PROG\_CNTRL0 command as an example, for the lmmi\_wdata bus, D0 is 0x80, D1 is 0x08, D2 is 0x03, D3 is 0x00, D4 is DATA[31:24], D5 is DATA[23:16], D6 is DATA[15:8], D7 is DATA[7:0], D8 is MASK[31:24], D9 is MASK[23:16], D10 is MASK[15:8], and D11 is MASK[7:0]. For more information, refer to the [PROG\\_CNTRL0](#) section.



**Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode**

Using the READ\_CNTRL0 command as an example, on the write phase (lmmi\_wr\_rdn at high), D0 is 0x84, D1 is 0x00, D2 is 0x04, and D3 is 0x00. On the read phase, R0, R1, R2, and R3 are the readback data for READ\_CNTRL0. For different commands, the readback data sizes are different. For more information, refer to the [READ\\_CNTRL0](#) section.



**Figure D.4. Waveform for LMMI Read Operation**

### D.3.3. CONFIG\_LMMIC Supported Commands

Only selected target configuration commands are supported by CONFIG\_LMMIC. The supported commands are listed in Table D.3 and are a subset of the target configuration commands listed in Table 6.12.

**Table D.3. CONFIG\_LMMIC Supported Commands**

| Command Name <sup>1</sup> | Byte0 | Byte1 | Byte2 | Byte3            | # DATA BYTES | # RETURN BYTES | PROG_QUALIFY | Type <sup>2</sup> | Description                                                                                                  |
|---------------------------|-------|-------|-------|------------------|--------------|----------------|--------------|-------------------|--------------------------------------------------------------------------------------------------------------|
| PROG_ENABLE               | 0x80  | 0x00  | 0x0c  | 0x00–0x02        | —            | —              | —            | IMM               | Enable programming based on byte3 value:<br>0x00 – CRAM/INIT/OTP<br>0x01 – CRAM/INIT only<br>0x02 – OTP only |
| PROG_DISABLE              | 0x80  | 0x00  | 0x0e  | 0x00             | —            | —              | —            | IMM               | Disable programming of CRAM/INIT/OTP.                                                                        |
| READ_IDCODE_PUB           | 0x01  | 0x01  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read 32-bit public IDCODE of the device.                                                                     |
| READ_IDCODE_PRV           | 0x01  | 0x02  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read 32-bit private IDCODE for the device.                                                                   |
| READ_UIDCODE_PUB_L        | 0x01  | 0x03  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read bits[31:0] of TraceID[63:0]                                                                             |
| READ_UIDCODE_PUB_H        | 0x01  | 0x04  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read bits[63:32] of TraceID[63:0]                                                                            |
| READ_SECURE_UID           | 0x84  | 0x00  | 0x28  | 0x00             | —            | 32             | —            | Read              | Read the device 256-bit Secure Unique ID.                                                                    |
| READ_USERCODE             | 0x01  | 0x05  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read 32-bit user code register.                                                                              |
| READ_DR_USERCODE          | 0x01  | 0x06  | 0x00  | 0x00             | —            | 4              | Y            | Read              | Read dry-run user code shadow register.                                                                      |
| READ_STATUS0              | 0x01  | 0x07  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read Status Register 0.                                                                                      |
| READ_STATUS1              | 0x01  | 0x08  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read Status Register 1.                                                                                      |
| READ_STATUS2              | 0x01  | 0x09  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read Status Register 2.                                                                                      |
| CHECK_BUSY                | 0x01  | 0x0b  | 0x00  | 0x00             | —            | 4              | —            | Read              | Read busy flag from Status Register 0.                                                                       |
| PROG_CNTRL0               | 0x80  | 0x08  | 0x03  | 0x00             | 8            | —              | Y            | Data Write        | Write Control Register 0.                                                                                    |
| READ_CNTRL0               | 0x84  | 0x00  | 0x04  | 0x00             | —            | 4              | —            | Read              | Read Control Register 0.                                                                                     |
| PROG_CNTRL1               | 0x80  | 0x08  | 0x05  | 0x00             | 8            | —              | Y            | Data Write        | Write Control Register 1.                                                                                    |
| READ_CNTRL1               | 0x84  | 0x00  | 0x06  | 0x00             | —            | 4              | —            | Read              | Read Control Register 1.                                                                                     |
| READ_UMR                  | 0x84  | 0x00  | 0x08  | 0x00             | —            | 16             | —            | Read              | Read user mode register.                                                                                     |
| DEVICE_CTRL               | 0x80  | 0x00  | 0x01  | CMD              | —            | —              | —            | IMM               | Device control.                                                                                              |
| DRY_RUN_CTRL              | 0x80  | 0x00  | 0x09  | MODE             | —            | —              | —            | IMM               | Dry run control.                                                                                             |
| REFRESH                   | 0x80  | 0x00  | 0x0a  | 0x00             | —            | —              | —            | Delay             | Equivalent to toggling the PROGRAMN pin.                                                                     |
| MSPI_BRIDGE               | 0x16  | 0x00  | 0x00  | 0x00             | Variable     | —              | —            | Data Write        | Bridge Data to Controller SPI.                                                                               |
| MSPI_BRIDGE_CLOCK         | 0x80  | 0x00  | 0x1d  | MODE             | —            | —              | —            | IMM               | Change controller SPI bridge clock settings.                                                                 |
| READ_SED_CRC              | 0x01  | 0x0c  | 0x00  | 0x00             | —            | 4              | Y            | Read              | Read expected SED 32-bit CRC.                                                                                |
| CALC_SED_CRC              | 0x80  | 0x00  | 0x4f  | 0x00             | —            | —              | —            | Delay             | Run SED scan and calculate 32-bit SED CRC in the device.                                                     |
| READ_SED_CRC_CALC         | 0x01  | 0x0d  | 0x00  | 0x00             | —            | 4              | Y            | Read              | Read 32-bit SED CRC calculated by the device.                                                                |
| READ_OTP                  | 0x84  | 0x00  | 0x4e  | ROW <sup>3</sup> | —            | 4              | —            | Read              | Read user OTP.                                                                                               |
| PROG_OTP                  | 0x80  | 0x04  | 0x4c  | ROW <sup>3</sup> | 4            | —              | Y            | Data Write        | Write user OTP.                                                                                              |
| PROG_OTP_SHADOW           | 0x80  | 0x08  | 0x55  | ROW <sup>3</sup> | 8            | —              | Y            | Data Write        | Write user OTP shadow register.                                                                              |
| LOCK_OTP_ROW              | 0x80  | 0x00  | 0x4d  | ROW <sup>3</sup> | —            | —              | Y            | IMM               | Lock user OTP row.                                                                                           |

| Command Name <sup>1</sup> | Byte0 | Byte1 | Byte2 | Byte3 | # DATA BYTES | # RETURN BYTES | PROG_QUALIFY | Type <sup>2</sup> | Description                  |
|---------------------------|-------|-------|-------|-------|--------------|----------------|--------------|-------------------|------------------------------|
| LOCK_USER_OTP_BLOCK       | 0x80  | 0x00  | 0x54  | 0x00  | —            | —              | Y            | IMM               | Lock user block of OTP rows. |

**Notes:**

1. The PORT\_REQUEST and PORT\_STATUS\_READ commands are not supported for the LMMI interface. Refer to [Table D.2](#) for information on sending these commands.
2. The different command types are described in the [Command Waveforms](#) section.
3. For more information on OTP memory access through the LMMI interface, refer to the Lattice Nexus 2 Configuration Security User Guide (FPGA-TN-02369).

## Appendix E. Flash Programming

### E.1. Flash Memory Space Partition

Table E.1 shows the number of internal memory blocks available for the LN2-MH-70ES1 device. Table E.2 shows the number of internal memory blocks available for the LN2-MH-20 and LN2-MH-20ES1 devices.

**Table E.1. Number of Blocks of Flash Memory for LN2-MH-70ES1 Device**

| Start Block | No. of Blocks <sup>1</sup> | Byte Address | Function               | Output File Generated by Radiant |
|-------------|----------------------------|--------------|------------------------|----------------------------------|
| 0           | 1                          | 0x00000000   | Header and Jump Table  | <user_project_design>_header.mcs |
| 1           | 191                        | 0x00010000   | CFG0(IMAGE0)           | <user_project_design>_0.jed      |
| 192         | 24                         | 0x00C00000   | UFM0 (EBR & LRAM INIT) | <user_project_design>_u0.jed     |
| 216         | 191                        | 0x00D80000   | CFG1(IMAGE1)           | <user_project_design>_1.jed      |
| 407         | 24                         | 0x01970000   | UFM1 (EBR & LRAM INIT) | <user_project_design>_u1.jed     |
| 431         | 10                         | 0x01AF0000   | USERDATA0              | <user_project_design>_ud0.mcs    |
| 441         | 10                         | 0x01B90000   | USERDATA1              | <user_project_design>_ud1.mcs    |
| 451         | 10                         | 0x01C30000   | USERDATA2              | <user_project_design>_ud2.mcs    |
| 461         | 10                         | 0x01CD0000   | USERDATA3              | <user_project_design>_ud3.mcs    |
| 471         | 10                         | 0x01D70000   | USERDATA4              | <user_project_design>_ud4.mcs    |
| 481         | 10                         | 0x01E10000   | USERDATA5              | <user_project_design>_ud5.mcs    |
| 491         | 10                         | 0x01EB0000   | USERDATA6              | <user_project_design>_ud6.mcs    |
| 501         | 10                         | 0x01F50000   | USERDATA7              | <user_project_design>_ud7.mcs    |
| 511         | 1                          | 0x01FF0000   | Backup Jump Table      | <user_project_design>_tail.mcs   |

**Note:**

- Each block represents 65,536 bytes of data.

**Table E.2. Number of Blocks of Flash Memory for LN2-MH-20 and LN2-MH-20ES1 Devices**

| Start Block | No. of Blocks <sup>1</sup> | Byte Address | Function               | Output File Generated by Radiant |
|-------------|----------------------------|--------------|------------------------|----------------------------------|
| 0           | 1                          | 0x00000000   | Header and Jump Table  | <user_project_design>_header.mcs |
| 1           | 80                         | 0x00010000   | CFG0(IMAGE0)           | <user_project_design>_0.jed      |
| 81          | 22                         | 0x00510000   | UFM0 (EBR & LRAM INIT) | <user_project_design>_u0.jed     |
| 103         | 80                         | 0x00670000   | CFG1(IMAGE1)           | <user_project_design>_1.jed      |
| 183         | 22                         | 0x00B70000   | UFM1 (EBR & LRAM INIT) | <user_project_design>_u1.jed     |
| 205         | 80                         | 0x00CD0000   | CFG2(IMAGE2)           | <user_project_design>_2.jed      |
| 285         | 22                         | 0x011D0000   | UFM2 (EBR & LRAM INIT) | <user_project_design>_u2.jed     |
| 307         | 80                         | 0x01330000   | CFG3(IMAGE3)           | <user_project_design>_3.jed      |
| 387         | 22                         | 0x01830000   | UFM3 (EBR & LRAM INIT) | <user_project_design>_u3.jed     |
| 409         | 12                         | 0x01990000   | USERDATA0              | <user_project_design>_ud0.mcs    |
| 421         | 12                         | 0x01A50000   | USERDATA1              | <user_project_design>_ud1.mcs    |
| 433         | 12                         | 0x01B10000   | USERDATA2              | <user_project_design>_ud2.mcs    |
| 445         | 12                         | 0x01BD0000   | USERDATA3              | <user_project_design>_ud3.mcs    |
| 457         | 12                         | 0x01C90000   | USERDATA4              | <user_project_design>_ud4.mcs    |
| 469         | 12                         | 0x01D50000   | USERDATA5              | <user_project_design>_ud5.mcs    |
| 481         | 12                         | 0x01E10000   | USERDATA6              | <user_project_design>_ud6.mcs    |
| 493         | 18                         | 0x01ED0000   | USERDATA7              | <user_project_design>_ud7.mcs    |
| 511         | 1                          | 0x01FF0000   | Backup Jump Table      | <user_project_design>_tail.mcs   |

**Note:**

- Each block represents 65,536 bytes of data.

## Appendix F. JTAG Daisy-Chain Configuration with External Host Controller

The appropriate bitstream data for Nexus 2 devices can be extracted from Lattice software-generated programming files (such as a STAPL file) for use by an external host controller to perform JTAG daisy-chain configuration. Due to current Nexus 2 device limitations, bitstream data must be correctly adjusted for JTAG daisy-chain configuration. Lattice programming tools perform this adjustment automatically. However, when a non-Lattice programming tool is used, the bitstream must be pre-adjusted. This section covers the following:

- Creating a JTAG daisy-chain configuration using Lattice software.
- Saving the .xcf file that contains the JTAG configuration used to define the daisy chain.
- Generating a STAPL file from the .xcf file.
- Locating and extracting bitstream data or adjusted bitstream data in the STAPL file for each Nexus 2 device in the daisy chain.

### F.1. Overview and Limitations

When Nexus 2 devices are not configured using the Lattice programming software but through JTAG in a daisy chain using an external host controller, the host controller is responsible for the following:

- Driving the JTAG state machine.
- Handling instruction or data register scans.
- Shifting all configuration data through the chain in the correct order.

Nexus 2 devices in the JTAG daisy chain (except when the Nexus 2 device is the first device in the chain) require *adjusted bitstream data* for successful configuration. Lattice software automatically generates these adjusted data segments when the STAPL file for a JTAG daisy chain is created. These adjusted data segments must be extracted for use by corresponding devices in the daisy chain. Using unadjusted data segments will result in configuration failure.

The procedures described in the following sections use an example daisy chain with two Nexus 2 LN2-CT-20ES1 devices. These procedures similarly apply to daisy chains with more than two Nexus 2 devices or with a mix of Nexus 2 and other devices.

## F.2. Creating a JTAG Daisy-Chain Configuration

Before generating a daisy-chain STAPL file, you must first create and verify the JTAG daisy-chain configuration in the Lattice programming tool. The example daisy chain, as shown in [Figure F.1](#), contains two devices as follows:

- Device 0: LN2-CT-20ES1 device (first device in the chain that is closer to TDI)
- Device 1: LN2-CT-20ES1 device (second device in the chain that is closer to TDO)

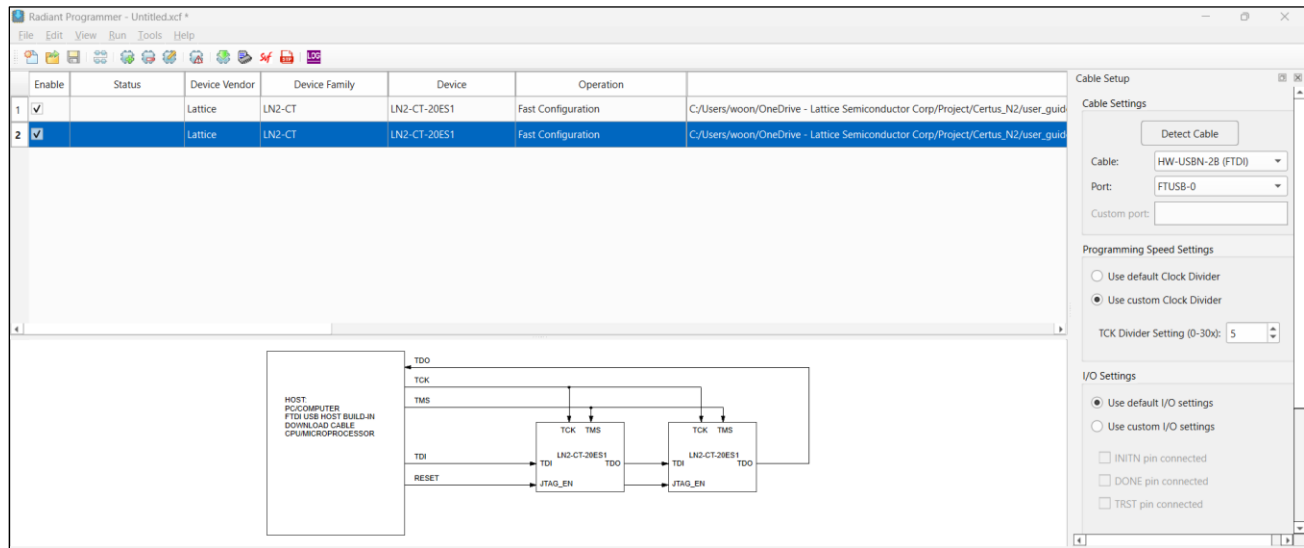


Figure F.1. JTAG Daisy Chain – Radiant Programmer

To create the JTAG daisy-chain configuration and define the programming operation, perform the following:

1. Launch the Lattice Radiant Programmer.
2. In the **Getting Started** window, select **Create a new blank project**, then click **OK**.

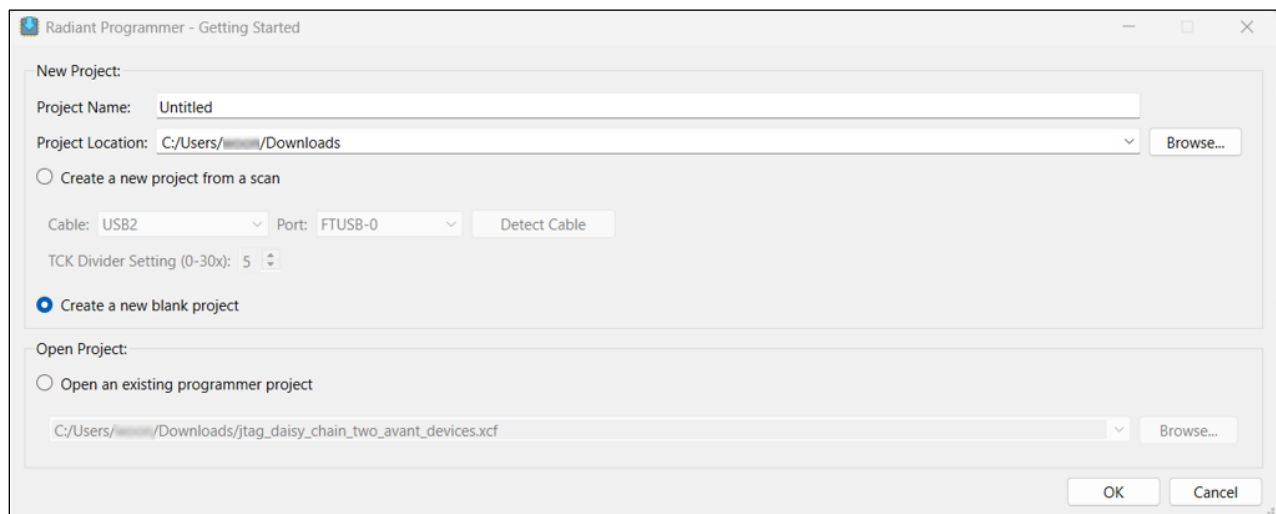
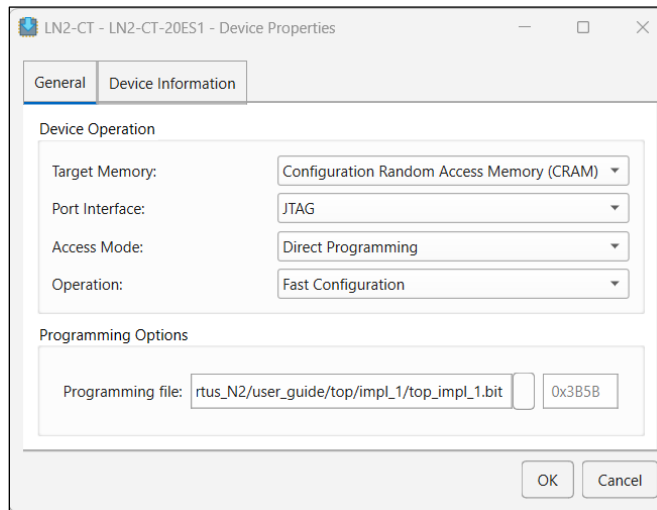


Figure F.2. JTAG Daisy Chain – Creating a New Blank Project

3. Detect or add devices to the JTAG chain.
  - If automatic chain detection is supported and a compatible cable is connected:
    - i. Click the **Scan Chain** button to automatically add devices in the JTAG daisy chain.
  - Otherwise, manually add each device.
    - j. Click the **Add Device** button.

- k. Select the applicable Nexuse 2 device (for example, LN2-CT-20ES1) from the device list.
  - l. Arrange the devices in the sequence order from TDI to TDO to match your hardware.
4. Assign programming files to each device.
  - For each device in the chain:
  - a. Double-click the **Operation** cell associated with the device. The **Device Properties** window appears.



**Figure F.3. JTAG Daisy Chain – Configuring the Device**

- b. Select the desired programming operation (for example, **Fast Configuration**).
    - c. Click the ... (Browse) button associated with **Programming file**.
    - d. Navigate to and select the compiled programming file (.bit file) for the device.
    - e. Click **OK** in the **Device Properties** window.
  5. Verify the chain and configuration options by confirming the following:
    - The order of devices matches hardware.
    - Each Nexus 2 device is assigned the correct .bit file.
    - The *Fast Configuration* operation is enabled.
- The JTAG daisy chain and programming operation are now fully defined through the Radiant Programmer.

### F.3. Saving the .xcf File

The .xcf file captures the JTAG sequence and is used by the Lattice deployment tool to generate a STAPL file.

To save the JTAG daisy-chain configuration as a .xcf file, perform the following:

1. In the Radiant Programmer, select **File > Save Project As**.
2. In the **Save XCF File As** window, choose a destination directory.
3. Enter a filename (for example, *Nexus2\_DaisyChain\_CT20.xcf*).
4. Ensure the file type is set to **XCF File (\*.xcf)**.
5. Click **Save**.

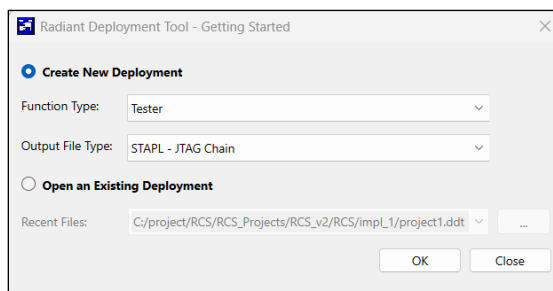
The .xcf file now contains the exact JTAG daisy-chain configuration (sequence of devices, order, operations, and associated programming files).

### F.4. Generating a STAPL File from the .xcf File

A STAPL file contains the adjusted bitstream data for each Nexus 2 device in the daisy chain.

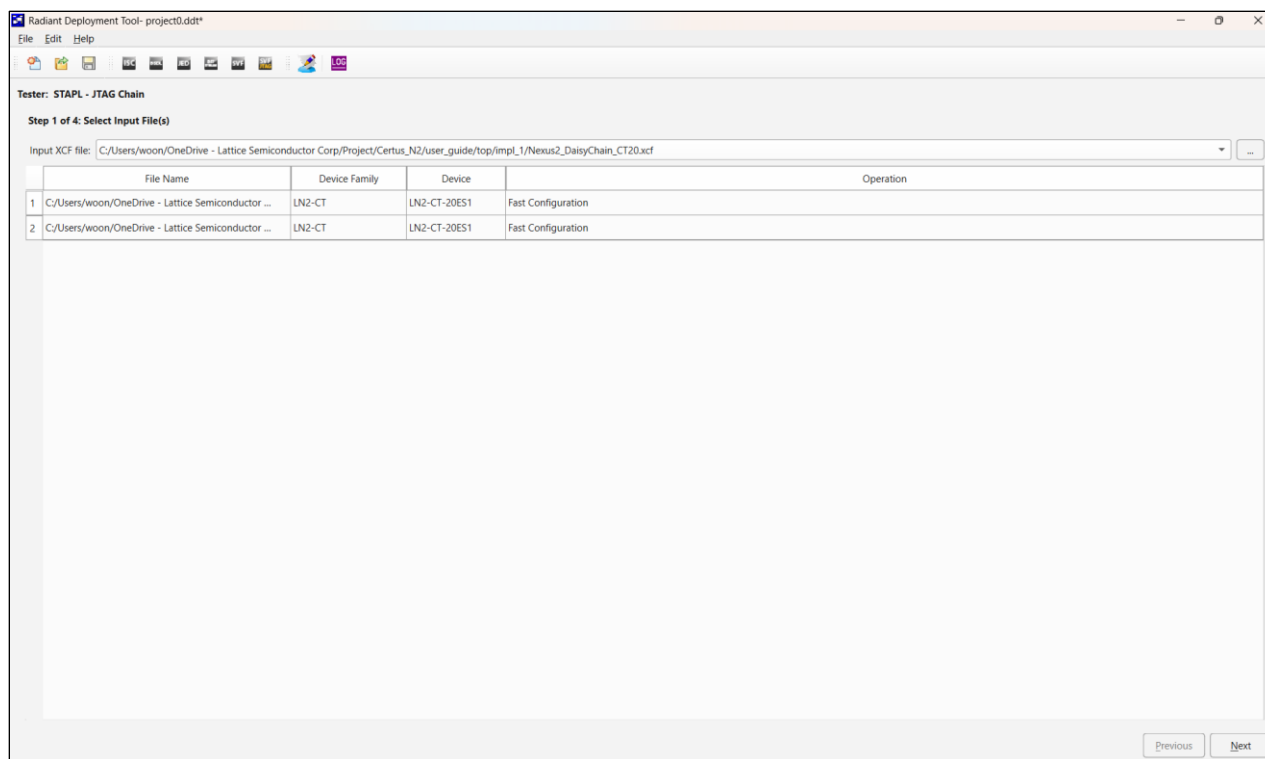
To generate a STAPL file from the .xcf file, perform the following:

1. In the Radiant Programmer, select **Tools > Deployment Tool** to launch the Lattice Radiant Deployment Tool. The **Getting Started** window appears.



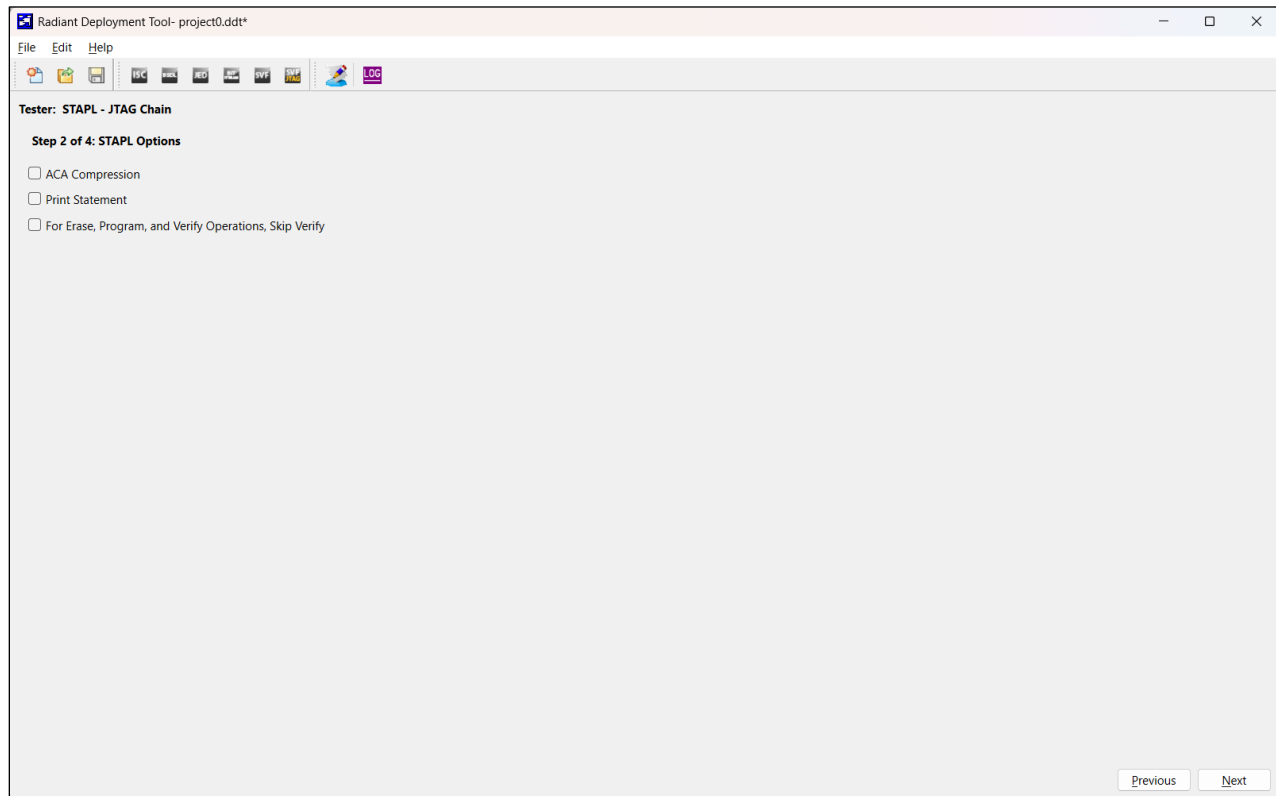
**Figure F.4. Radiant Deployment Tool – Getting Started Window**

2. Select the following options:
  - Create New Deployment
  - Function Type: Tester
  - Output File Type: STAPL – JTAG Chain
3. Click **OK**.
4. In the Radiant Deployment Tool *Step 1 of 4: Select Input File(s)*:
  - a. Click the ... (Browse) button associated with the **Input XCF File** field.
  - b. Navigate to and select the .xcf file you saved earlier (for example, *Nexus2\_DaisyChain\_CT20.xcf*).
  - c. Confirm that the deployment tool correctly lists the devices in the JTAG daisy chain and corresponding programming files and operations.
  - d. Click **Next**.



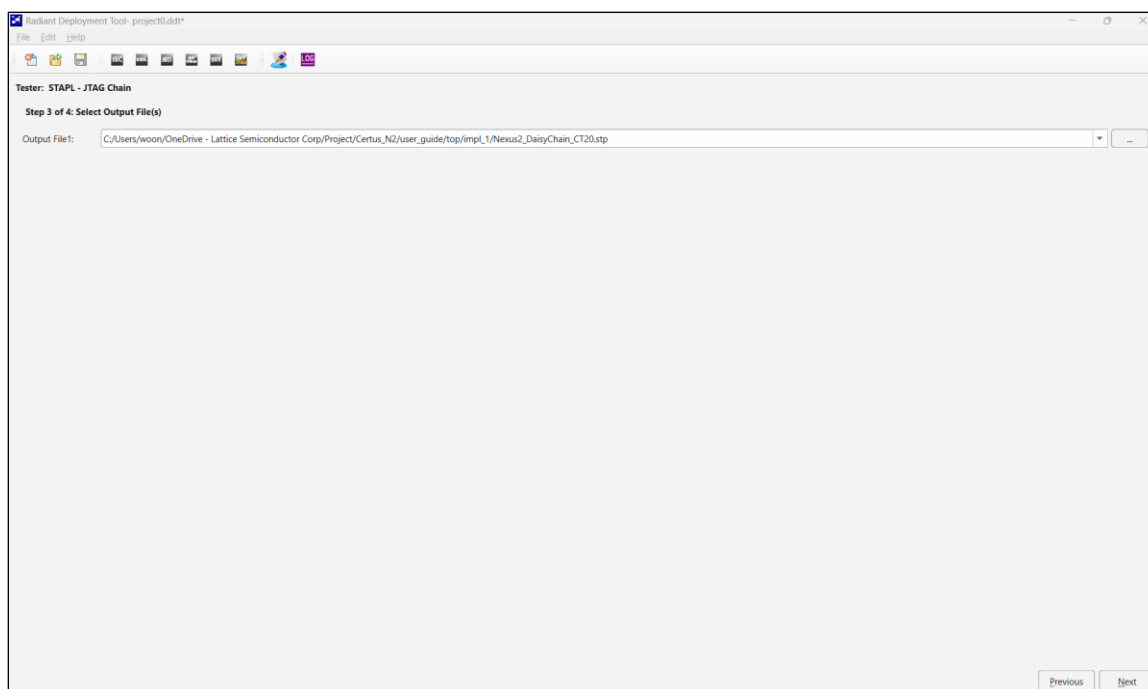
**Figure F.5. Radiant Deployment Tool – Step 1 of 4: Select Input File(s)**

5. In the Radiant Deployment Tool *Step 2 of 4: STAPL Options*:
  - a. Uncheck all options.
  - b. Click **Next**.



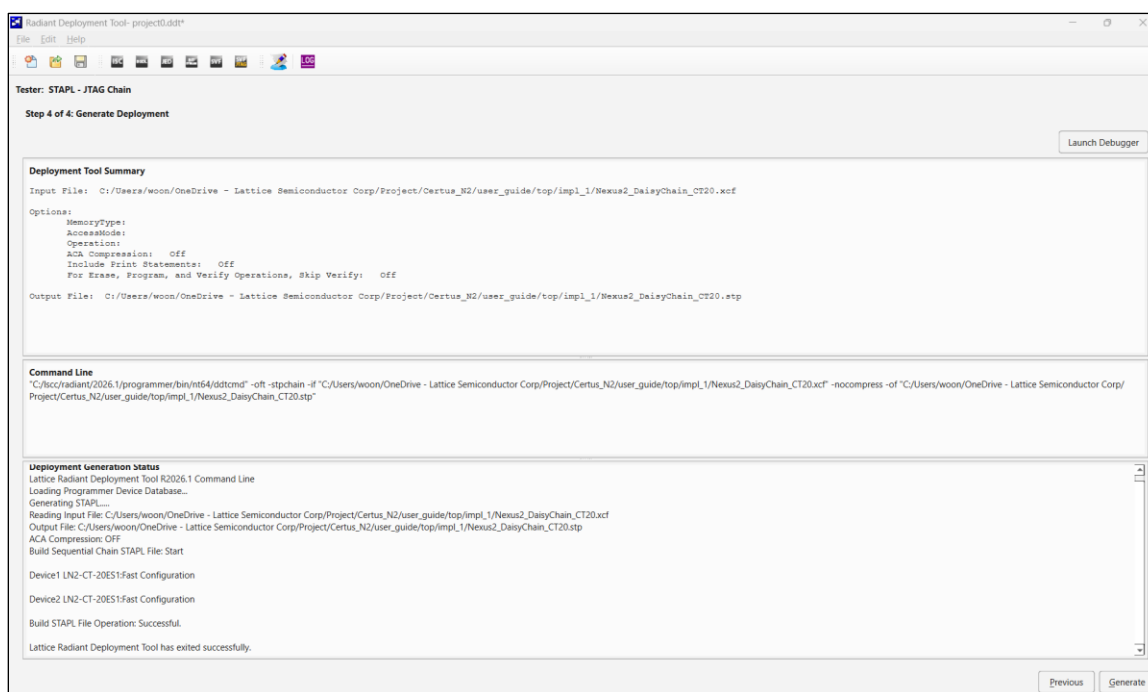
**Figure F.6. Radiant Deployment Tool – Step 2 of 4: STAPL Options**

6. In the Radiant Deployment Tool *Step 3 of 4: Select Output File(s)*:
  - a. Click the ... (Browse) button associated with the **Output File 1** field.
  - b. Choose a destination directory.
  - c. Enter a filename (for example, *Nexus2\_DaisyChain\_CT20.stp*).
  - d. Ensure the file type is set to STAPL (\*.stp).
  - e. Click **Save**.
  - f. Click **Next**.



**Figure F.7. Radiant Deployment Tool – Step 3 of 4: Select Output File(s)**

7. In the Radiant Deployment Tool *Step 4 of 4: Generate Deployment*, click **Generate**.



**Figure F.8. Radiant Deployment Tool – Step 4 of 4: Generate Deployment**

Once the operation is successful, a STAPL file that encapsulates the full JTAG daisy-chain configuration sequence for the Nexus 2 devices, including the required adjusted bitstream data, is now available.

## F.5. Locating and Extracting Adjusted Bitstream Data in the STAPL File

The STAPL file is a text script that contains the following:

- JTAG instructions and sequences
- Data arrays and constants representing configuration bitstreams

With the STAPL file generated, to locate and extract the bitstream data segments that must be used by the external host controller for each Nexus 2 device in the JTAG daisy chain, perform the following:

1. Open the generated STAPL file (for example, *Nexus2\_DaisyChain\_CT20.stp*) using a text editor capable of handling large files.
2. In the text editor, use the search function to locate bitstream identifiers for the Nexus 2 devices.  
For example, in the daisy chain that contains two devices:
  - First Nexus 2 device (Device 0 in the chain) – search for the bitstream keyword *FUSEARRAY\_TDI\_0*. This keyword corresponds to the bitstream data (typically an array or long data string) used for the first Nexus 2 device in the chain.
  - Second Nexus 2 device (Device 1 in the chain) – search for the bitstream keyword *AdjustFUSEARRAY\_TDI\_1*. This keyword corresponds to the adjusted bitstream data (typically an array or long data string) that must be used for the second Nexus 2 device in the daisy chain. Note that the STAPL file may also contain *FUSEARRAY\_TDI\_1*. However, this is not the adjusted bitstream data and must not be used for configuration in the daisy chain.

**Note:** Do not use the bitstream data associated with the keyword format *FUSEARRAY\_TDI\_<n – 1>*, where n is the n<sup>th</sup> Nexus 2 device in the device chain, for Nexus 2 devices beyond the first device. The associated bitstream data are not adjusted for daisy chain use and will cause configuration failures in the devices. For any Nexus 2 device after the first device, search for the bitstream data associated with the keyword format *AdjustFUSEARRAY\_TDI\_<n – 1>*, where n is the n<sup>th</sup> Nexus 2 device in the daisy chain.

3. Extract the bitstream data by copying the entire data payload associated with the desired bitstream keywords (for example, *FUSEARRAY\_TDI\_0* and *AdjustFUSEARRAY\_TDI\_1*). The bitstream data is formatted as hex strings. Preserve the order of the bits or bytes as defined.

Bitstream data for each Nexus 2 device can now be provided to the host controller for configuring corresponding devices in the JTAG daisy chain.

## 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](#)
- [Mach-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 Multi-Boot User Guide \(FPGA-TN-02385\)](#)
- [Lattice Nexus 2 sysCONFIG User Guide \(FPGA-TN-02370\)](#)
- [Lattice Nexus 2 sysDSP User Guide \(FPGA-TN-02362\)](#)
- [Lattice Nexus 2 sys/O User Guide \(FPGA-TN-02365\)](#)
- [Sub-LVDS Signaling Using Lattice Devices \(FPGA-TN-02028\)](#)
- [Electrical Recommendations for Lattice SERDES \(FPGA-TN-02077\)](#)
- [High-Speed PCB Design Considerations \(FPGA-TN-02178\)](#)
- [Lattice Nexus 2 Configuration Security User Guide \(FPGA-TN-02369\)](#)
- [Lattice Nexus 2 One-Time Programmable Security User Guide \(FPGA-TN-02436\)](#)

Other references:

- [Lattice Memory Mapped Interface and Lattice Interrupt Interface \(FPGA-UG-02039\)](#)
- [Watchdog Timer IP User Guide \(FPGA-IPUG-02097\)](#)
- [Lattice Radiant Software web page](#)
- [Lattice Insights](#) for Lattice Semiconductor training courses and learning plans

## Technical Support Assistance

Submit a technical support case through [www.latticesemi.com/techsupport](http://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, September 2026

| Section                        | Change Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| All                            | <ul style="list-style-type: none"> <li>Updated descriptions to reference internal non-volatile memory, internal flash, or on-chip flash where applicable.</li> <li>Updated description to reference both <i>external SPI storage device</i> and <i>internal flash memory</i> where applicable.</li> <li>Updated descriptions to reference self-download mode where applicable.</li> <li>Changed references from <i>SPI flash</i> or <i>SPI flash memory</i> to <i>non-volatile memory</i> where applicable.</li> <li>Changed references from <i>SPI flash device</i> to <i>non-volatile memory storage device</i> where applicable.</li> <li>Changed references from <i>SPI flash</i> to <i>flash</i> where applicable.</li> <li>Made minor editorial changes.</li> </ul>                                                                                                                                                                                                                                                                                                                                                               |
| Abbreviations in This Document | <ul style="list-style-type: none"> <li>Added <i>CPHA</i>, <i>CPOL</i>, <i>FPGA</i>, <i>GOE</i>, <i>GSRN</i>, <i>GWE</i>, <i>IDCODE</i>, <i>SDM</i>, and <i>STAPL</i>.</li> <li>Removed <i>PAR</i>, <i>SEC</i>, and <i>SER</i>.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Introduction                   | <ul style="list-style-type: none"> <li>Removed reference to section on finalizing the device setup.</li> <li>Added notes regarding Mach-N2 device.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Features                       | Added references to on-chip flash.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Definitions of Terms           | <p>Updated definitions for the following:</p> <ul style="list-style-type: none"> <li>BIT</li> <li>Dual-Boot</li> <li>Multi-Boot</li> <li>Ping-Pong Boot</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Configuration Details          | <ul style="list-style-type: none"> <li>In <a href="#">Table 4.1. Bitstream Size versus Recommended SPI Flash Size</a>: <ul style="list-style-type: none"> <li>Updated bitstream size for LN2-CT-20 (maximum EBR only).</li> <li>Updated recommended SPI flash sizes for LN2-CT-20 (maximum EBR, single boot and dual boot).</li> <li>Added LN2-MH-20 device.</li> </ul> </li> <li>In the <a href="#">Configuration Ports Arbitration</a> section: <ul style="list-style-type: none"> <li>Added note on controller SPI mode and SDM applicability.</li> </ul> </li> <li>In <a href="#">Figure 4.1. Configuration Control Flow</a>: <ul style="list-style-type: none"> <li>Updated figure to include note explaining <i>I/O Ready</i>.</li> <li>Added note cross references within the diagram.</li> <li>Added "JTAG Active" decision diamond in the CFGMODE High = Yes path.</li> </ul> </li> <li>In <a href="#">Figure 4.3. Configuration from PROGRAMN Timing</a>: <ul style="list-style-type: none"> <li>Updated figure to include <math>t_{INIT\_HIGH}</math>.</li> <li>Added note on <math>t_{INITL}</math>.</li> </ul> </li> </ul> |
| Configuration Process and Flow | <ul style="list-style-type: none"> <li>In the <a href="#">Configuration</a> section: <ul style="list-style-type: none"> <li>Added note on controller SPI mode and SDM applicability.</li> <li>Added the <a href="#">Self-Download Mode Configuration</a> section.</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Device Configuration           | <ul style="list-style-type: none"> <li>In the <a href="#">Controller SPI Mode</a> section: <ul style="list-style-type: none"> <li>Added note on applicability of mode only to Certus-N2 devices.</li> </ul> </li> <li>Removed the Specifications and Timing Diagrams - Target SPI Port Waveforms section under <a href="#">Target SPI Mode</a>.</li> <li>Updated <a href="#">Figure 6.4. Read Command Waveforms</a>, <a href="#">Figure 6.6. Data Write Command Waveforms</a>, and <a href="#">Figure 6.7. Immediate Action Command Waveforms</a>.</li> <li>Updated <a href="#">Figure 6.5. Data Read Command Waveforms</a> and title to match section header.</li> <li>In the <a href="#">JTAG Mode</a> section: <ul style="list-style-type: none"> <li>Added a note to the first bullet of the advantages of the JTAG port list.</li> </ul> </li> <li>In the <a href="#">JTAG to Controller SPI Bridge</a> section:</li> </ul>                                                                                                                                                                                                        |

| Section                                 | Change Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                         | <ul style="list-style-type: none"> <li>Added note on JTAG to controller SPI bridge accessibility in relation to CFGMODE.</li> <li>In the <a href="#">Dual-Boot and Multi-Boot</a> section: <ul style="list-style-type: none"> <li>Updated description.</li> <li>Added reference to Lattice Avant Multi-Boot User Guide for dual-boot mode.</li> </ul> </li> <li>In the <a href="#">Ping-Pong Boot</a> section: <ul style="list-style-type: none"> <li>Updated description.</li> <li>Added reference to Lattice Avant Multi-Boot User Guide for ping-pong boot mode.</li> <li>Updated <a href="#">Figure 6.14. Jump Table</a>.</li> </ul> </li> <li>In <a href="#">Table 6.7. Status Register 1</a>: <ul style="list-style-type: none"> <li>Updated bit[27] to reserved.</li> <li>Updated bits[19:18] to Device State and updated description.</li> <li>Updated bit[14] to JTAG sysCONFIG Lock and added description.</li> <li>Updated bit[13] to JTAG ispTracy Lock and added description.</li> <li>Updated bit[12] to JTAG Boundary Scan Lock and added description.</li> </ul> </li> <li>In <a href="#">Table 6.9. Control Register 0</a>: <ul style="list-style-type: none"> <li>Updated description for TRAN_SEC_ENGINE.</li> </ul> </li> <li>In the <a href="#">CRAM_WRITE_ADDRESS</a> section: <ul style="list-style-type: none"> <li>In <a href="#">Table 6.18. CRAM_WRITE_ADDRESS Command</a>: <ul style="list-style-type: none"> <li>Corrected Byte2 value.</li> </ul> </li> </ul> </li> <li>Added the <a href="#">Self-Download Mode</a> section.</li> <li>Added the <a href="#">Flash Memory Space Partition</a> section.</li> </ul> |
| Software Selectable Options             | <ul style="list-style-type: none"> <li>In <a href="#">Table 7.3. MASTER_SPI_PORT Option</a>: <ul style="list-style-type: none"> <li>Added note on XSPI setting in relation to MSPI_RX_EDGE.</li> </ul> </li> <li>In <a href="#">Table 7.16. MSPI_RX_EDGE Option</a>: <ul style="list-style-type: none"> <li>Added note on use of default setting when MASTER_SPI_PORT = XSPI.</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Daisy Chaining                          | <ul style="list-style-type: none"> <li>Added description of the general daisy-chain configuration flow.</li> <li>In the <a href="#">Flow-Through Mode</a> section: <ul style="list-style-type: none"> <li>Added description of configuration data flow.</li> <li>Added description on the maximum MCLK frequency and optional buffer.</li> <li>Added <a href="#">Table 8.1. Required sysCONFIG Settings for Flow-Through Mode</a>.</li> <li>Updated <a href="#">Figure 8.1. Lattice Nexus 2 in Configuration Daisy Chain in Flow-Through Mode</a> by adding an optional buffer on the MCLKP line.</li> </ul> </li> <li>In the <a href="#">Bypass Mode</a> section: <ul style="list-style-type: none"> <li>Added description on the maximum MCLK frequency and optional buffer.</li> <li>Added <a href="#">Table 8.2. Required sysCONFIG Settings for Bypass Mode</a>.</li> <li>Updated <a href="#">Figure 8.2. Lattice Nexus 2 in Configuration Daisy Chain in Bypass Mode</a> by adding an optional buffer on the MCLKP line.</li> </ul> </li> <li>In the <a href="#">SCM Mode with External Host</a> section: <ul style="list-style-type: none"> <li>Added <a href="#">Table 8.3. Required sysCONFIG Settings for SCM Mode with External Host</a>.</li> </ul> </li> <li>Added the <a href="#">Merging Daisy-Chain Bitstreams</a> section.</li> </ul>                                                                                                                                                                                                                                                                                          |
| Appendix B. Nexus 2 Device Information  | <p>In <a href="#">Table B.1. Lattice Nexus 2 Device ID</a>:</p> <ul style="list-style-type: none"> <li>Updated logic capacity and 32-bit IDCODE for LN2-CT-20.</li> <li>Added LN2-CT and LN2-MH devices.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Appendix C. Modifying User OTP Settings | <ul style="list-style-type: none"> <li>In the <a href="#">Radiant Programmer Support for OTP Memory Access</a> section: <ul style="list-style-type: none"> <li>Updated description of the pseudo programming option in relation to shadow registers, refresh event, and power cycle.</li> </ul> </li> <li>Updated <a href="#">Figure C.1. Non-volatile Configuration Memory Access from Radiant Programmer</a>, <a href="#">Figure C.2. Feature Rows Programming Operations</a>, <a href="#">Figure C.3. Programming Feature Row</a>, <a href="#">Figure C.4. Updating Feature Row</a>, <a href="#">Figure C.5. Programming Non-volatile Control Register 1</a>, and <a href="#">Figure C.6. Control Register 1 Window</a>.</li> <li>In <a href="#">Table C.1. Feature Row Items</a>:</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

| Section                                                                  | Change Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                          | <ul style="list-style-type: none"> <li>Removed all reserved rows.</li> <li>Removed security-related settings. Security-related settings are documented in the Lattice Nexus 2 One-Time Programmable Security User Guide.</li> <li>Updated description for Boot_Offset[47:0].</li> <li>Removed Notes 1 through 4.</li> <li>In the <a href="#">Advanced Security Keys Programming</a> section: <ul style="list-style-type: none"> <li>Replaced content with a statement containing reference to the Lattice Nexus 2 One-Time Programmable Security User Guide.</li> <li>Added a note on device setup finalization.</li> </ul> </li> </ul> |
| Appendix D. Configuration Access from User Logic                         | <ul style="list-style-type: none"> <li>Added note to <a href="#">Table D.2. LMMI CFG Offset List</a> regarding write sequence of LMMI_CFG_PORT_REQUEST.</li> <li>In the <a href="#">LMMI CFG Command Execution</a> section: <ul style="list-style-type: none"> <li>Updated description on PROG_CNTRL0 command example.</li> <li>Updated <a href="#">Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode</a>.</li> <li>Updated description on READ_CNTRL0 command example.</li> <li>Updated <a href="#">Figure D.4. Waveform for LMMI Read Operation</a>.</li> </ul> </li> </ul>                                                |
| Appendix E. Flash Programming                                            | Added new section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Appendix F. JTAG Daisy-Chain Configuration with External Host Controller | Added new section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| References                                                               | <ul style="list-style-type: none"> <li>Added Mach-N2 web page.</li> <li>Added Lattice Nexus 2 One-Time Programmable Security User Guide.</li> <li>Added links to FAQ for security-related documents.</li> <li>Removed Reveal User Guide.</li> <li>Updated existing listing to Lattice Radiant Software web page.</li> </ul>                                                                                                                                                                                                                                                                                                             |

#### Revision 0.81, December 2024

| Section                        | Change Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| All                            | <ul style="list-style-type: none"> <li>Made editorial changes and modifications to cover Certus-N2 device.</li> <li>Changed instances of MSIO, SSIO, MCLK, and SCLK to MDQ, SDQ, MCLKP, and SCLKP.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Definition of Terms            | Updated description for Multi-Boot.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Configuration Details          | Updated description to reference target mode in relation to JTAG or target SPI port in the Configuration Ports Arbitration section.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Configuration Process and Flow | <ul style="list-style-type: none"> <li>Updated description to reference <i>target mode</i> in the Configuration section.</li> <li>Updated description to reference <i>Control Register 1 bit[2:0]</i> in relation to preamble timer in the Controller SPI Configuration section.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Device Configuration           | <ul style="list-style-type: none"> <li>In the Controller SPI Mode section: <ul style="list-style-type: none"> <li>Removed 33-Ohm damping resistor recommendation for higher MCLK frequencies with a fast slew rate.</li> <li>Updated Figure 6.2. Nexus 2 Controller SPI Port with SPI Flash.</li> </ul> </li> <li>In Table 6.1. Controller SPI Configuration Port Pins: <ul style="list-style-type: none"> <li>Updated function, direction, and description for MMOSI/MDQ0 and MMISO/MDQ1.</li> <li>Updated function for MDQ[2:3] and MDQ[4:7].</li> </ul> </li> <li>In Table 6.2. Target SPI Configuration Port Pins: <ul style="list-style-type: none"> <li>Updated function for SMOSI/SDQ0, SMISO/SDQ1, SDQ[2:3], and SDQ[4:7].</li> </ul> </li> <li>In the Immediate Action (IMM) Command Waveforms section: <ul style="list-style-type: none"> <li>Updated Figure 6.9. Immediate Action Command Waveforms.</li> </ul> </li> <li>In the Target SPI to Controller SPI Bridge section: <ul style="list-style-type: none"> <li>Updated Figure 6.11. Target SPI to Controller SPI Bridge Functional Diagram and Figure 6.12. Target SPI to Controller SPI Bridge Block Diagram.</li> </ul> </li> </ul> |

| Section                                          | Change Summary                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                  | <ul style="list-style-type: none"> <li>In the JTAG to Controller SPI Bridge section: <ul style="list-style-type: none"> <li>Updated Figure 6.13. JTAG to Controller SPI Bridge Block Diagram.</li> </ul> </li> <li>Updated name and description for Device State in Table 6.7. Status Register 1.</li> <li>Updated bits[3:0] to reserved in Table 6.11. User Mode Register.</li> <li>In the BITSTREAM_BURST section under the Target Configuration Command Details section: <ul style="list-style-type: none"> <li>Updated segmented bitstream burst mode steps.</li> <li>Updated Figure 6.18. Segmented Bitstream Burst Mode Timing Waveforms.</li> </ul> </li> <li>Updated SED_CRC size in Table 6.76. READ_SED_CRC Return Data.</li> <li>Added Table 6.79. READ_SED_CRC_CALC Return Data.</li> </ul> |
| Software Selectable Options                      | <ul style="list-style-type: none"> <li>Removed note on OTP in the SLAVE_SPI_PORT section.</li> <li>Updated XSPI description to reference xSPI mode in Table 7.2. SLAVE_SPI_PORT Option.</li> <li>Updated XSPI and XSPI_DIFF_CLK descriptions to reference xSPI mode in Table 7.3. MASTER_SPI_PORT Option.</li> <li>Updated description in the CONFIGIO_VOLTAGE_BANK1, CONFIGIO_VOLTAGE_BANK2, and MultiBoot Offset sections.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                 |
| Daisy Chaining                                   | Updated Figure 8.1. Lattice Nexus 2 in Configuration Daisy Chain in Flow-Through Mode, Figure 8.2. Lattice Nexus 2 in Configuration Daisy Chain in Bypass Mode, and Figure 8.3. Lattice Nexus 2 in Configuration Daisy Chain in SCM Mode with External Host.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| References                                       | <ul style="list-style-type: none"> <li>Added links for available references and rearranged some items.</li> <li>Removed unavailable links.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Appendix A. Nexus 2 Target Programming Guide     | Updated target SPI pin names in relation of the Lattice programming cable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Appendix C. Modifying User OTP Settings          | Added note on Secure Field Usage lifecycle stage to Table C.1. Feature Row Items.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Appendix D. Configuration Access from User Logic | <ul style="list-style-type: none"> <li>Removed precautionary statement regarding utilizing CONFIG_LMMIC to access configuration logic.</li> <li>Updated Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode title.</li> <li>In Table D.3. CONFIG_LMMIC Supported Commands: <ul style="list-style-type: none"> <li>Added commands MSPI_BRIDGE and MSPI_BRIDGE_CLOCK.</li> <li>Removed commands PORT_REQUEST and PORT_STATUS_READ.</li> <li>Added note regarding PORT_REQUEST and PORT_STATUS_READ.</li> </ul> </li> <li>Minor editorial changes.</li> </ul>                                                                                                                                                                                                                                     |

#### Revision 0.80, September 2024

| Section | Change Summary       |
|---------|----------------------|
| All     | Preliminary release. |



[www.latticesemi.com](http://www.latticesemi.com)