Lattice Avant sysCONFIG User Guide

Preliminary Technical Note

FPGA-TN-02299-0.88

January 2026

Disclaimers

Inclusive Language

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
AES	Advanced Encryption Standard
BBRAM	Battery Backed RAM
BSDL	Boundary Scan Description Language
BSE	Bitstream Engine
CID	Customer ID
CFG	Configuration
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)
DRC	Design Rule Checking
DTR	Dual Transfer Rate
EBR	Embedded Block RAM
ECDSA	Elliptic Curve Digital Signature Algorithm
FSM	Finite State Machine
GPIO	General Purpose I/O
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
PAR	Place and Route
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
SEC	Soft Error Correction
SED	Soft Error Detection
SER	Soft Error Rate
SFDP	Serial Flash Discoverable Parameters
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSPI	Target Serial Peripheral Interface
TAP	Test Access Port (JTAG)
TCK	Test Clock Pin
TDI	Test Data Input

Abbreviation	Definition
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 Avant™ FPGA platform is a low power mid-range general purpose FPGA platform optimized for a wide range of applications across multiple markets. It supports many new and unique features, thereby making a Lattice Avant FPGA one of the best-in-class FPGAs in its logic density range. Lattice Avant FPGAs (herein referred to as Avant FPGA or Avant 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 Avant FPGA is built using volatile SRAM; therefore, an external 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 Avant device when it powers up or anytime the device configuration needs to be updated. The Avant device provides a rich set of features for configuring the FPGA or programming the external 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 Avant 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.

Note: Waveforms presented in this document are for reference only; for detailed timing recommendations, refer to the Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

2. Features

The following are main programming and configuration features of Avant devices:

- Ultrafast I/O configuration for instant-on support
- Fast device configuration with controller SPI (x1, x2, x4); xSPI (x8, dual transfer rate)
- 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) in either configuration mode or user function mode
- User-selectable booting sequence
- Bitstream encryption – AES-256. For detailed information, refer to the Lattice Avant Configuration Security User Guide (FPGA-TN-02335).
- Bitstream authentication – ECDSA and RSA. For detailed information, refer to the Lattice Avant Configuration Security User Guide (FPGA-TN-02335).
- 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 flash or other memory device. It is a binary file and is programmed unmodified into the SPI flash 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's 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), for the purpose of checking the bitstream file integrity.
Dual-Boot	This feature allows the FPGA device to support two configuration images that reside in an SPI flash device. Whenever loading failure occurs with the primary image, the FPGA device searches for and loads the secondary image. Both images come from an off-chip non-volatile SPI 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 an external SPI flash memory.
Number Formats	The following nomenclature is used to denote the radix of numbers: • 0x: Numbers preceded by 0x are hexadecimal. • b (suffix): Numbers suffixed with b are binary. • All other numbers are decimal. **Note:** 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.
Ping-Pong Boot	This feature allows the FPGA device to utilize the jump table to select an image for booting without changing the location of the image in SPI flash.
Port	The physical connection used to perform programming and some configuration operations. Ports on the Avant device include JTAG, target SPI, and controller SPI.
Programming	The process used to alter the contents of the external configuration memory.
Refresh	The process of triggering a configuration data load operation. It is activated by PROGRAMN pin pulsing or REFRESH command execution (which emulates PROGRAMN pin pulsing).
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 Avant 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 non-volatile memory or transmitted to the device through a configuration port.

4.1. Bitstream and SPI Flash Sizes

Avant devices are SRAM-based FPGAs. The SRAM configuration memory must be loaded from an external non-volatile memory that can store all the configuration data. The size of the configuration data varies. It is dependent on the amount of logic resources available in the FPGA and the number of pre-initialized embedded block RAM (EBR) components. A design using the largest Avant 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)[1]	Recommended SPI Flash Size (Mb)	
			Single Boot	Dual Boot
LAV-AT-E/G/X30	No EBR	44	64	128
	Maximum EBR	64.1	128	256
LAV-AT-E/G/X50	No EBR	99.8	128	256
	Maximum EBR	131.4	256	512
LAV-AT-E/G/X70	No EBR	99.8	128	256
	Maximum EBR	149.4	256	512

Note:

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

4.2. Programming and Configuration Ports

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

Table 4.2. Avant 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

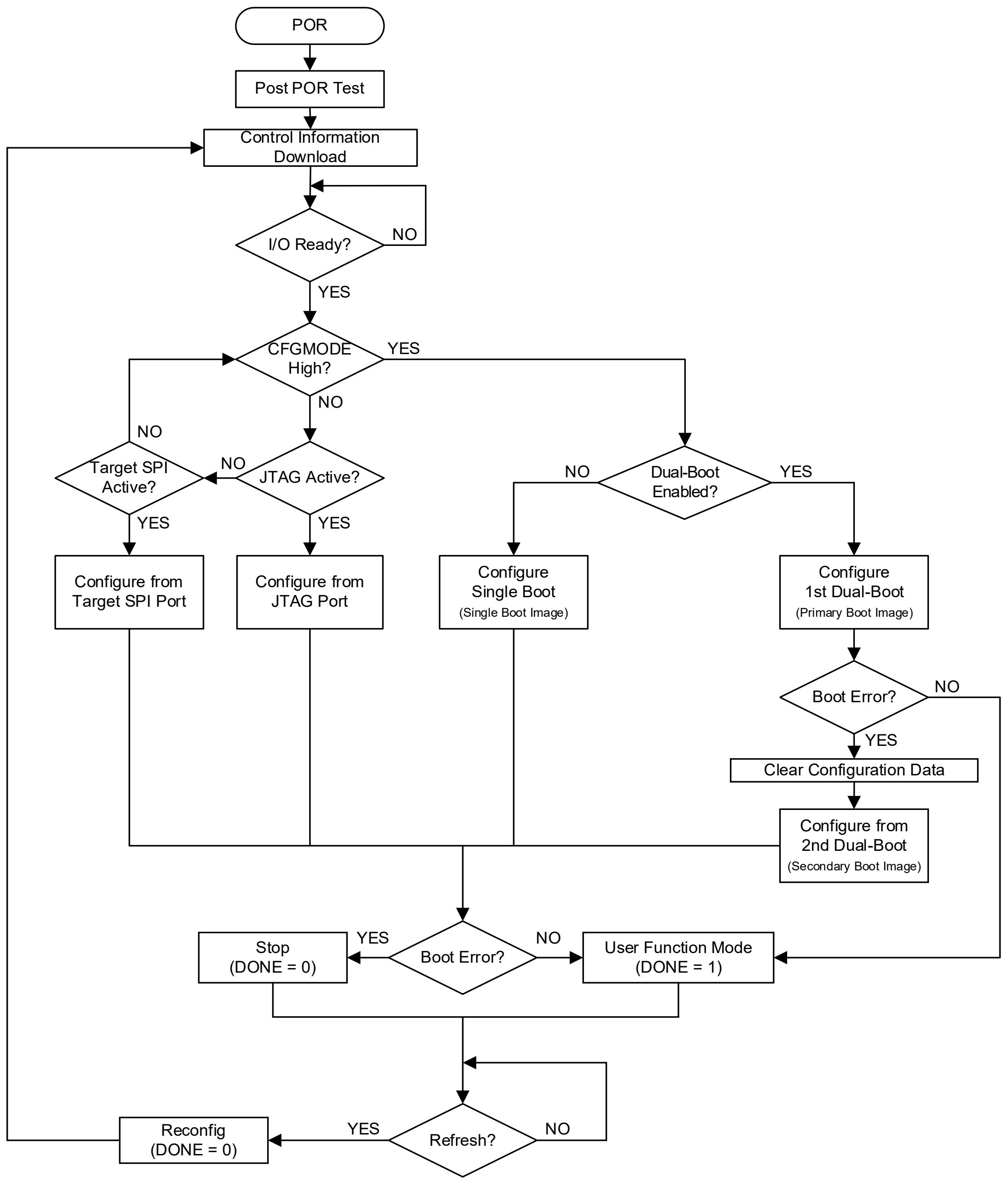
At power up, PROGRAMN pin pulsing, or REFRESH command execution, the configuration logic erases the user logic configuration data in the device and optionally erases EBR data depending on settings in Control Register 1 (CR1). After the erasure process completes, if the CFGMODE pin is high, the device enters controller SPI mode (automatic booting) and starts the configuration data download from an external SPI flash; 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 or target SPI mode. If the CFGMODE pin is toggled low during controller SPI mode booting, the boot process is aborted and the device switches to target SPI mode. The status register provides an indicator which 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 mode. To start the boot process over the controller SPI port, 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.



Note: Refresh is either PROGRAMN pulsing or REFRESH command execution.

Figure 4.1. Configuration Control Flow

4.4. sysCONFIG Pins

Avant 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.

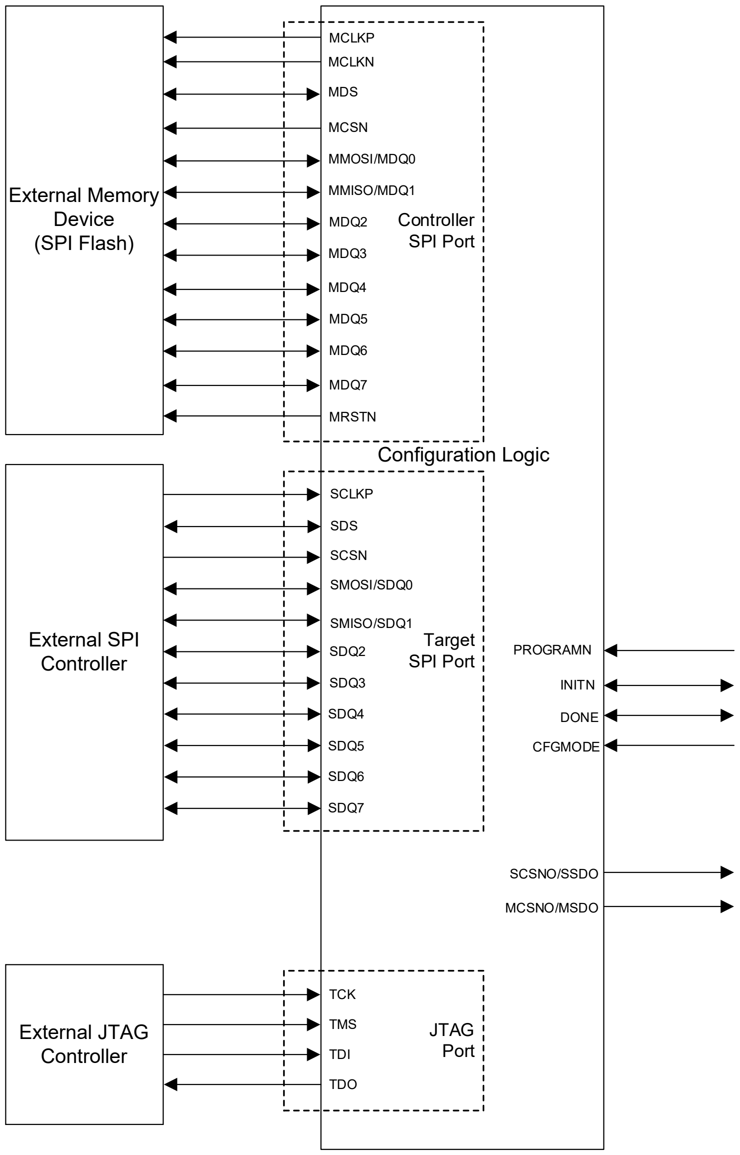


Figure 4.2. sysCONFIG Pins

When configuration of an Avant device is complete, the device enters user function mode. In user function mode, target SPI sysCONFIG pins will default to become general purpose I/O pins. This means that the target SPI port cannot be used to configure the Avant 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[1]	1’b0[1]
	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[2]	MCSNO/MSDO	GPIO	UP	MCSNO/MSDO	—	—
	SCSNO/SSDO	Bank 2	Shared[2]	MCSNO/MSDO	GPIO	UP	—	SCSNO/SSDO	—
Controller SPI	MCLKP	Bank 1	Shared[2]	MCLKP	GPIO	UP/DOWN[3]	MCLKP	—	—
	MCLKN	Bank 1	Shared[2]	MCLKN	GPIO	UP/DOWN[4]	MCLKN	—	—
	MCSN[5]	Bank 1	Shared[2]	MCSN	GPIO	UP	MCSN	—	—
	MMOSI/MDQ0	Bank 1	Shared[2]	MMOSI/MDQ0	GPIO	UP	MMOSI/MDQ0	—	—
	MMISO/MDQ1	Bank 1	Shared[2]	MMISO/MDQ1	GPIO	UP	MMISO/MDQ1	—	—
	MDQ[2:7]	Bank 1	Shared[2]	MDQ[2:7]	GPIO	UP	MDQ[2:7]	—	—
	MDS	Bank 1	Shared[2]	MDS	GPIO	DOWN	MDS	—	—
	MRSTN	Bank 1	Shared[2]	MRSTN	GPIO	UP	MRSTN	—	—
Target SPI	SCLKP	Bank 2	Shared[2]	SCLKP	GPIO	UP/DOWN[3]	—	SCLKP	—
	SCSN[6]	Bank 2	Shared[2]	SCSN	GPIO	UP	—	SCSN	—
	SMOSI/SDQ0	Bank 2	Shared[2]	SMOSI/SDQ0	GPIO	UP	—	SMOSI/SDQ0	—
	SMISO/SDQ1	Bank 2	Shared[2]	SMISO/SDQ1	GPIO	UP	—	SMISO/SDQ1	—
	SDQ[2:7]	Bank 2	Shared[2]	SDQ[2:7]	GPIO	UP	—	SDQ[2:7]	—
	SDS	Bank 2	Shared[2]	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; 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 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 $t_{PROGRAMN}$ for it to be recognized by the FPGA. This minimum time is defined in the sysCONFIG Port Timing Specifications of the Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

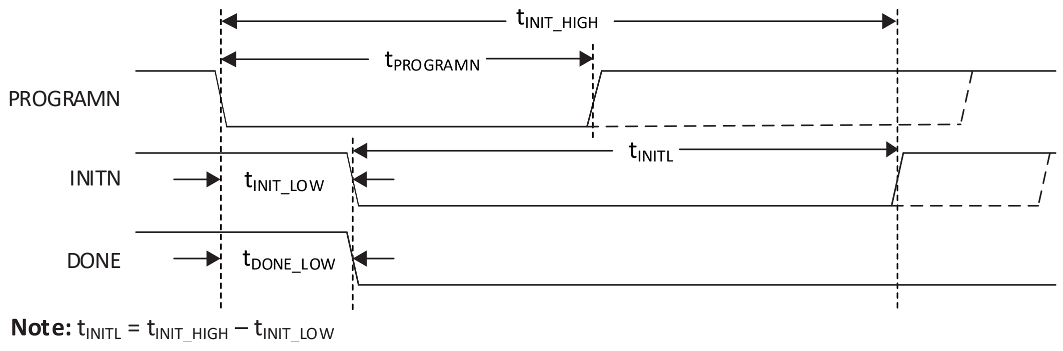


Figure 4.3. Configuration from PROGRAMN Timing

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

- Toggling the PROGRAMN pin during the initialization phase (t_{INITL} period) will 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_{INIT_LOW}, to indicate that the initialization phase is in progress. After the t_{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 memory device.

INITN can be asserted low by an external agent before the t_{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.

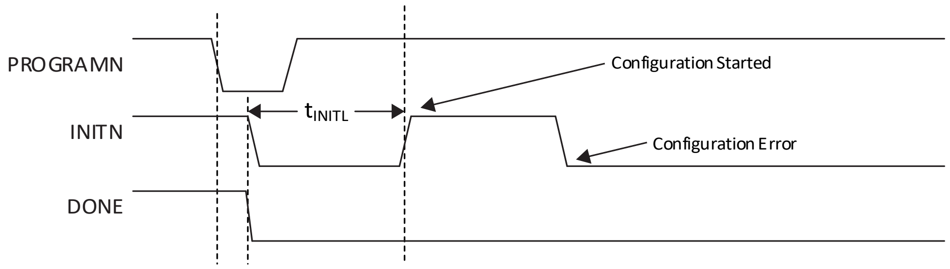


Figure 4.4. Configuration Error Notification

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 time out error is encountered when loading from the external memory device. This can occur when the device is in controller SPI configuration mode and the external 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 which 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 which 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 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 Avant Platform – Specifications Data Sheet (FPGA-DS-02112). MCLKP actively drives an external 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 memory device. Refer to the Controller SPI Mode section for details.

The Avant 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 to 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 Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

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 kΩ to 10 kΩ pull-up resistor to the MCSN pin on the Avant 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[2:7]

On the controller SPI port, MDQ[2:3] becomes the D[2:3] of the data bus in quad mode, or MDQ[2:7] becomes D[2:7] 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_{CCIO2} 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 Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

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 k Ω 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[2:7]

On the target SPI port, SDQ[2:3] becomes D[2:3] of the data bus in quad mode, or SDQ[2:7] becomes D[2:7] 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 the Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112). 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. Avant 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[2:3]	
	XSPI	MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[2:7], MDS	
	XSPI_DIFF_ CLK	MCLKP, MCLKN, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[2:7], 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[2:3]	
	XSPI	SCLKP, SCSN, SMOSI/SDQ0, SMISO/SDQ1, SDQ[2:7], SDS	

5. Configuration Process and Flow

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

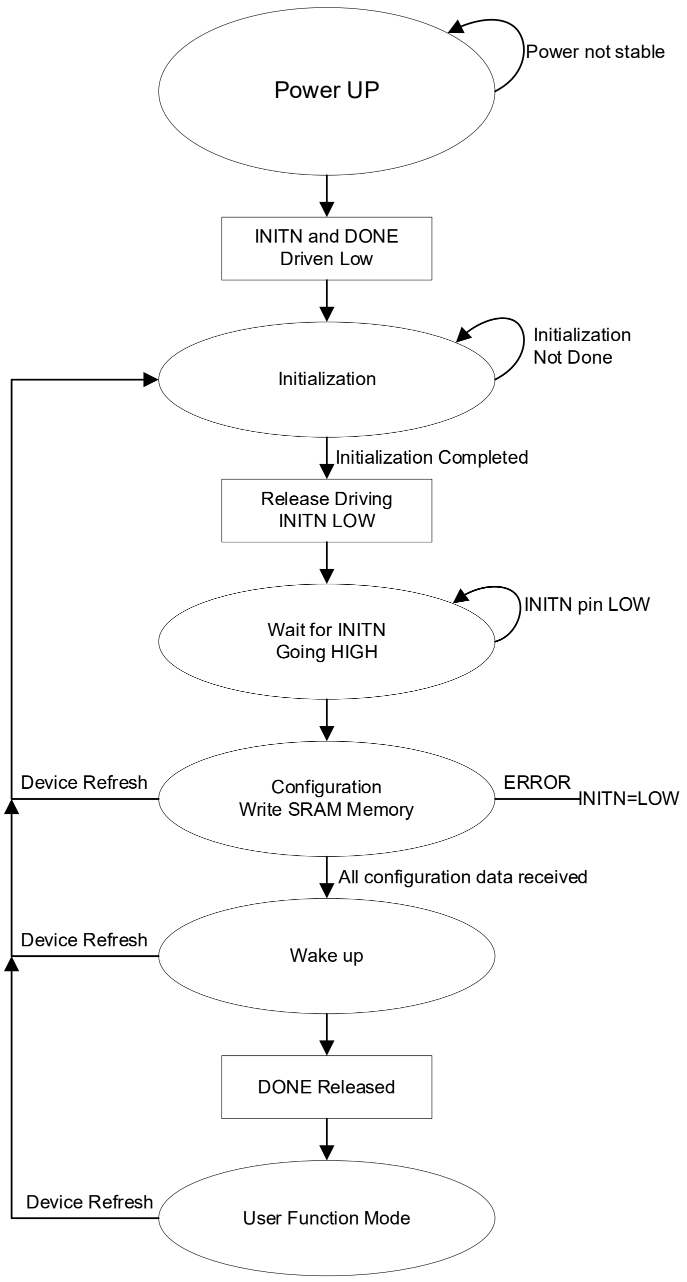


Figure 5.1. Configuration Flow

5.1. Power-Up Sequence

For the Avant 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 Avant device drives INITN and DONE low. When INITN and DONE are asserted low, the device enters the initialization phase.

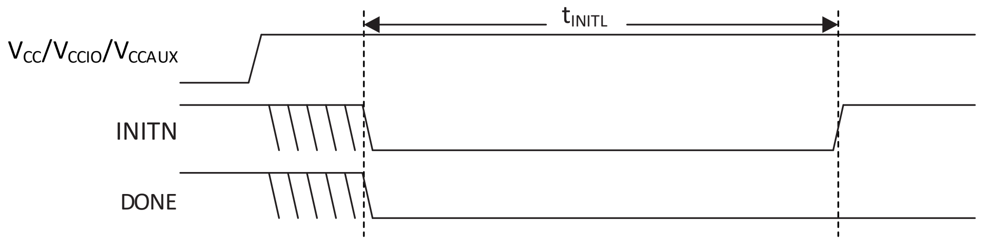


Figure 5.2. Configuration from POR Timing

5.2. Initialization

The Avant 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 Avant 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 Avant 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 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.

5.3.1. Controller SPI Configuration

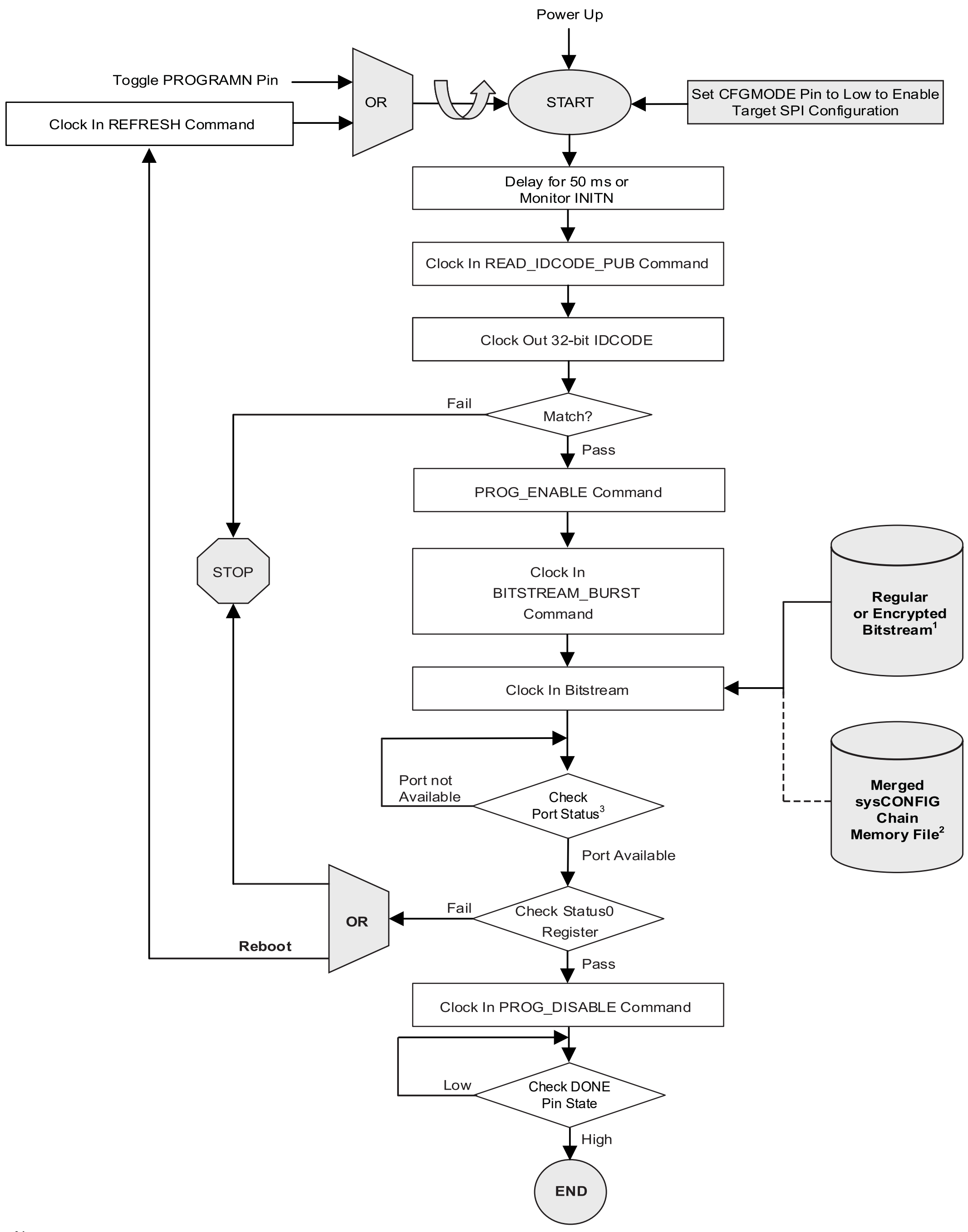
If CFGMODE is high, the Avant device enters the controller SPI mode and starts the external SPI flash boot process with signature verification. The device attempts a signature read-back in a finite loop until the expected result is received or the loop count is exceeded. A correct signature read allows 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 Avant 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 Avant 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 Avant device retries the SFDP/LSCC signature read until three consecutive matches are found. When successful, the Avant 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 Avant device continues fetching data from non-volatile memory to configure the FPGA SRAM configuration memory. The Avant 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 Avant 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

5.4. 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 which 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. 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 *GSR enabled* 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 Avant 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:

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

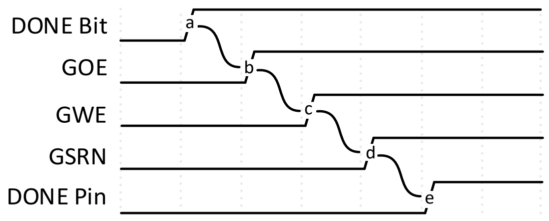


Figure 5.4. Wake-Up Sequence with DAISY_CHAIN_WAIT_DONE = DISABLED

If the Avant 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:

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

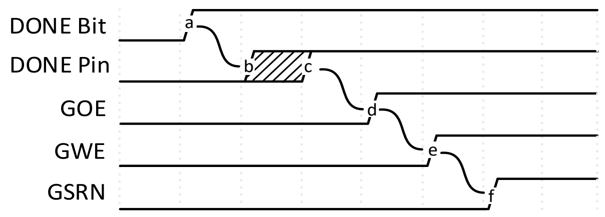


Figure 5.5. Wake-Up Sequence with DAISY_CHAIN_WAIT_DONE = ENABLED

5.5. Early I/O Release

The Avant 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 utilized as early I/O pins. If bitstream authentication and early I/O release are enabled in the Avant device, the user must provide an AES key in addition to an ECDSA or RSA key.

5.6. User Function Mode

The Avant 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 Avant device provides multiple options for loading configuration data into the SRAM configuration memory, either using external 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.

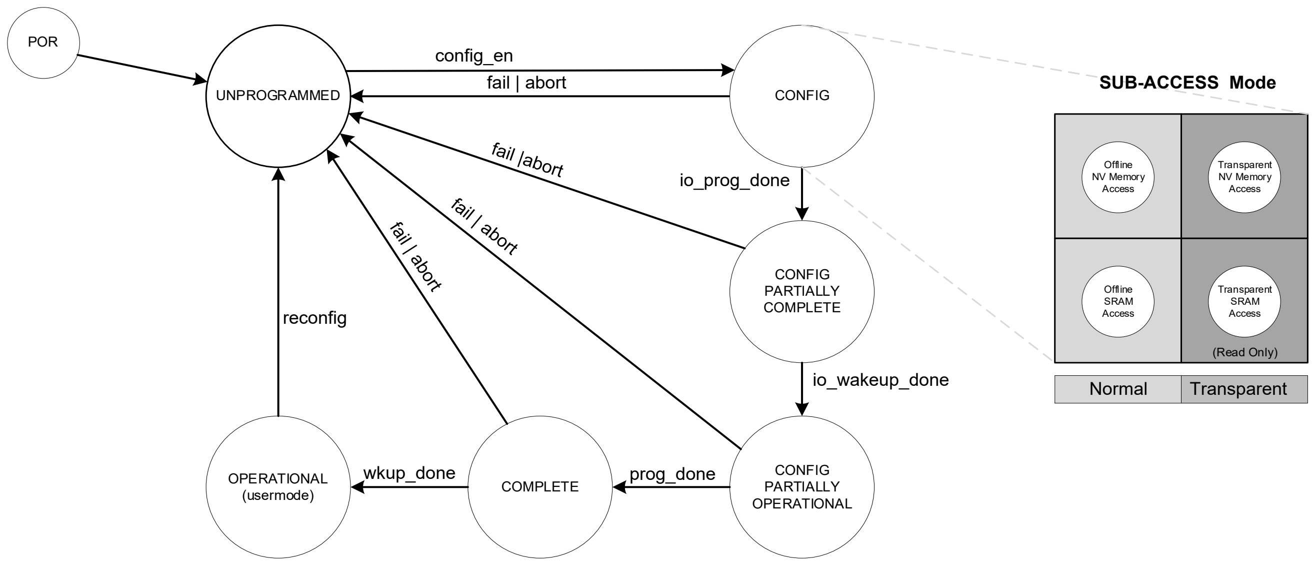


Figure 6.1. Configuration Modal States

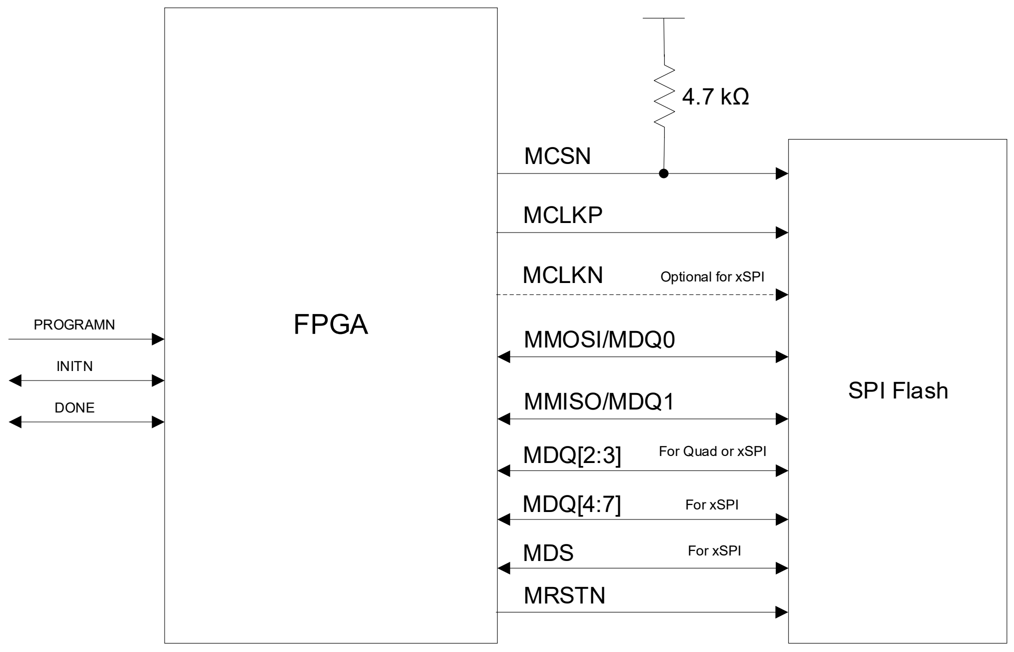
- Unprogrammed – In this state, the device may be blank or incompletely programmed. The device's system I/O pins are disabled (tri-stated) such that they cannot contend with output drivers that may exist in external devices connected to the device. Any program qualified command will have its operation nullified if executed in this state. Upon power-up, 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 will transition 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

In controller SPI mode, the Avant 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 Avant 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. Avant Controller SPI Port with SPI Flash

Once the SPI flash contains the configuration data, the configuration can be tested by either pulsing the PROGRAMN pin, transmitting a REFRESH command, or cycling power to the board. The Avant 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 Avant 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[1]	CSN	Output	Chip select used to enable an external SPI flash containing configuration data.
MMOSI/MDQ0	MOSI	Output	Carries output data from the Avant 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 Avant device configuration logic.
	D1	Input/Output	Input/Output pin for bitstream data, used in dual, quad, or xSPI mode.
MDQ[2:3]	D[2:3]	Input/Output	Input/Output pins for bitstream data from SPI flash, used only in quad or xSPI mode.
MDQ[4:7]	D[4:7]	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 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-Boot and Multi-Boot

Both the primary and secondary configuration images are stored in external SPI 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 SPI flash. Data is read from the external SPI 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 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 Avant device supports dual-boot with the controller SPI mode. If the primary image fails to load correctly, the device starts loading data from the golden sector in the SPI flash device. A blank external flash device 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 an SPI flash 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 (whose address is specified in user-programmable OTP; default is 0x000000). If the FPGA fails to configure, it automatically loads the golden boot file in the last 256 bytes of the SPI flash memory. 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 SPI flash fails to be programmed correctly.

Multi-boot, or the ability to dynamically reconfigure from multiple design bitstreams, is similar to dual-boot, where there is one primary (working) bitstream and one golden bitstream, but also supports up to 16 alternate 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 operation, refer to Lattice Avant Multi-Boot User Guide (FPGA-TN-02314).

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

6.2.5. Ping-Pong Boot

The ping-pong boot mode utilizes 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 SPI flash pointing to the golden boot image. Refer to Figure 6.3 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 SPI flash, and the primary bitstream is selected with the JUMP command in the jump table. To swap the order of booting, simply re-program these three instructions.

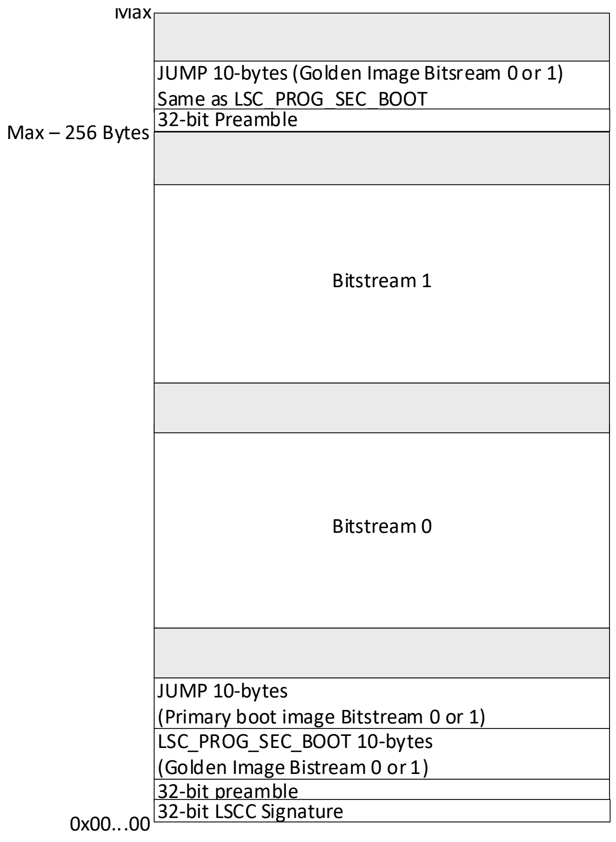

Max
JUMP 10-bytes (Golden Image Bitsream 0 or 1)
Same as LSC_PROG_SEC_BOOT
32-bit Preamble
Max – 256 Bytes
Bitstream 1
Bitstream 0
JUMP 10-bytes
(Primary boot image Bitstream 0 or 1)
LSC_PROG_SEC_BOOT 10-bytes
(Golden Image Bistream 0 or 1)
32-bit preamble
32-bit LSCC Signature
0x00…00

Figure 6.3. Jump Table

6.2.6. Dual, Quad, and xSPI Controller SPI Read Mode

The controller SPI mode in the Avant 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.

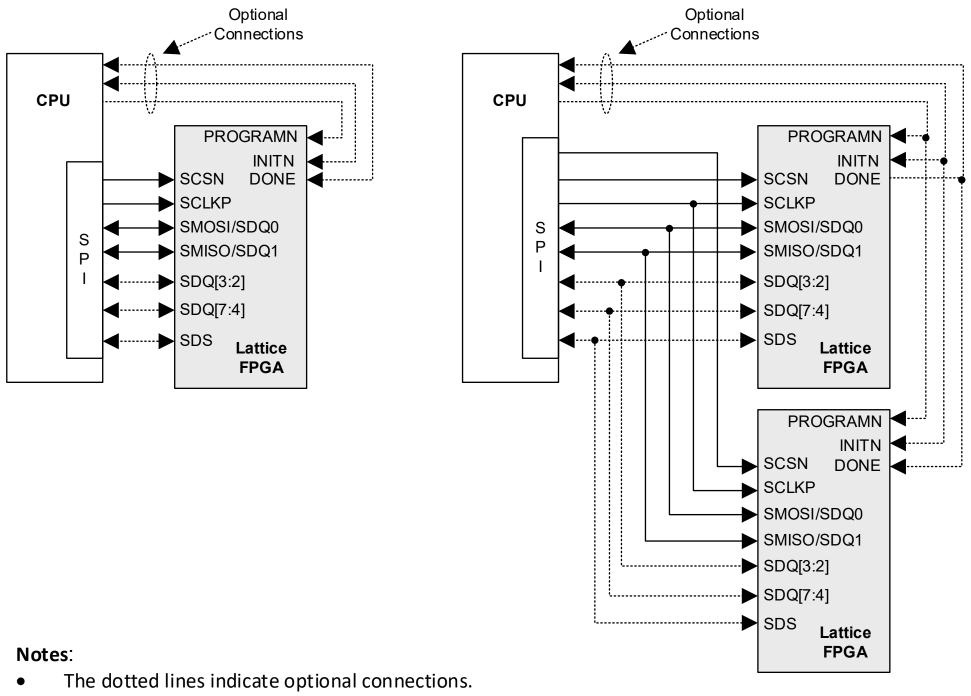
6.3. Target SPI Mode

The Avant 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 Avant 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 6.4. Avant 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 Avant device configuration logic.
SCSN[1]	CSN	Input with weak pull-up	Avant device configuration logic target SPI chip select input. CSN is an active low input. High to low transition – Starts transmitting a command. Low to high transition – Completes or terminates the current command.
SMOSI/SDQ0	MOSI	Input	Carries output data from the external SPI controller to the Avant 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 Avant 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[2:3]	D[2:3]	Input/Output	D[2:3] of the data bus for quad or xSPI mode.
SDQ[4:7]	D[4:7]	Input/Output	D[4:7] of the data bus for xSPI mode.
SDS	DS	Input/Output	xSPI data strobe

Note:
1. Use 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_{CCLK}	—	180	MHz

For detailed AC timing requirement for the Avant target SPI configuration port, refer to the sysCONFIG Port Timing Specifications of the Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

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 Avant 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 Avant 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.

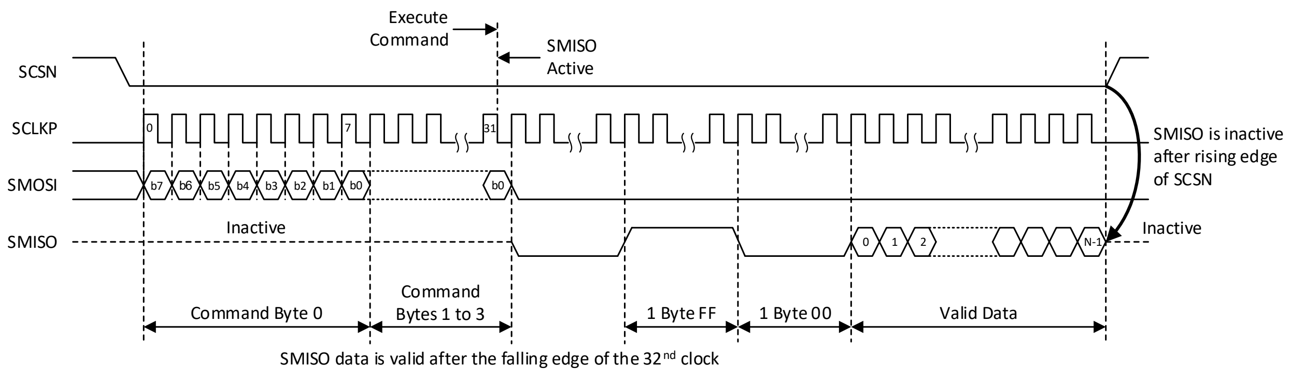


Figure 6.5. Read Command Waveforms

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 Avant 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.

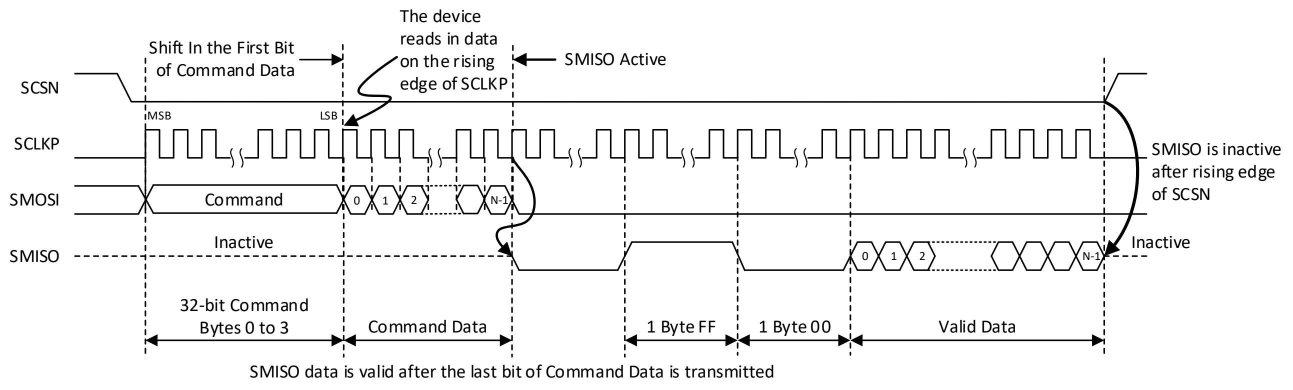

Shift In the First Bit of Command Data
The device reads in data on the rising edge of SCLKP
SMISO Active
SCSN
SCLKP
SMOSI
SMISO
MSB
LSB
Command
0
1
2
N-1
Inactive
0
1
2
N-1
SMISO is inactive after rising edge of SCSN
Inactive
32-bit Command Bytes 0 to 3
Command Data
1 Byte FF
1 Byte 00
Valid Data
SMISO data is valid after the last bit of Command Data is transmitted

Figure 6.6. 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.

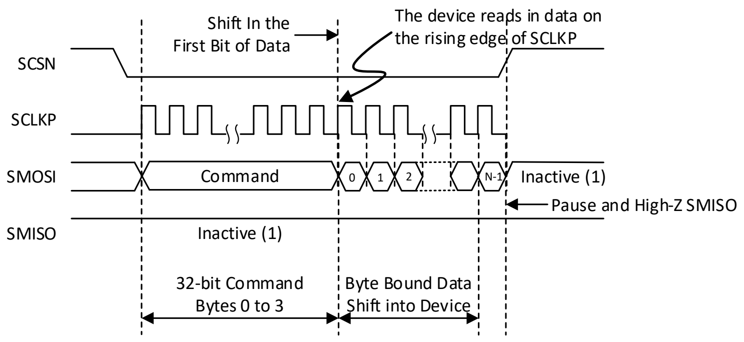

Shift In the First Bit of Data
The device reads in data on the rising edge of SCLKP
SCSN
SCLKP
SMOSI
SMISO
Command
0
1
2
N-1
Inactive (1)
Pause and High-Z SMISO
Inactive (1)
32-bit Command Bytes 0 to 3
Byte Bound Data Shift into Device

Figure 6.7. 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.

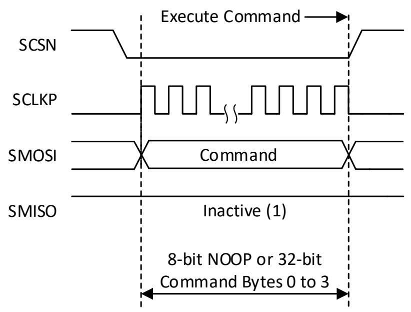


Figure 6.8. Immediate Action Command Waveforms

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.

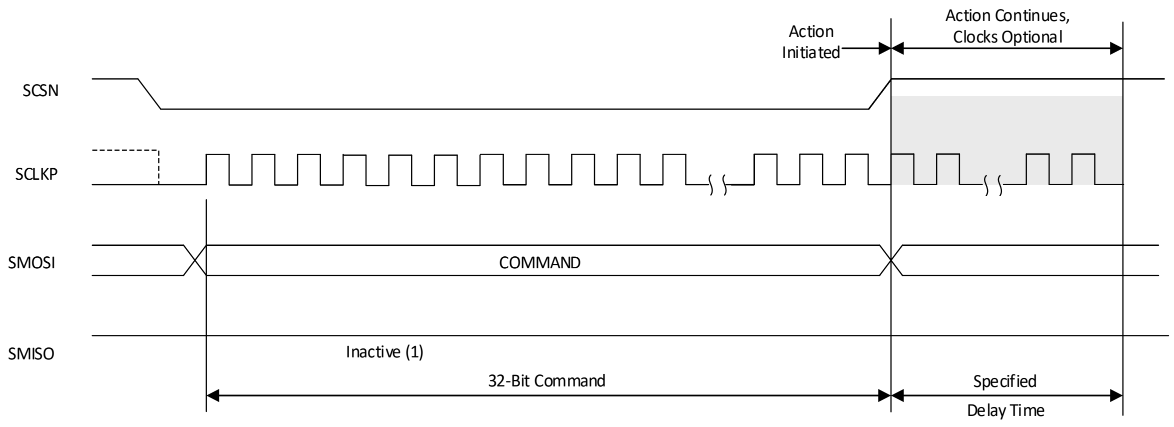


Figure 6.9. Delayed Action Command Waveforms

6.3.7. Target SPI to Controller SPI Bridge

As the controller SPI port and target SPI port on the Avant device utilize separate pins, the Avant device provides the target SPI to controller SPI bridge, which enables programming the external SPI flash through the Avant 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 6.10.

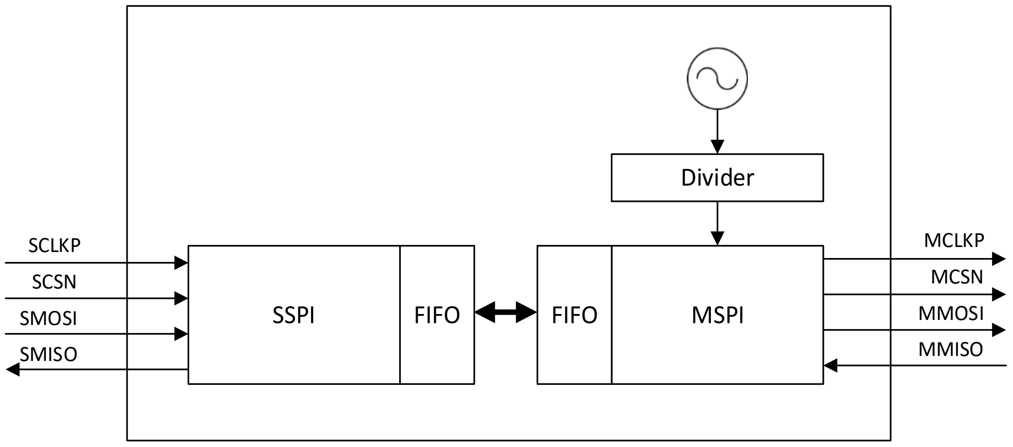


Figure 6.10. Target SPI to Controller SPI Bridge Functional Diagram

The default support for this bridging is serial data (x1) mode only. Figure 6.11. shows a block diagram example of target SPI to controller SPI bridge utilization.

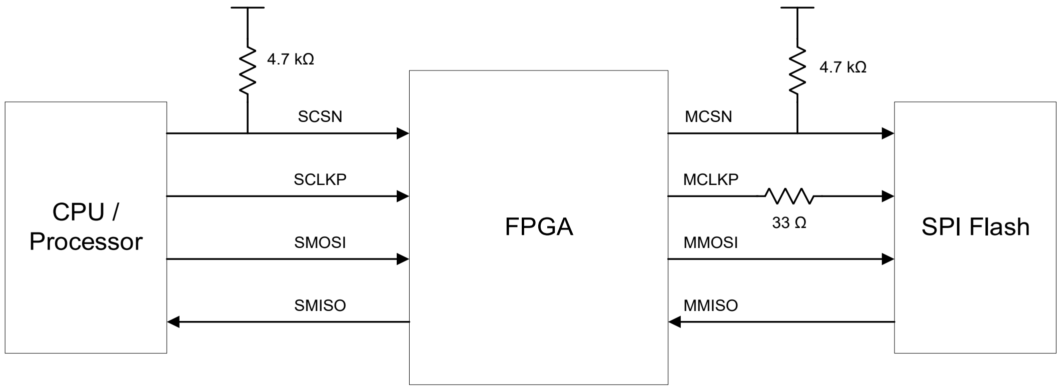


Figure 6.11. Target SPI to Controller SPI Bridge Block Diagram

6.4. JTAG Mode

The Avant 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 Avant devices provides:

- External flash memory programming in normal or transparent mode (background programming)
- Direct SRAM configuration
- Full access to the Avant 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 Avant device to be combined in a chain of other programmable logic.
- Reveal Debug – The Lattice Reveal debug tool is an embeddable logic analyzer tool that allows analysis of the user design implemented in an Avant 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 files for the Avant 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 Avant JTAG port, refer to the JTAG Port Timing Specifications of the Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

6.4.2. JTAG to Controller SPI Bridge

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

Note: In user function mode, to program the external SPI flash through the JTAG port, controller SPI port persistence must be enabled.

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 6.12 shows a block diagram example of the JTAG to controller SPI bridge utilization.

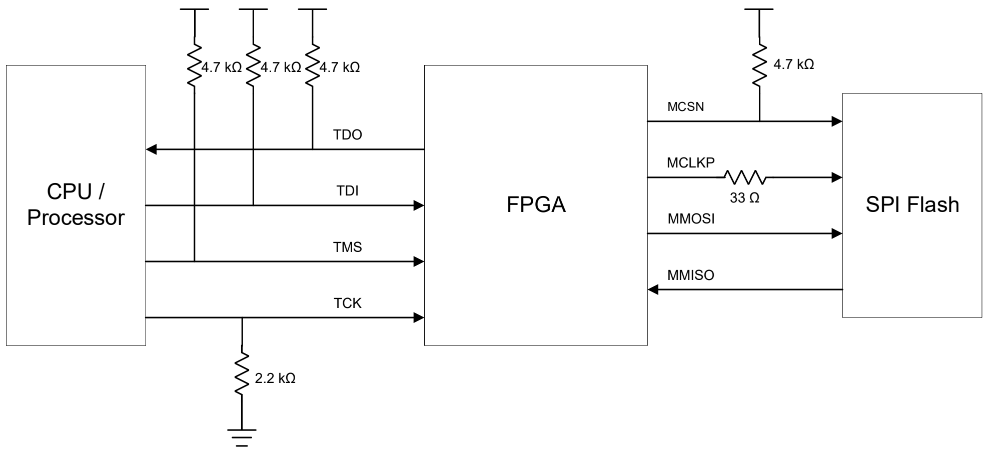


Figure 6.12. JTAG to Controller SPI Bridge Block Diagram

6.4.3. JTAG Reveal Support

Avant 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 Avant 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 which interacts with the Avant 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 for 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 PORT_REQUEST section for data register details.
PORT_STATUS_READ	0x7B	32	Y	Refer to the return data in the PORT_STATUS_READ 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 JTAG to Controller SPI Bridge 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 utilizes 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.

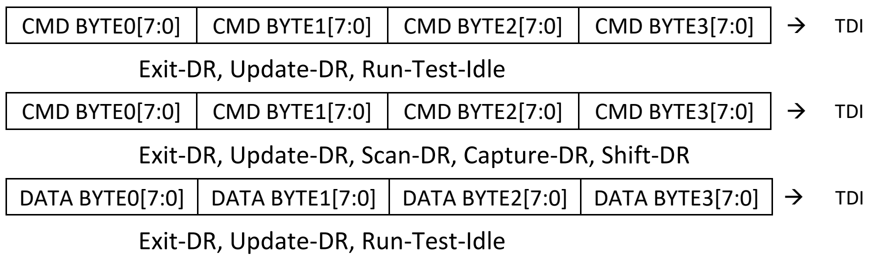


Figure 6.13. JTAG CONFIGURATION_DATA_SHIFT

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.

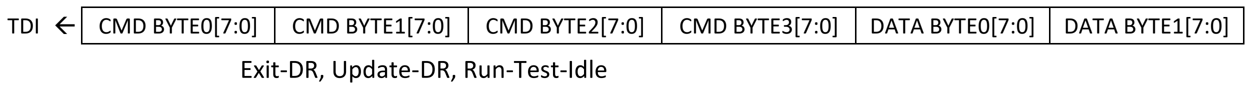


Figure 6.14. JTAG CONFIGURATION_BURST

6.5. 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.

6.5.1. 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 001 – Command error 0: Transaction ended before next command received 010 – Command error 1: Transaction did not end before next command received 011 – Invalid command error 100 – Blocked command received 101 – Reserved … 111 – Reserved
[28:25]	BSE Secondary Boot Error Code	0000 – No error has occurred. 0001 – There was a mismatch to the device ID code 0010 – There was an illegal command detected 0011 – There was a Frame CRC checksum error 0100 – No preamble before preamble timeout in controller SPI 0101 – Bitstream Engine execution was aborted by the user 0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed) 0111 – There was a CRC checksum error 1000 – Authentication Error 1001 – Authentication Setup Error 1010 – Reserved 1011 – RX FIFO Overflow Error 1100 – Global CRC Checksum Error 1101 – Command and SFDP Conflict 1110 – Security Error 1111 – Misc Error
[24:21]	BSE Primary Boot Error Code	0000 – No error has occurred. 0001 – There was a mismatch to the device ID code 0010 – There was an illegal command detected 0011 – There was a Frame CRC checksum error 0100 – No preamble before preamble timeout in controller SPI 0101 – Bitstream Engine execution was aborted by the user 0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed) 0111 – There was a CRC checksum error 1000 – Authentication Error 1001 – Authentication Setup Error 1010 – Reserved 1011 – RX FIFO Overflow Error 1100 – Global CRC Checksum Error 1101 – Command and SFDP Conflict 1110 – Security Error 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. 00 – None 01 – Failed due to error in bitstream 10 – Failed due to unstable V_{CCIO}
[18:17]	Preamble	Indicates what preamble was receive through the last bitstream segment. 00 – Plain 01 – Auth Header 10 – Authenticated 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. 0 – Low 1 – High
[9]	Auth DONE	Authentication is done. 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. 0 – Low 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 001 – Reserved 010 – Reserved 011 – Invalid Command error 100 – Blocked Command received 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.

6.5.2. 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. 0x0 – None 0x1 – AES256-GMAC 0x2 – Reserved 0x3 – Reserved 0x4 – ECDSA256 (digest = SHA2-256 or AES256-GCM1) 0x5 – ECDSA384 (digest = SHA2-384 or AES256-GCM1) 0x6 – ECDSA521 (digest = SHA2-512 or AES256-GCM1) 0x7 – Reserved … 0x9 – Reserved 0xA – RSA2048 (digest = SHA2-256 or AES256-GCM1) 0xB – RSA3072 (digest = SHA2-384 or AES256-GCM1) 0xC – RSA4096 (digest = SHA2-512 or AES256-GCM1) 0xD – Reserved … 0xF – Reserved
[19:18]	Device State	00 – Security engine off 01 – Security engine on 10 – Reserved 11 – Reserved
[17]	ATM Global Alarm	There was a system ATM alarm trigger event.
[16]	PUF_Enrolled	0 – PUF not enrolled 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 watch dog 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 0001 – There was a mismatch to the device ID code 0010 – There was an illegal command detected 0011 – There was a Frame CRC checksum error 0100 – No preamble before preamble timeout in controller SPI 0101 – Bitstream Engine execution was aborted by the user 0110 – Bitstream Engine last SRAM address programmed, but did not terminate (overprogrammed) 0111 – There was a CRC checksum error 1000 – Authentication Error 1001 – Authentication Setup Error 1010 – Bitstream Engine Timeout Error 1011 – RX FIFO Overflow Error 1100 – Global CRC checksum Error 1101 – Command and SFDP Conflict 1110 – Security Error 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.

6.5.3. 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 00 – Normal OTP 01 – Key erase in progress 10 – Key erase in progress 11 – Keys erased
[22:21]	BBRAM Status	Bit 0: Other Data Valid 1 – If the following are all true: • BBRAM initialized • Any of the following rows are non-zero: • Min bitstream version • User Data 0-7 • Monotonic counter 0-1 • Unused BBRAM words Bit 1: BS Keys Valid 1 – If the following are all true: • BBRAM initialized. • Bitstream AES Key is valid. • Bitstream Authentication Public Key hash is valid.
[20]	Fabric Lockdown Zeroization	0 – Fabric has not been zeroized due to lockdown 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 1 – FIPS mode enabled (from OTP feature setting)
[0]	BBRAM Valid	BBRAM has gone through initialization. 0 – Contents invalid 1 – Contents valid

6.6. 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 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.

6.6.1. 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[1]	0	Enable transparent CRAM programming
[24]	TRAN_INIT[1]	0	Enable transparent INIT bus programming
[23]	TRAN_SEC_ENGINE	0	Security engine mode 0 – Bitstream Security 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. 00 – None 01 – None 10 – Bypass Mode 11 – Flow-Through Mode
[16:15]	INTN OPT	0	Over-ride INITN. If CR0[16] is set to *1*, the DONE pin is overridden by CR0[15].
[14:13]	DONE OPT	0	Over-ride 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.

6.6.2. 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. 0 – 200 ms 1 – 100 ms 2 – 50 ms 3 – 40 ms 4 – 20 ms 5 – 1 ms 6 – 500 µs 7 – 100 µs
[26]	MSPI_CPHA	0	Controller SPI CPHA 0 – Data sampled at the leading (first) edge of the clock 1 – Data sampled at the trailing (second) edge of the clock
[25]	MSPI_CPOL	0	Controller SPI CPOL 0 – Active high clock. Clock is low in idle state. 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 0 – Data sampled at the leading (first) edge of the clock 1 – Data sampled at the trailing (second) edge of the clock
[20]	SSPI_CPOL	0	Target SPI CPOL 0 – Active high clock. Clock is low in idle state. 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 000 – SPI auto/SCM mode disabled 001 – SCM mode 010 – SPI auto x1 mode 011 – SPI auto x2 mode 100 – SPI auto x4 mode 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
			0 – Disabled (default) – bitstream sent in one continuous stream
			1 – 200 s
			2 – 100 s
			3 – 50 s
			4 – 25 s
			5 – 10 s
			6 – 5 s
			7 – 1 s
			8 – 750 ms
			9 – 500 ms
			10 – 250 ms
[2:0]	MSPI Preamble Timer Count	0	Controller SPI preamble timeout value
			0 – 200 ms
			1 – 100 ms
			2 – 50 ms
			3 – 40 ms
			4 – 20 ms
			5 – 1 ms
			6 – 500 µs
			7 – 100 µs

6.7. 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 1 – User gsrn is synchronous
[112]	mspi_addr_source_sel	0	BSE	Controller SPI address selection for multiple boot 0 – mc_mspi_addr[47:0] 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 watch dog timer count value
[31]	wdt_en	0	USERWDT	Use to enable the user watch dog timer
[30]	wdt_mode	0	USERWDT	Select watch dog timer mode 0 – Single shot 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. 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

6.8. Configuration Commands

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

6.8.1. Command Format

All commands except no operation (NOOP) consist of four mandatory bytes plus a variable number of optional data bytes which 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.

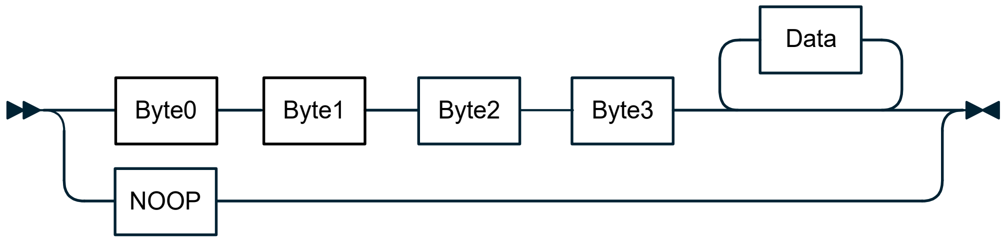


Figure 6.15. Configuration Command Format

6.8.2. Read Commands

Some commands read information from the device and output the result data back through the interface which sent the command. The number of bytes of result data is determined by the command which 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.

6.8.2.1. 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 0xFFFFFF00 and then captures the return data.
  - When return data is complete, the shift register captures 0xFFFFFFFF.
- Usage
  - Host monitors captured data until it sees 0xFFFFFF00, then samples return data.

6.8.2.2. 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 (DPSI/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.

6.8.3. 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[1]	Type[2]	Description
PROG_ENABLE	0x80	0x00	0x0c	0x00–0x02	—	—	—	IMM	Enable programming based on byte3 value: 0x00 – CRAM/INIT/OTP 0x01 – CRAM/INIT only 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. # data bytes = $\sum_{i=1}^{NFRAMES}(frame\_len_i + 3)$
CRAM_READ_INC	0x0a	Level, Zone	NFRAMES [15:8]	NFRAMES [7:0]	—	See note[3].	Y	Read	Read configuration memory data frames selected by the address register and post increment the address. Target SPI port only supports x1 mode for this command.
INIT_READ_REG	0x0f	0x00	NFRAMES [15:8]	NFRAMES [7:0]	4	See note[4]	Y	Data Read	Read INIT register data.
INIT_READ_EBR	0x0f	0x08	NFRAMES [15:8]	NFRAMES [7:0]	4	See note[5]	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[1]	Type[2]	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.

6.8.4. Target Configuration Command Details

6.8.4.1. 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

6.8.4.2. 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

6.8.4.3. 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 in 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.

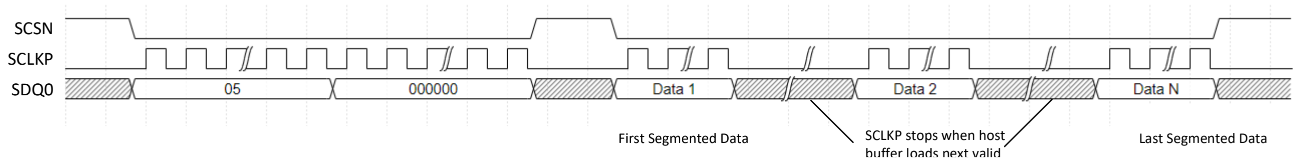


Figure 6.16. Segmented Bitstream Burst Mode Timing Waveforms

6.8.4.4. 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

6.8.4.5. 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	0x001	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

6.8.4.6. 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, Frame1 Data, CRC[15:8], CRC[7:0], 0x00, … 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

6.8.4.7. 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:

Readback Size = (Frame Byte Count) × 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

6.8.4.8. 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]

6.8.4.9. 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]

6.8.4.10. 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

6.8.4.11. 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]

6.8.4.12. 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]

6.8.4.13. 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]

6.8.4.14. 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]

6.8.4.15. 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]

6.8.4.16. 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]

6.8.4.17. 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]

6.8.4.18. 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]

6.8.4.19. 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]

6.8.4.20. 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]

6.8.4.21. 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

6.8.4.22. 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 will retain their current values. This instruction will not check for any errors.

New_CR0 = (Old_CR0 & MASK) | (DATA & ~MASK)

Example to change CR0[0] to 0:

Current Register Value	=	0x000000AB
Data	=	0x00000000
Mask	=	0xFFFFFFFE
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], DATA[7:0], MASK[31:24], MASK[23:16], MASK[15:8], MASK[7:0]

6.8.4.23. 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]

6.8.4.24. 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], DATA[7:0], MASK[31:24], MASK[23:16], MASK[15:8], MASK[7:0]

6.8.4.25. 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]

6.8.4.26. 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]

6.8.4.27. 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.

6.8.4.28. 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 00 – No dry-run 01 – Dry-run for primary bitstream 10 – Dry-run for secondary bitstream 11 – Target interface dry-run

6.8.4.29. REFRESH

The REFRESH command is equivalent to pulsing the PROGRAMN pin.

Table 6.65. REFRESH Command

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

6.8.4.30. 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 00 – x1 (SPI) 01 – x2 (DSPI) 10 – x4 (QSPI) 11 – x8 DDR (xSPI 8D-8D-8D)

6.8.4.31. 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 which uses a JTAG instruction (PROG_SPI).

Table 6.68. MSPI_BRIDGE Command

Byte0	Byte1	Byte2	Byte3	Data
0x16	0x00	0x00	0x00	DATA_BYTES[N]…

6.8.4.32. 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}) 00 – 450 MHz 01 – 360 MHz 10 – 300 MHz 11 – 240 MHz

6.8.4.33. 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 which 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 00 – Nowait request for exclusive access Request returns quickly with success/failure status 01 – Wait request for exclusive access 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 10 – Force request for exclusive access Always succeeds, even if another port previously had exclusive access 11 – Release exclusive access All ports are available/active

6.8.4.34. 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 0 – command mode 1 – bitstream mode
[2]	busy	PORT_REQUEST command is being processed
[1:0]	port_active	This bitfield is only valid when busy == 0 00 – AVAILABLE: This port is active but non-exclusive. Simultaneous access from multiple ports may cause conflicts and unexpected device behavior. 01 – EXCLUSIVE: This port is active and has exclusive control 10 – INACTIVE: This port is inactive and a different port is active 11 – DISABLED: This port is permanently inactive (for example, if a hard_lock OTP bit is set)

6.8.4.35. 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]

6.8.4.36. 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

6.8.4.37. 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]

6.8.4.38. 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]

6.8.4.39. 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

6.8.4.40. 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], DATA[7:0], MASK[31:24], MASK[23:16], MASK[15:8], MASK[7:0]

Table 6.86. PROG_OTP_SHADOW Operand

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

6.8.4.41. 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

6.8.4.42. 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

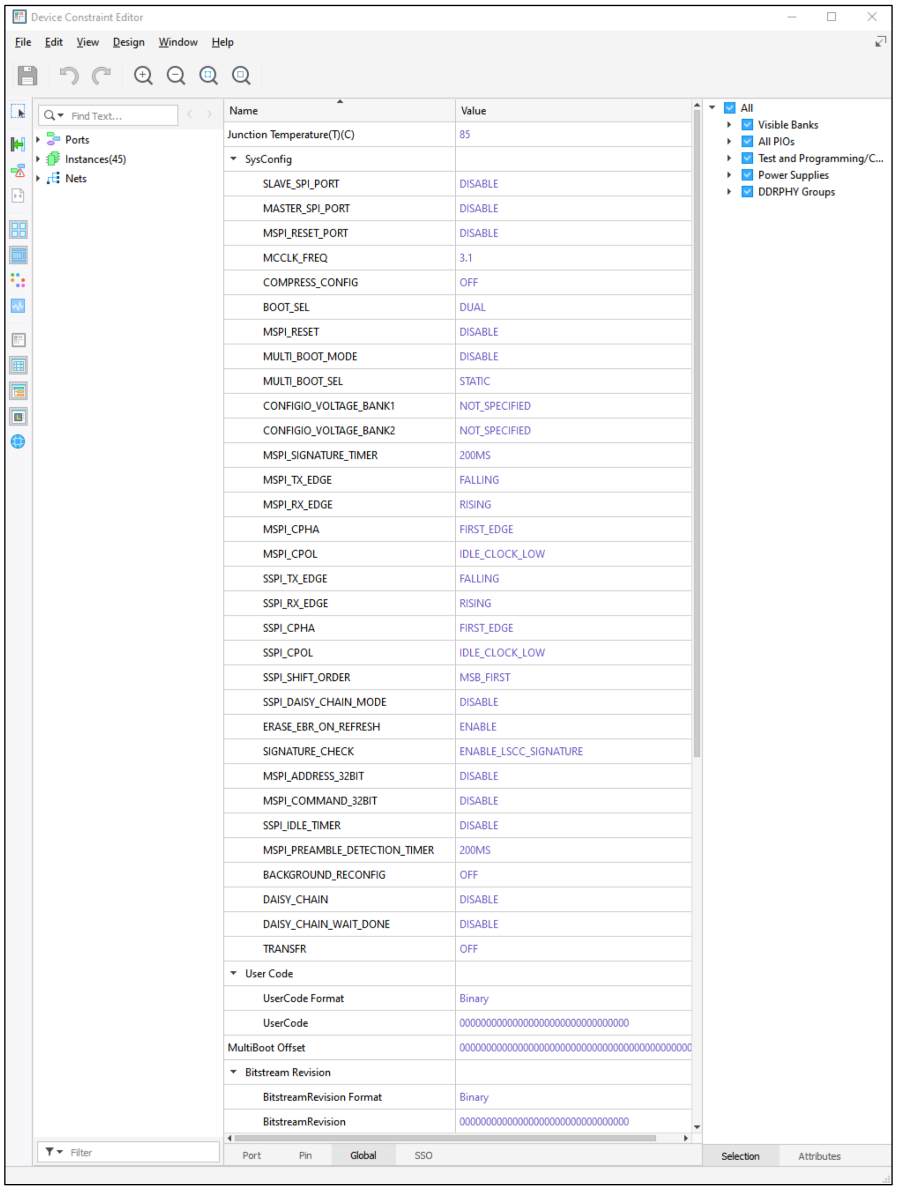
7. Software Selectable Options

The operation of the Lattice Avant device configuration logic is managed through options selected in the Device Constraint Editor of the Lattice Radiant design software. Lattice Avant devices use the built-in arbitration logic, as described in the Configuration Ports Arbitration section, to select the configuration port. Configuration logic preferences such as the persistence of the configuration port can be set through the Device Constraint Editor.

7.1. Accessing Software Selectable sysCONFIG Options

To access the software selectable sysCONFIG options, perform the following:

1. In the Lattice Radiant design software, open your design project file.
2. Open the Device Constraint Editor.
3. Click on the **Global** tab in the Device Constraint Editor.
4. Locate the sysCONFIG tree. Figure 7.1 shows sysCONFIG options in the Global page.



Name	Value
Junction Temperature(T)(C)	85
SysConfig	
SLAVE_SPI_PORT	DISABLE
MASTER_SPI_PORT	DISABLE
MSPI_RESET_PORT	DISABLE
MCCLK_FREQ	3.1
COMPRESS_CONFIG	OFF
BOOT_SEL	DUAL
MSPI_RESET	DISABLE
MULTI_BOOT_MODE	DISABLE
MULTI_BOOT_SEL	STATIC
CONFIGIO_VOLTAGE_BANK1	NOT_SPECIFIED
CONFIGIO_VOLTAGE_BANK2	NOT_SPECIFIED
MSPI_SIGNATURE_TIMER	200MS
MSPI_TX_EDGE	FALLING
MSPI_RX_EDGE	RISING
MSPI_CPHA	FIRST_EDGE
MSPI_CPOL	IDLE_CLOCK_LOW
SSPI_TX_EDGE	FALLING
SSPI_RX_EDGE	RISING
SSPI_CPHA	FIRST_EDGE
SSPI_CPOL	IDLE_CLOCK_LOW
SSPI_SHIFT_ORDER	MSB_FIRST
SSPI_DAISY_CHAIN_MODE	DISABLE
ERASE_EBR_ON_REFRESH	ENABLE
SIGNATURE_CHECK	ENABLE_LSCC_SIGNATURE
MSPI_ADDRESS_32BIT	DISABLE
MSPI_COMMAND_32BIT	DISABLE
SSPI_IDLE_TIMER	DISABLE
MSPI_PREAMBLE_DETECTION_TIMER	200MS
BACKGROUND_RECONFIG	OFF
DAISY_CHAIN	DISABLE
DAISY_CHAIN_WAIT_DONE	DISABLE
TRANSFR	OFF
User Code	
UserCode Format	Binary
UserCode	00000000000000000000000000000000
MultiBoot Offset	00000000000000000000000000000000000000000000
Bitstream Revision	
BitstreamRevision Format	Binary
BitstreamRevision	00000000000000000000000000000000

Figure 7.1. sysCONFIG Options in Global Page of Device Constraint Editor

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[1]
MASTER_SPI_PORT	DISABLE	DISABLE, SERIAL, DUAL, QUAD, XSPI, XSPI_DIFF_CLK[1]
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[1]
BOOT_SEL	DUAL	DUAL, SINGLE
MSPI_RESET	DISABLE	DISABLE, ENABLE[1]
MULTI_BOOT_MODE	DISABLE	DISABLE, ENABLE[1]
MULTI_BOOT_SEL	STATIC	STATIC, DYNAMIC[1]
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[1]
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[1]
SSPI_SHIFT_ORDER	MSB_FIRST	MSB_FIRST, LSB_FIRST[1]
SSPI_DAISY_CHAIN_MODE	DISABLE	DISABLE, SCM[1], AUTO[1]
ERASE_EBR_ON_REFRESH	DISABLE	DISABLE, ENABLE[1]
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[1], ON[1]
DAISY_CHAIN	DISABLE	DISABLE, BYPASS[1], FLOW_THROUGH[1]
DAISY_CHAIN_WAIT_DONE	DISABLE	DISABLE, ENABLE[1]
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 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[2:3]) 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[2:7], 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[2:7]) 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[2:3]) in quad mode when the device is in user function mode. This preference also prevents over-assigning I/O functions to these pins.
	XSPI[1]	Persist the SPI port I/O pins (MCLKP, MCSN, MMOSI/MDQ0, MMISO/MDQ1, MDQ [2:7], 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[2:7], 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 if 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 flash 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 ±15%) to begin retrieving data from the external SPI flash. 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 SPI flash 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 MultiBoot Offset 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 bank1 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 *MS* 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 *MS* for milliseconds and *US* 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).

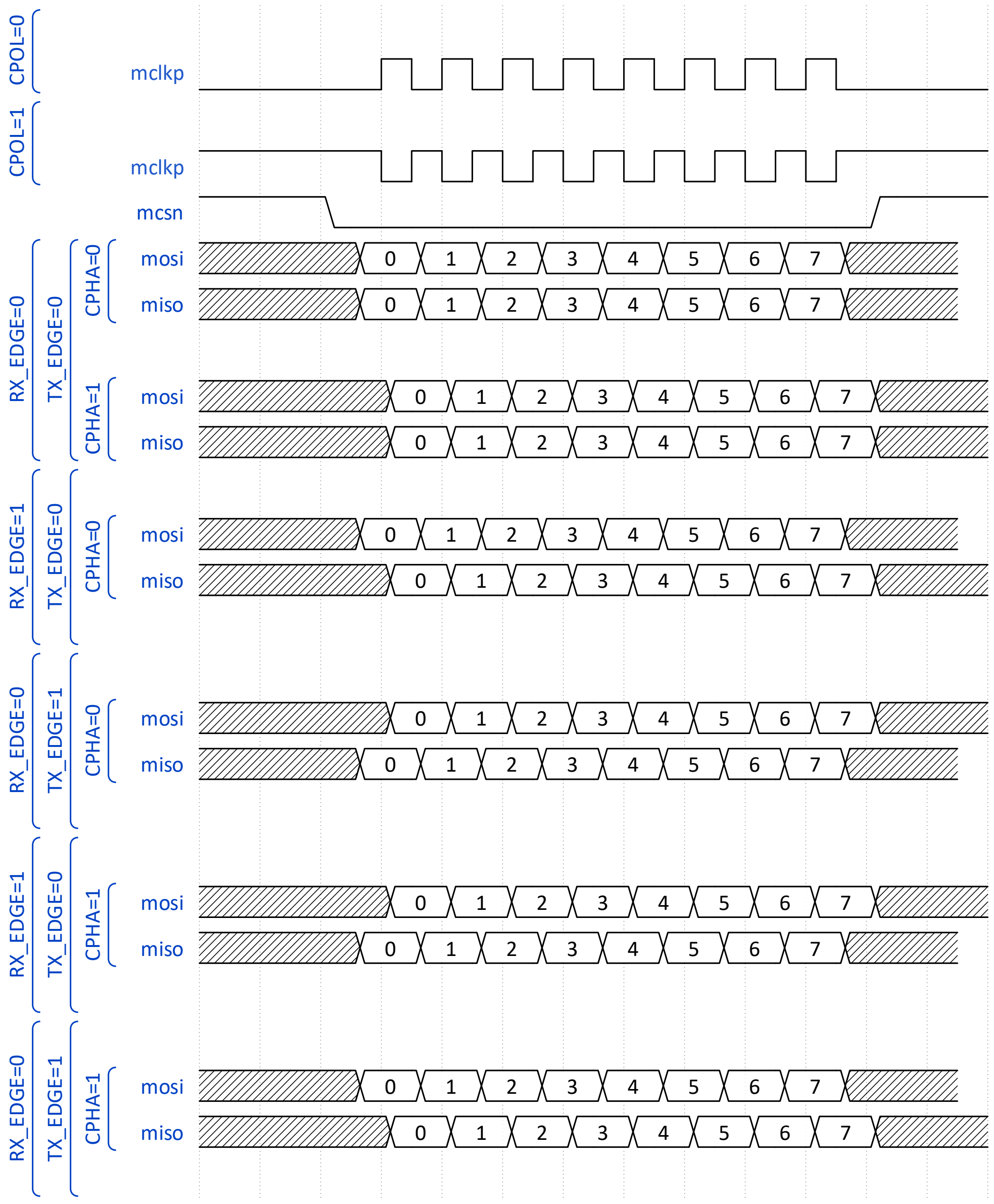
Table 7.14. MSPI_TX_EDGE Option

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)[1]	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 transmitting and receiving 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 transmitting and receiving 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 *MS* 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 *MS* for milliseconds and *US* 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 Table 6.12 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. The data format is set using the UserCode Format option. 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 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 flash memory to be stored in a 48-bit register in the FPGA device.

7.2.36. BitstreamRevision Format

The BitstreamRevision Format 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 utilize the same device and use the same 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 storage device more efficiently, several different FPGA bitstreams from various devices and designs can share a single configuration mechanism by using a daisy chain method. The Avant 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 Avant device supports flow-through mode, bypass mode, and SCM mode with external host.

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.

An example of the Lattice Avant 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.

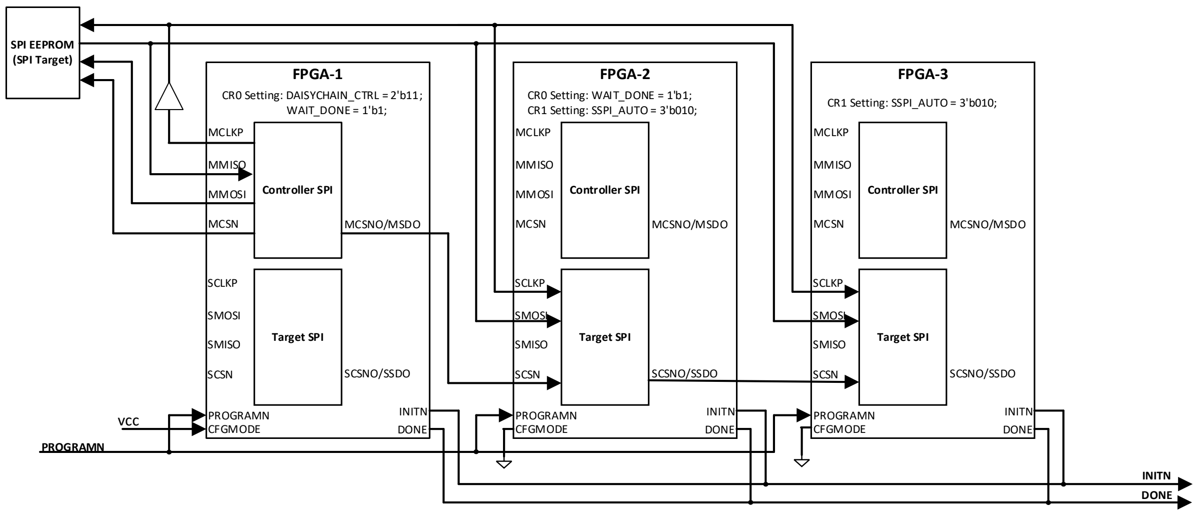


Figure 8.1. Lattice Avant in Configuration Daisy Chain in Flow-Through Mode

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 Avant 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.

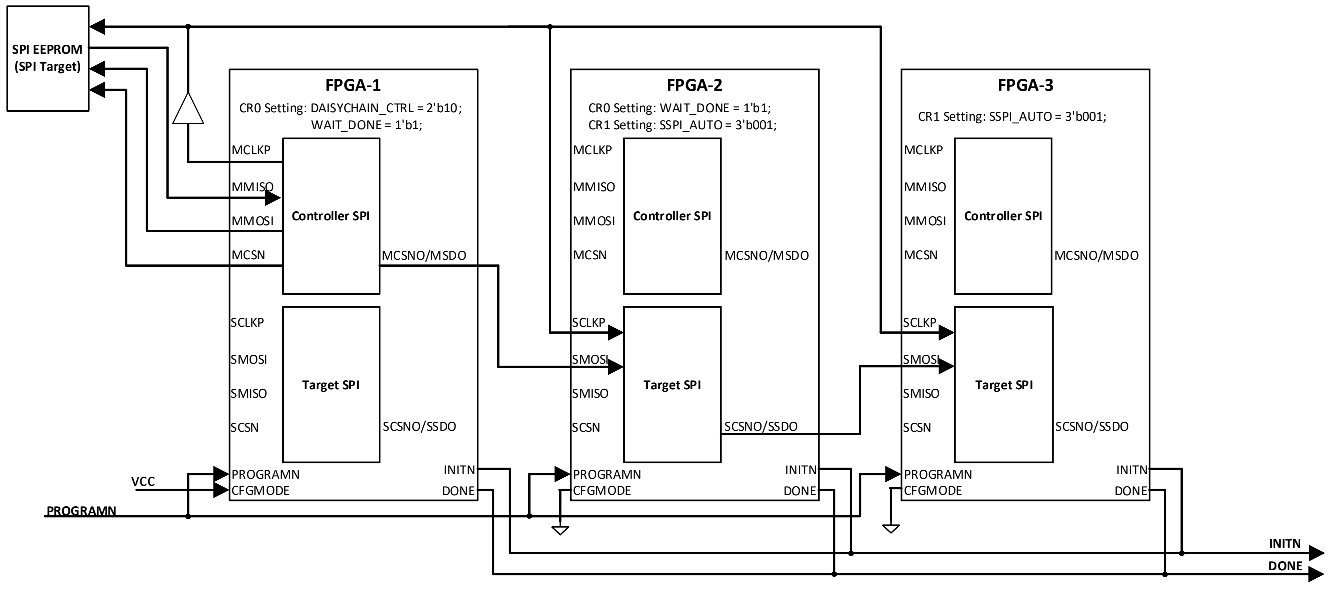


Figure 8.2. Lattice Avant in Configuration Daisy Chain in Bypass Mode

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.

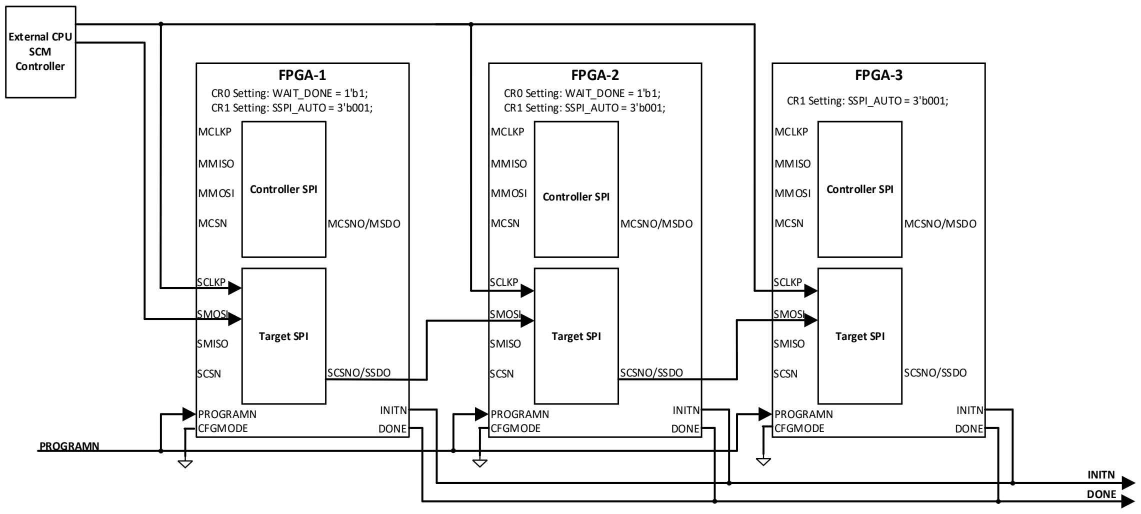

External CPU SCM Controller
FPGA-1
CR0 Setting: WAIT_DONE = 1'b1; CR1 Setting: SSPI_AUTO = 3'b001;
FPGA-2
CR0 Setting: WAIT_DONE = 1'b1; CR1 Setting: SSPI_AUTO = 3'b001;
FPGA-3
CR1 Setting: SSPI_AUTO = 3'b001;
MCLKP
MMISO
MMOSI
MCSN
Controller SPI
MCSNO/MSDO
SCLKP
SMOSI
SMISO
SCSN
Target SPI
SCSNO/SSDO
PROGRAMN
CFGMODE
INITN
DONE
PROGRAMN
INITN
DONE

Figure 8.3. Lattice Avant in Configuration Daisy Chain in SCM Mode with External Host

Appendix A. Avant Target SPI Programming Guide

The target SPI port of the Lattice Avant 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 the user 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. Avant Device Information

B.1. Device ID

Table B.1. Lattice Avant Device ID

Device Family Code	Product Name	Logic Capacity	32-bit IDCODE
LAV-AT	E70	637k System Logic Cells	0x710A4043
	E50	409k System Logic Cells	0x310A3043
	E30	262k System Logic Cells	0x310A2043
	G70	637k System Logic Cells	0x310A4043
	G50	409k System Logic Cells	0x110A3043
	G30	262k System Logic Cells	0x110A2043
	X70	637k System Logic Cells	0x110A4043
	X50	409k System Logic Cells	0x010A3043
	X30	262k System Logic Cells	0x010A2043

B.2. Configuration Memory

Table B.2. Lattice Avant CRAM Frame Sizes

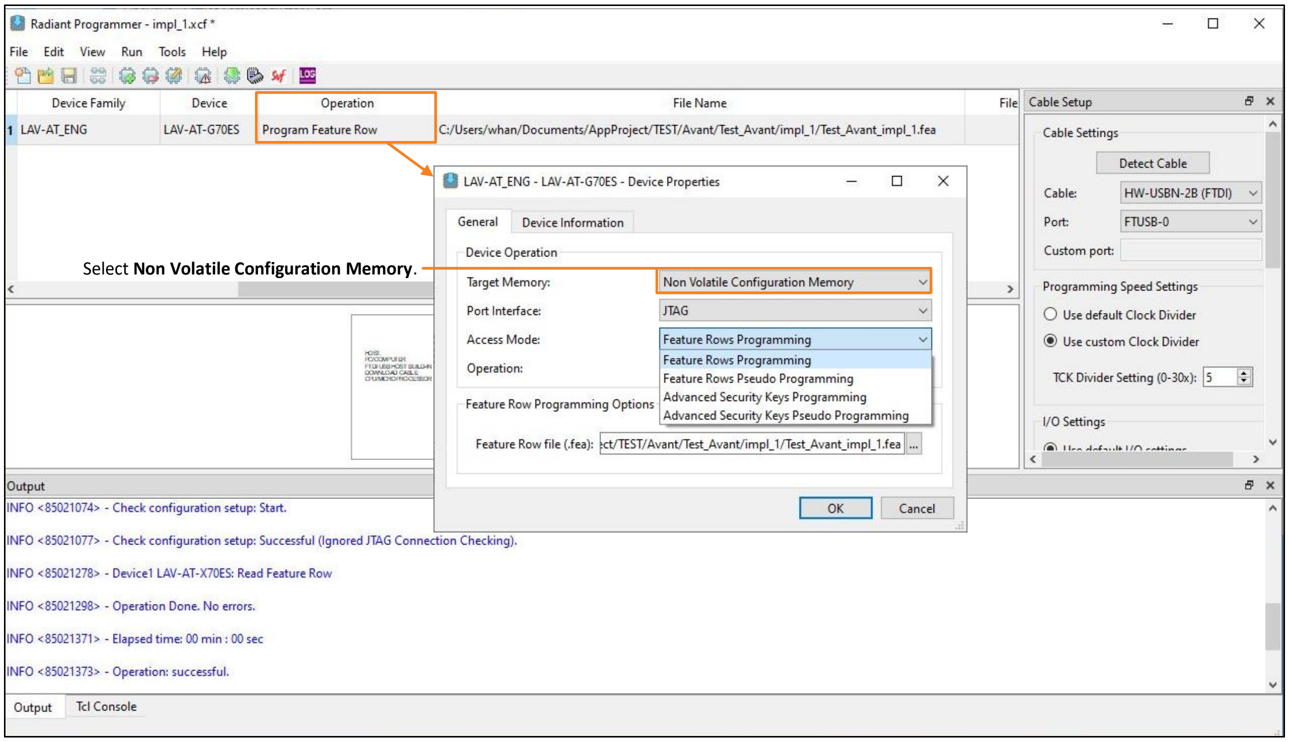
Device Family Code	Product Name	Number of CRAM Levels	Number of CRAM Zones	CRAM Frame Length, Level 0 (bytes)	CRAM Frame Length, Level 1 (bytes)
LAV-AT	E70	2	9	86	92
	E50	2	7	70	78
	E30	2	5	62	70
	G70	2	9	86	92
	G50	2	7	70	78
	G30	2	5	62	70
	X70	2	9	86	92
	X50	2	7	70	78
	X30	2	5	62	70

Appendix C. Modifying User OTP Settings

The Avant device user OTP memory (also known as the feature row) contain design-specific information for Avant 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 updated with the corresponding non-volatile memory data during a refresh event.


Radiant Programmer - impl_1.xcf *
File Edit View Run Tools Help
Cable Setup
Detect Cable
Select **Non Volatile Configuration Memory**.
LAV-AT_ENG - LAV-AT-G70ES - Device Properties
General
Device Information
Device Operation
OK
Cancel
Output
Tcl Console

Device Family	Device	Operation	File Name
LAV-AT_ENG	LAV-AT-G70ES	Program Feature Row	C:/Users/whan/Documents/AppProject/TEST/Avant/Test_Avant/impl_1/Test_Avant_impl_1.fea

Cable Settings
Cable: HW-USBN-2B (FTDI)
Port: FTUSB-0
Custom port:
Programming Speed Settings
( ) Use default Clock Divider
(●) Use custom Clock Divider
TCK Divider Setting (0-30x): 5
I/O Settings

Target Memory: Non Volatile Configuration Memory
Port Interface: JTAG
Access Mode: Feature Rows Programming
Operation:
- Feature Rows Programming
- Feature Rows Pseudo Programming
- Advanced Security Keys Programming
- Advanced Security Keys Pseudo Programming

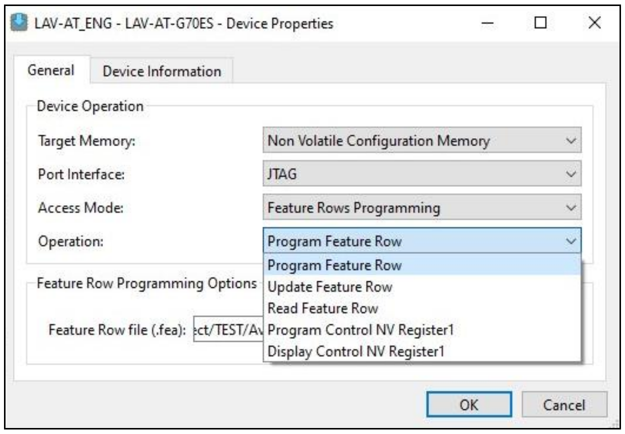
Feature Row Programming Options
Feature Row file (.fea): ct/TEST/Avant/Test_Avant/impl_1/Test_Avant_impl_1.fea

```
INFO <85021074> - Check configuration setup: Start.
INFO <85021077> - Check configuration setup: Successful (Ignored JTAG Connection Checking).
INFO <85021278> - Device1 LAV-AT-X70ES: Read Feature Row
INFO <85021298> - Operation Done. No errors.
INFO <85021371> - Elapsed time: 00 min : 00 sec
INFO <85021373> - Operation: successful.
```

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.


LAV-AT_ENG - LAV-AT-G70ES - Device Properties
General
Device Information
Device Operation
Feature Row Programming Options
Program Feature Row
Update Feature Row
Read Feature Row
Program Control NV Register1
Display Control NV Register1
OK
Cancel

Target Memory: Non Volatile Configuration Memory
Port Interface: JTAG
Access Mode: Feature Rows Programming
Operation: Program Feature Row
Feature Row file (.fea): ɛct/TEST/Av

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

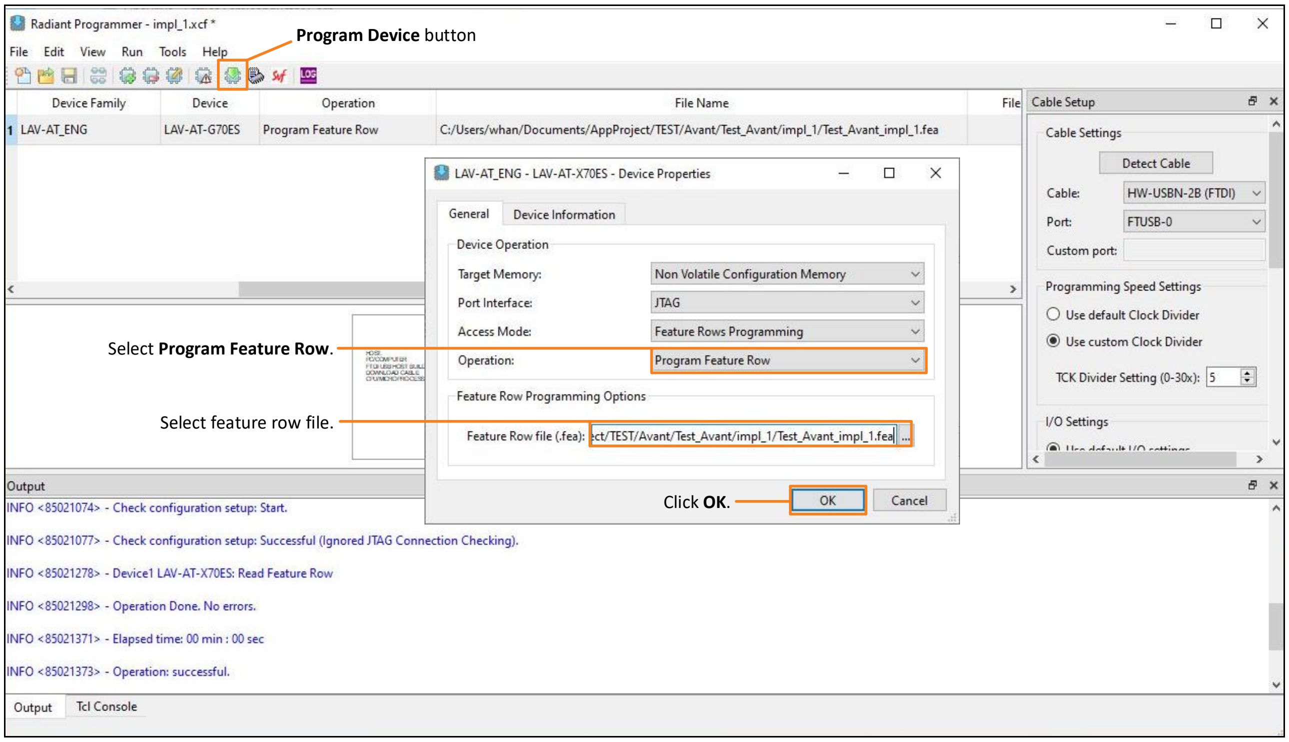
Item	Default	Description
Boot_SEL	0	Boot mode select 0 – Dual boot 1 – Single boot
Cust_IDCODE_EN	0	Customer IDCODE enable 0 – OFF 1 – ON
MSPI_Reset_EN	0	Enable HW reset signal to flash device
tI[1:0]	0	Extra delay cycles for MCSN idle time
tT[2:0]	0	Extra delay cycles from MCLKP to MSCN high
tL[2:0]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

Item	Default	Description
MSPI_CPOL	0	MSPI clock polarity select 0 – Active-high clocks selected. In idle state, MCLKP is low. 1 – Active-low clocks selected. In idle state, MCLKP is high.
MSPI_TX_Edge	0	MSPI transmit data edge select 0 – Transmit data on MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA 1 – Transmit data on inverted MCLKP edge
MSPI_RX_Edge	0	MSPI receive data edge select 0 – Receive data on MCLKP edge as defined by MSPI_CPOL and MSPI_CPHA 1 – Receive data on inverted MCLKP edge
SSPI_CPHA	0	SSPI clock format select 0 – Sampling of data occurs at the leading (first) edge of SCLKP 1 – Sampling of data occurs at the trailing (second) edge of SCLKP
SSPI_CPOL	0	SSPI clock polarity select 0 – Active-high clocks selected. In idle state, SCLKP is low. 1 – Active-low clocks selected. In idle state, SCLKP is high.
SSPI_TX_Edge	0	SSPI transmit data edge select 0 – Transmit data on SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA 1 – Transmit data on inverted SCLKP edge
SSPI_RX_Edge	0	SSPI receive data edge select 0 - Receive data on SCLKP edge as defined by SSPI_CPOL and SSPI_CPHA 1 - Receive data on inverted SCLKP edge
SSPI_LSBF	0	SSPI shifting direction select 0 – MSB first shifting for transmitting and receiving data 1 – LSB first shifting for transmitting and receiving data
SSPI Auto[2:0]	0	Enable SSPI auto function for downstream daisy chain 000 – SPI auto/SCM mode disabled 001 – SCM mode 010 – SPI auto x1 mode 011 – SPI auto x2 mode 100 – SPI auto x4 mode 101 – SPI auto DDR x8 mode 110 – Reserved 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 Commands	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	Target idle timer count value 0000 – Disabled (default); bitstream sent in one continuous stream 0001 – 200s 0010 – 100s 0011 – 50s 0100 – 25s 0101 – 10s 0110 – 5s 0111 – 1s 1000 – 750ms 1001 – 500ms 1010 – 250ms 1011 – 100ms 1100 – 75ms 1101 – 50ms 1110 – 25ms 1111 – 10ms
MSPI Preamble Timer count value[2:0]	0	Controller preamble timer count value 000 – 200ms 001 – 100ms 010 – 50ms 011 – 40ms 100 – 20ms 101 – 1ms 110 – 500us 111 – 100us
Boot_Offset[47:0]	0	Address of primary boot image in SPI flash

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.


Radiant Programmer - impl_1.xcf *
File Edit View Run Tools Help
Program Device button
Select **Program Feature Row**.
Select feature row file.
Click **OK**.
OK
Cancel
Cable Setup
Detect Cable
General
Device Information
Output
Tcl Console

Device Family	Device	Operation	File Name
1 LAV-AT_ENG	LAV-AT-G70ES	Program Feature Row	C:/Users/whan/Documents/AppProject/TEST/Avant/Test_Avant/impl_1/Test_Avant_impl_1.fea

LAV-AT_ENG - LAV-AT-X70ES - Device Properties

Device Operation
Target Memory: Non Volatile Configuration Memory
Port Interface: JTAG
Access Mode: Feature Rows Programming
Operation: Program Feature Row

Feature Row Programming Options
Feature Row file (.fea): ct/TEST/Avant/Test_Avant/impl_1/Test_Avant_impl_1.fea

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

Programming Speed Settings
Use default Clock Divider
Use custom Clock Divider
TCK Divider Setting (0-30x): 5

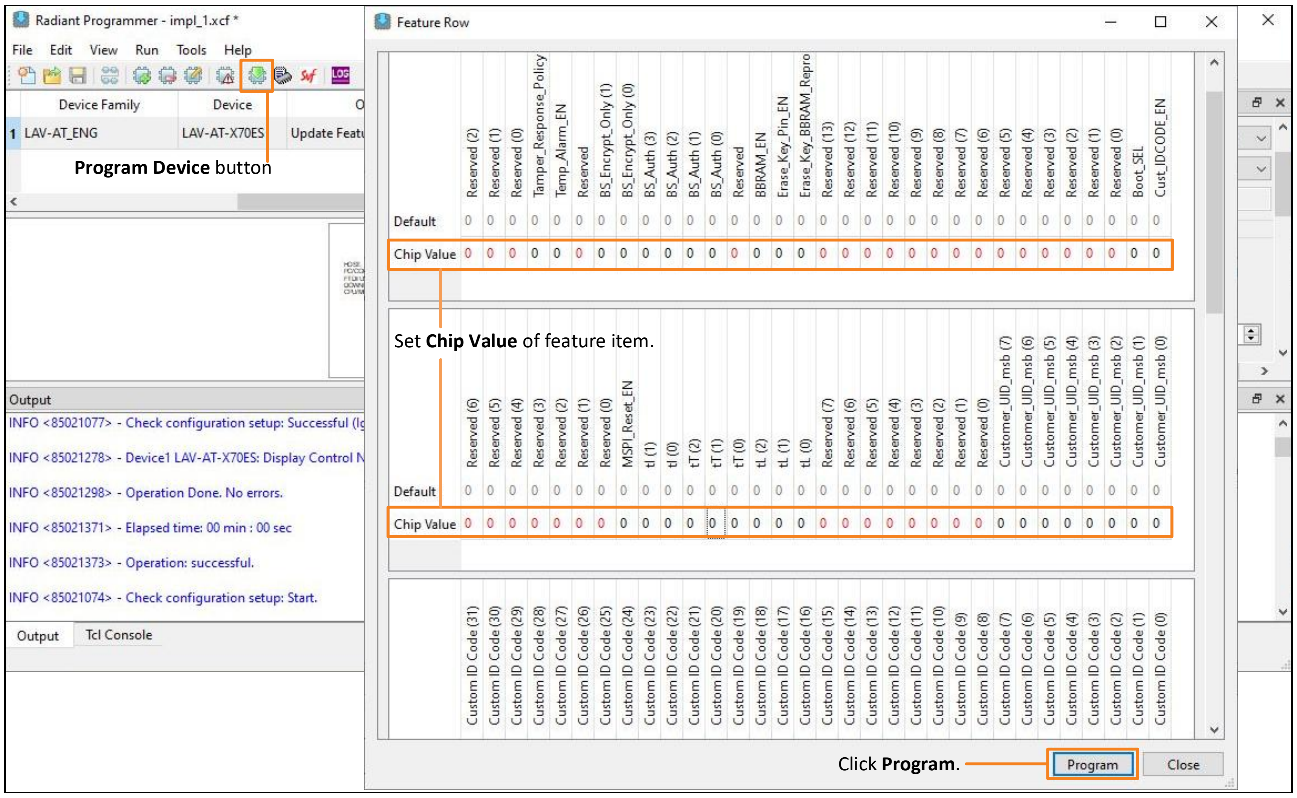
I/O Settings

```
INFO <85021074> - Check configuration setup: Start.
INFO <85021077> - Check configuration setup: Successful (Ignored JTAG Connection Checking).
INFO <85021278> - Device1 LAV-AT-X70ES: Read Feature Row
INFO <85021298> - Operation Done. No errors.
INFO <85021371> - Elapsed time: 00 min : 00 sec
INFO <85021373> - Operation: successful.
```

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.



	Device Family	Device	O
1	LAV-AT_ENG	LAV-AT-X70ES	Update Featu

```
INFO <85021077> - Check configuration setup: Successful (Ig
INFO <85021278> - Device1 LAV-AT-X70ES: Display Control N
INFO <85021298> - Operation Done. No errors.
INFO <85021371> - Elapsed time: 00 min : 00 sec
INFO <85021373> - Operation: successful.
INFO <85021074> - Check configuration setup: Start.
```

	Reserved (2)	Reserved (1)	Reserved (0)	Tamper_Response_Policy	Temp_Alarm_EN	Reserved	BS_Encrypt_Only (1)	BS_Encrypt_Only (0)	BS_Auth (3)	BS_Auth (2)	BS_Auth (1)	BS_Auth (0)	Reserved	BBRAM_EN	Erase_Key_Pin_EN	Erase_Key_BBRAM_Repro	Reserved (13)	Reserved (12)	Reserved (11)	Reserved (10)	Reserved (9)	Reserved (8)	Reserved (7)	Reserved (6)	Reserved (5)	Reserved (4)	Reserved (3)	Reserved (2)	Reserved (1)	Reserved (0)	Boot_SEL	Cust_IDCODE_EN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

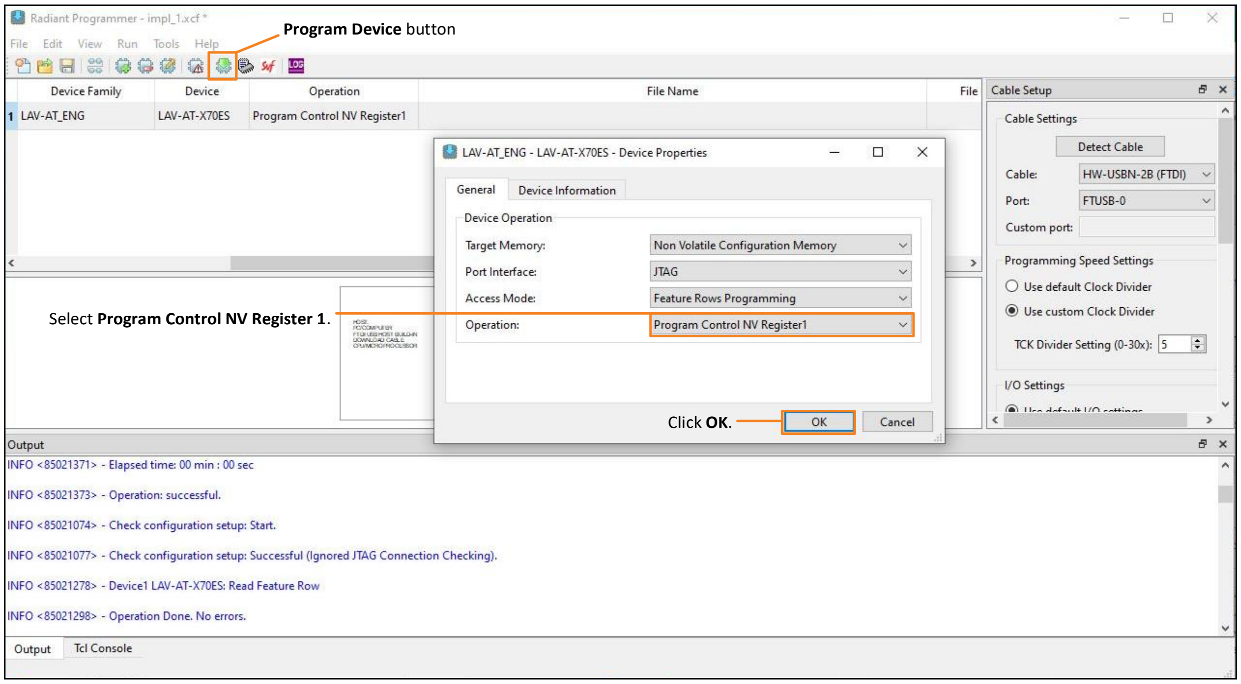
	Reserved (6)	Reserved (5)	Reserved (4)	Reserved (3)	Reserved (2)	Reserved (1)	Reserved (0)	MSPI_Reset_EN	tI (1)	tI (0)	tT (2)	tT (1)	tT (0)	tL (2)	tL (1)	tL (0)	Reserved (7)	Reserved (6)	Reserved (5)	Reserved (4)	Reserved (3)	Reserved (2)	Reserved (1)	Reserved (0)	Customer_UID_msb (7)	Customer_UID_msb (6)	Customer_UID_msb (5)	Customer_UID_msb (4)	Customer_UID_msb (3)	Customer_UID_msb (2)	Customer_UID_msb (1)	Customer_UID_msb (0)
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Custom ID Code (31)	Custom ID Code (30)	Custom ID Code (29)	Custom ID Code (28)	Custom ID Code (27)	Custom ID Code (26)	Custom ID Code (25)	Custom ID Code (24)	Custom ID Code (23)	Custom ID Code (22)	Custom ID Code (21)	Custom ID Code (20)	Custom ID Code (19)	Custom ID Code (18)	Custom ID Code (17)	Custom ID Code (16)	Custom ID Code (15)	Custom ID Code (14)	Custom ID Code (13)	Custom ID Code (12)	Custom ID Code (11)	Custom ID Code (10)	Custom ID Code (9)	Custom ID Code (8)	Custom ID Code (7)	Custom ID Code (6)	Custom ID Code (5)	Custom ID Code (4)	Custom ID Code (3)	Custom ID Code (2)	Custom ID Code (1)	Custom ID Code (0)

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.


Radiant Programmer - impl_1.xcf *
File Edit View Run Tools Help
Program Device button
Device Family
Device
Operation
File Name
Cable Setup
Cable Settings
Detect Cable
LAV-AT_ENG - LAV-AT-X70ES - Device Properties
General
Device Information
Device Operation
Select **Program Control NV Register 1**.
Click **OK**.
OK
Cancel
Output
Tcl Console

	Device Family	Device	Operation	File Name
1	LAV-AT_ENG	LAV-AT-X70ES	Program Control NV Register1	

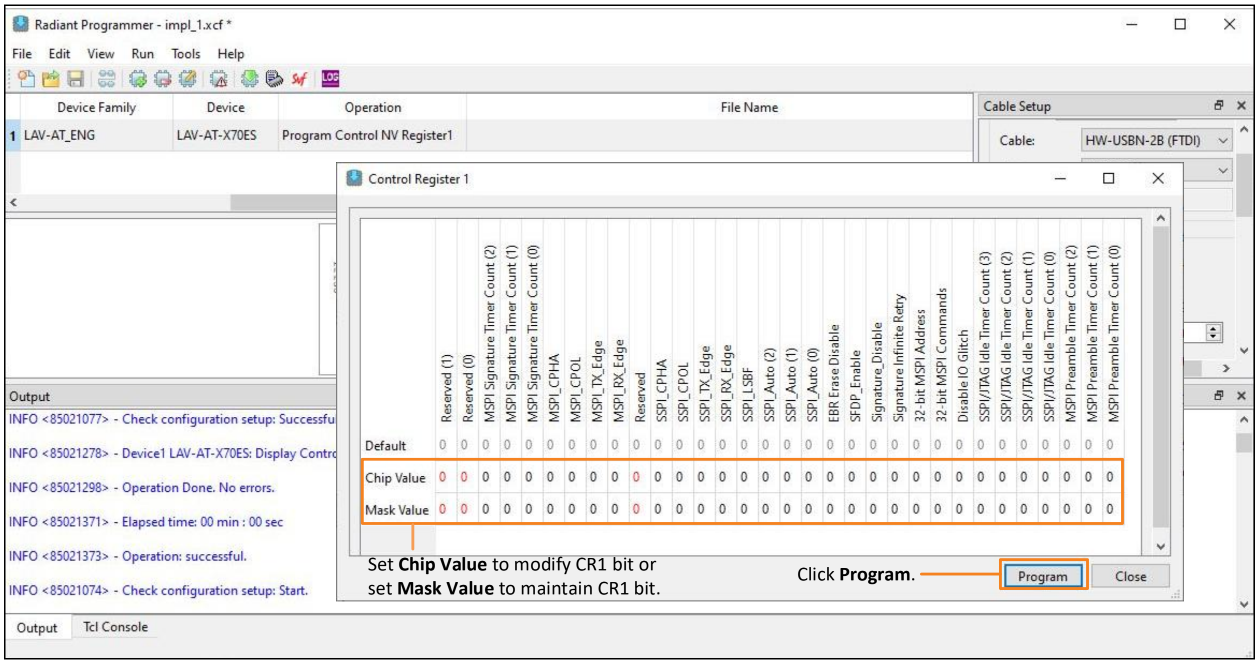
Target Memory: Non Volatile Configuration Memory
Port Interface: JTAG
Access Mode: Feature Rows Programming
Operation: Program Control NV Register1

Cable: HW-USBN-2B (FTDI)
Port: FTUSB-0
Custom port:
Programming Speed Settings
( ) Use default Clock Divider
(●) Use custom Clock Divider
TCK Divider Setting (0-30x): 5
I/O Settings

```
INFO <85021371> - Elapsed time: 00 min : 00 sec
INFO <85021373> - Operation: successful.
INFO <85021074> - Check configuration setup: Start.
INFO <85021077> - Check configuration setup: Successful (Ignored JTAG Connection Checking).
INFO <85021278> - Device1 LAV-AT-X70ES: Read Feature Row
INFO <85021298> - Operation Done. No errors.
```

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.


Radiant Programmer - impl_1.xcf *
File Edit View Run Tools Help
Device Family
Device
Operation
File Name
Cable Setup
Cable: HW-USBN-2B (FTDI)
Control Register 1
Set **Chip Value** to modify CR1 bit or
set **Mask Value** to maintain CR1 bit.
Click **Program**.
Program
Close
Output
Tcl Console

	Device Family	Device	Operation	File Name
1	LAV-AT_ENG	LAV-AT-X70ES	Program Control NV Register1	

	Reserved (1)	Reserved (0)	MSPI Signature Timer Count (2)	MSPI Signature Timer Count (1)	MSPI Signature Timer Count (0)	MSPI_CPHA	MSPI_CPOL	MSPI_TX_Edge	MSPI_RX_Edge	Reserved	SSPI_CPHA	SSPI_CPOL	SSPI_TX_Edge	SSPI_RX_Edge	SSPI_LSBF	SSPI_Auto (2)	SSPI_Auto (1)	SSPI_Auto (0)	EBR Erase Disable	SFDP_Enable	Signature_Disable	Signature Infinite Retry	32-bit MSPI Address	32-bit MSPI Commands	Disable IO Glitch	SSPI/JTAG Idle Timer Count (3)	SSPI/JTAG Idle Timer Count (2)	SSPI/JTAG Idle Timer Count (1)	SSPI/JTAG Idle Timer Count (0)	MSPI Preamble Timer Count (2)	MSPI Preamble Timer Count (1)	MSPI Preamble Timer Count (0)
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chip Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mask Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

```
INFO <85021077> - Check configuration setup: Successfu
INFO <85021278> - Device1 LAV-AT-X70ES: Display Contro
INFO <85021298> - Operation Done. No errors.
INFO <85021371> - Elapsed time: 00 min : 00 sec
INFO <85021373> - Operation: successful.
INFO <85021074> - Check configuration setup: Start.
```

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 Avant One-Time Programmable Security User Guide (FPGA-TN-02384).

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 Avant 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.

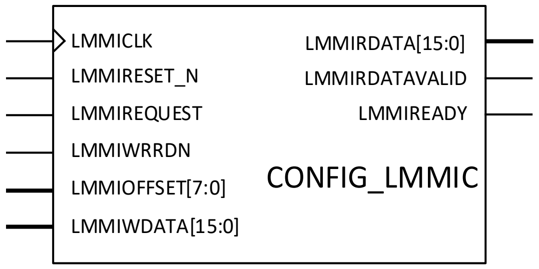


Figure D.1. CONFIG_LMMIC Primitive Pin Diagram

Table D.1. CONFIG_LMMIC Primitive Pins[1]

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.

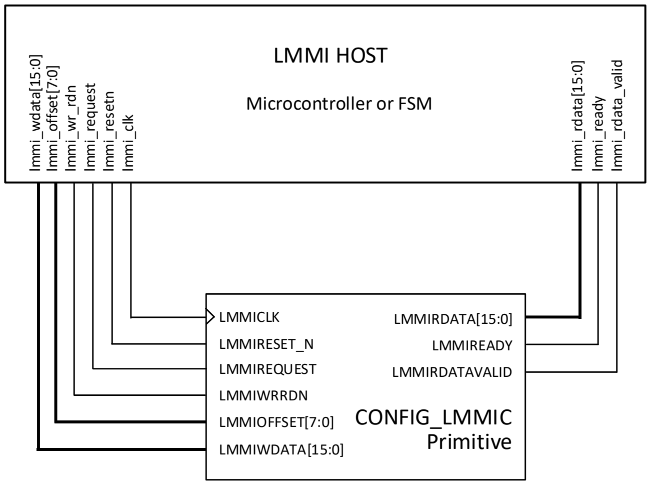


Figure D.2. CONFIG_LMMIC Primitive Connection

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

```
    wire        lmmi_clk;
    wire        lmmi_resetn;
    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_resetn),          // 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 Avant 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][1]	16'h0	Request to enable LMMI access to CFG logic LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000000; => NOWAIT LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000001; => WAIT LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000002; => FORCE LMMI_CFG_PORT_REQUEST[31:0] = 32'h7A000003; => RELEASE
8'h02		LMMI_CFG_PORT_REQUEST[31:16][1]	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
				LMMI_CFG_16BIT_ENABLE[0] = 1'b0; => LMMI_CFG_DATA = {8'h00, CFG_DATA[7:0]};
				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] first 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 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.

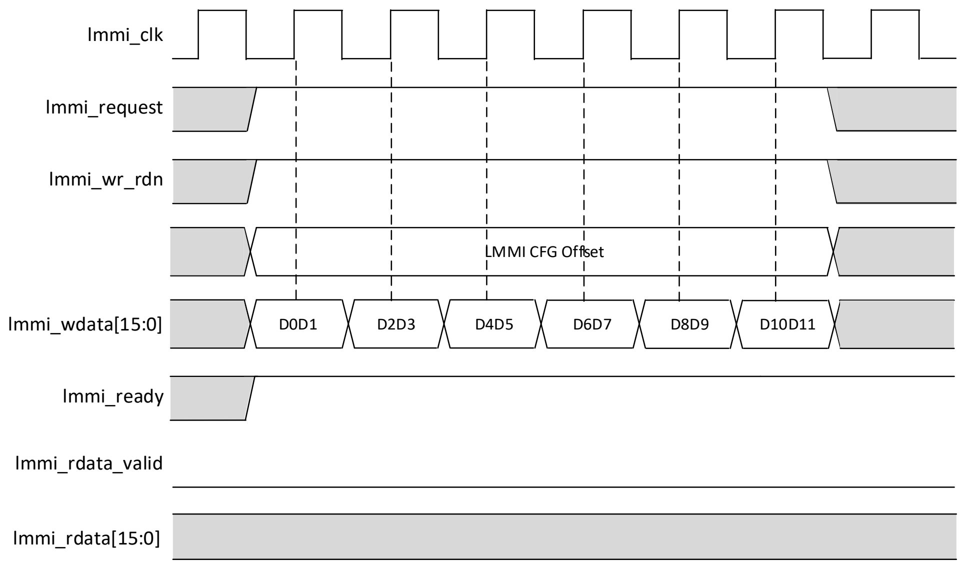


Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode

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.

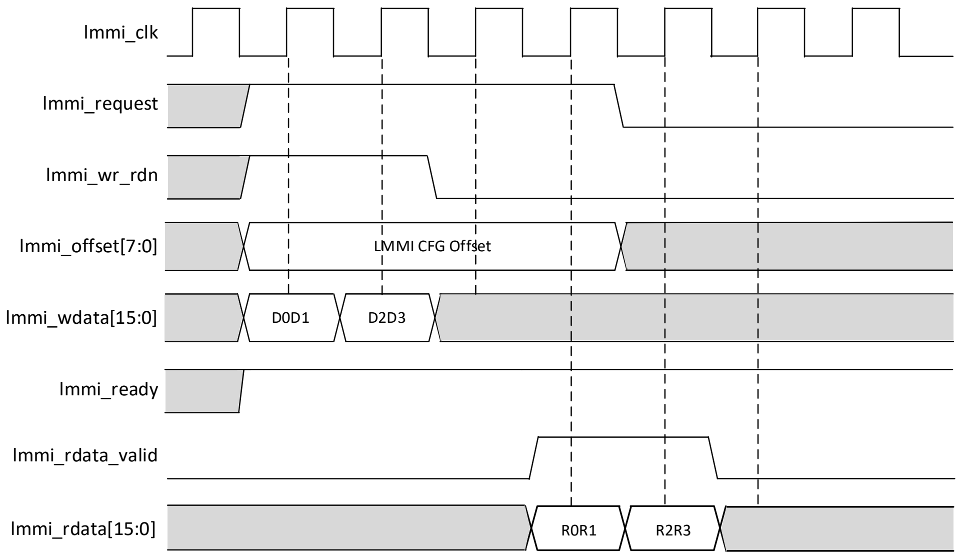


Figure D.4. Waveform for LMMI Read Operation

The following are important guidelines for using the CONFIG_LMMI primitive:

- LMMI Request for Command Data Transmission
  The lmmi_request signal must be asserted high when transmitting command data. This ensures that the command is recognized and processed correctly by the receiving system.
- LMMI Request for Read Operation
  During a read operation, after the command data has been sent, the lmmi_request signal must remain high for a specific number of clock cycles, denoted as N. The value of N corresponds to the number of expected data bytes to be returned divided by two. For instance, if the expected return data is 8 bytes, when using the CONFIG_LMMI primitive, the lmmi_request signal must be kept high for four clock cycles to ensure proper data retrieval.

Failure to adhere to these guidelines can lead to unexpected behavior in the LMMI block. One such issue is the lmmi_ready signal getting stuck at low indefinitely. This situation requires toggling of the lmmi_reset signal to recover and resume normal operation. Ensuring the lmmi_request signal is managed correctly is crucial to avoid such disruptions.

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[1]	Byte0	Byte1	Byte2	Byte3	# DATA BYTES	# RETURN BYTES	PROG_ QUALIFY	Type[2]	Description
PROG_ENABLE	0x80	0x00	0x0c	0x00–0x02	—	—	—	IMM	Enable programming based on byte3 value: 0x00 – CRAM/INIT/OTP 0x01 – CRAM/INIT only 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.

Command Name[1]	Byte0	Byte1	Byte2	Byte3	# DATA BYTES	# RETURN BYTES	PROG_ QUALIFY	Type[2]	Description
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[3]	—	4	—	Read	Read user OTP.
PROG_OTP	0x80	0x04	0x4C	ROW[3]	4	—	Y	Data Write	Write user OTP.
PROG_OTP_ SHADOW	0x80	0x08	0x55	ROW[3]	8	—	Y	Data Write	Write user OTP shadow register.
LOCK_OTP_ROW	0x80	0x00	0x4D	ROW[3]	—	—	Y	IMM	Lock user OTP row.
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 Avant Configuration Security User Guide (FPGA-TN-02335).

References

- Lattice Avant Platform - Overview Data Sheet (FPGA-DS-02107)
- Lattice Avant Platform - Specifications Data Sheet (FPGA-DS-02112)
- Avant-E web page
- Avant-G web page
- Avant-X web page

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

- IP Cores and Reference Designs for Avant Devices
- Kits, Boards, and Demonstrations for Avant Devices

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

- Lattice Avant Embedded Memory User Guide (FPGA-TN-02289)
- Lattice Avant Multi-Boot User Guide (FPGA-TN-02314)
- Lattice Avant sysCONFIG User Guide (FPGA-TN-02299)
- Lattice Avant sysDSP User Guide (FPGA-TN-02293)
- Lattice Avant sysI/O User Guide (FPGA-TN-02297)
- 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 Avant Configuration Security User Guide (FPGA-TN-02335)
- Lattice Avant One-Time Programmable Security User Guide (FPGA-TN-02384)

Other references:

- Lattice Memory Mapped Interface and Lattice Interrupt Interface (FPGA-UG-02039)
- Watchdog Timer IP User Guide (FPGA-IPUG-02097)
- Reveal User Guide for Radiant Software (2024.1)
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

Revision 0.88, January 2026

Section	Change Summary
Configuration Details	In Figure 4.3. Configuration from PROGRAMN Timing: • Updated figure to include t_{INIT_HIGH}. • Added note on t_{INITL}.
Software Selectable Options	In Table 7.10. MULTI_BOOT_SEL Option: • Corrected typographical error in option name.
Daisy Chaining	• In the Flow-Through Mode section: • Added description on the maximum MCLK frequency and optional buffer. • Updated Figure 8.1. Lattice Avant in Configuration Daisy Chain in Flow-Through Mode by adding an optional buffer on the MCLKP line. • In the Bypass Mode section: • Added description on the maximum MCLK frequency and optional buffer. • Updated Figure 8.2. Lattice Avant in Configuration Daisy Chain in Bypass Mode by adding an optional buffer on the MCLKP line.
Appendix C. Modifying User OTP Settings	In Table C.1. Feature Row Items: • Updated item name to *32-bit MSPI Commands*.
Appendix D. Configuration Access from User Logic	In the LMMI CFG Command Execution section: • Updated description on PROG_CNTRL0 command example. • Updated Figure D.3. Waveform for LMMI Write Operation in 16-Bit Mode. • Updated description on READ_CNTRL0 command example. • Updated Figure D.4. Waveform for LMMI Read Operation.

Revision 0.87, July 2025

Section	Change Summary
Introduction	Removed reference to section on finalizing the device setup.
Features	• Linked Lattice Avant Configuration Security User Guide (FPGA-TN-02335) references to FAQ page. • Removed note regarding specific feature support for Avant devices.
Configuration Details	In Table 4.1. Bitstream Size versus Recommended SPI Flash Size: • Updated bitstream sizes for LAV-AT-E/G/X30 (maximum EBR only), LAV-AT-E/G/X50, and LAV-AT-E/G/X70. • Updated recommended SPI flash sizes for LAV-AT-E/G/X30 (maximum EBR, single boot and dual boot).
Device Configuration	• Removed the Specifications and Timing Diagrams - Target SPI Port Waveforms section under Target SPI Mode. • Updated Figure 6.5. Read Command Waveforms, Figure 6.7. Data Write Command Waveforms, and Figure 6.8. Immediate Action Command Waveforms. • Updated Figure 6.6. Data Read Command Waveforms and title to match section header. • In Table 6.7. Status Register 1: • Updated bit[27] to reserved. • Updated bits[19:18] to Device State and updated description. • Updated bit[14] to JTAG sysCONFIG Lock and added description. • Updated bit[13] to JTAG ispTracy Lock and added description. • Updated bit[12] to JTAG Boundary Scan Lock and added description. • In Table 6.9. Control Register 0: • Updated description for TRAN_SEC_ENGINE.
Software Selectable Options	• In Table 7.3. MASTER_SPI_PORT Option: • Added note on XSPI setting in relation to MSPI_RX_EDGE.

Section	Change Summary
	• In Table 7.16. MSPI_RX_EDGE Option: • Added note on use of default setting when MASTER_SPI_PORT = XSPI.
Appendix C. Modifying User OTP Settings	• In Table C.1. Feature Row Items: • Removed all reserved rows. • Removed security-related settings. Security-related settings are documented in the Lattice Avant One-Time Programmable Security User Guide (FPGA-TN-02384). • Removed Notes 1 through 4. • Replaced content in the Advanced Security Keys Programming section with a statement containing reference to the Lattice Avant One-Time Programmable Security User Guide (FPGA-TN-02384). • Added a note on device setup finalization.
Appendix D. Configuration Access from User Logic	In Table D.3. CONFIG_LMMIC Supported Commands: • Linked Lattice Avant Configuration Security User Guide (FPGA-TN-02335) reference to FAQ page in Note 3.
References	• Linked Lattice Avant Configuration Security User Guide (FPGA-TN-02335) reference to FAQ page. • Added Lattice Avant One-Time Programmable Security User Guide (FPGA-TN-02384).

Revision 0.86, December 2024

Section	Change Summary
All	• Made editorial changes. • Changed instances of MSIO, SSIO, MCLK, and SCLK to MDQ, SDQ, MCLKP, and SCLKP.
Abbreviations in This Document	• Updated section title and introducing sentence. • Added and removed items.
Introduction	Updated description.
Features	• Removed frequencies listed with SPI modes. • Added mention of configuration mode and user function mode to configuration bridging listing. • Modified bitstream encryption listing and updated reference document. • Updated bitstream authentication reference document. • Removed listing on transparent programming. • Updated device names to Avant-E, Avant-G, and Avant-X in note.
Definition of Terms	• Updated description for BIT, Port, Dual-Boot, Multi-Boot, Number Formats, Transparent Mode, Refresh, and Dry-Run listings. • Changed listing from User Mode to User Function Mode. • Changed listing from Direct Mode (Foreground Mode) to Unprogrammed Mode and updated description. • Removed controller SPI, target SPI, and offline mode listings. • Reorganized content into a table.
Configuration Details	• Updated description. • Changed section title from Bitstream/PROM Sizes to Bitstream and SPI Flash Sizes. • In Table 4.1. Bitstream Size versus Recommended SPI Flash Size: • Changed table title. • Reworked table headers. • Updated bitstream size for LAV-AT-E/G/X50 with no EBR. • Updated bitstream size and recommended flash sizes for LAV-AT-E/G/X50 with maximum EBR. • Updated note to table. • In the Programming and Configuration Ports section: • Updated section title.

Section	Change Summary
	• Added description moved from previous Controller Configuration Process and Flow section. • In the Configuration Ports Arbitration section: • Updated description. • Updated Figure 4.1. Configuration Control Flow. • Reorganized information into sysCONFIG Pins and sysCONFIG Pin List sections. • In the sysCONFIG Pins section: • Updated description. • Updated Figure 4.2. sysCONFIG Pins. • In the sysCONFIG Pin List section: • In Table 4.3. Default State of sysCONFIG Pins: • Renamed table header from Hardware Default to Unprogrammed Mode Default. • Renamed table header from Software Default to User Function Mode Default. • Updated configuration mode default states for sysCONFIG pins CFGMODE, PROGRAMN, MMOSI/MDQ0, MMISO/MDQ1, MDQ[2:7], MDS, SMOSI/SDQ0, SMISO/SDQ1, and SDQ[2:7]. • Updated unprogrammed mode default states for sysCONFIG pins MMOSI/MDQ0, MMISO/MDQ1, MDQ[2:7], MDS, SMOSI/SDQ0, SMISO/SDQ1, SDQ[2:7], and SDS. • Removed note on shared sysCONFIG pin type achieved through non-volatile EFUSE feature setting. • Added note on CFGMODE pin state for target SPI and JTAG modes. • Rearranged notes and numberings. • Reorganized PROGRAMN, INITN, and DONE sections into System Pins section. • In the System Pins section: • Added CFGMODE section. • Updated descriptions in the PROGRAMN, INITN, and DONE sections. • Moved MCSNO/MSDO and SCSNO/SSDO sections into this section. • In the Controller SPI sysCONFIG Pins section: • Updated description. • Updated initial frequency of MCLKP to nominally 3.1 MHz in the MCLKP section. • Updated description in the MCSN section. • Removed the Avant MCLK Valid Frequencies table and replaced with cross reference to Table 7.5. MCCLK_FREQ Option. • Added the MRSTN section. • Updated description in the SMOSI/SDQ0, SMISO/SDQ1, and SDQ[2:7] sections under the Target SPI sysCONFIG Pins section. • Updated description in the TCK, TMS, TDI, and TDO sections under the JTAG Pins section. • Reorganized information into Port Persistence section and PERSISTENT Control Bits and sysCONFIG Port Persistence Options subsections. • In the PERSISTENT Control Bits section: • Updated section title and description. • In the sysCONFIG Port Persistence Options section: • In Table 4.4. sysCONFIG Port Persistence Options: • Updated table title. • Added XSPI_DIFF_CLK setting for MASTER_SPI_PORT. • Updated XSPI pins affected for MASTER_SPI_PORT. • Updated headers from Port Setting to Option Name, Value to Setting, and Details to Description. • Removed note on JTAG enable pin.
Configuration Process and Flow	• Updated naming to user function mode in Figure 5.1. Configuration Flow. • Moved discussion on programming and configuration ports to the Programming and

Section	Change Summary
	Configuration Ports section. • In the Power-Up Sequence section: • Updated power supply input rails to V_{CCIO1} and V_{CCIO2}. • Updated Figure 5.2. Configuration from POR Timing. • Removed note on condition ignored by Avant-AT-E70B devices in relation to PROGRAMN in the Initialization section. • In the Configuration section: • Updated description. • Reorganized content into Controller SPI Configuration and Target SPI Configuration sections. • In Target SPI Configuration section: • Removed description on highlights in previous Target SPI Configuration Flow Diagrams section under the Target SPI Mode section and moved updated content into this section. • Updated Figure 5.3. Target SPI Configuration Flow. • In the Wake-Up section: • Reorganized content into Wake-Up Signals and Wake-Up Sequence sections. • Updated description in previous Wake-Up Sequence section and merged content into re-worked section. • Updated description in the Early I/O Release section. • Updated description in the User Function Mode section. • Removed the Clearing the Configuration Memory and Re-initialization section.
Device Configuration	• Updated description. • Updated description for *Unprogrammed* and *Operational* in the Configuration Modal States section. • Renamed and moved Configuration Commands section. • Removed Command Fields Definition section in Configuration Commands section. • In the Controller SPI Mode section: • Updated description. • Updated nominal MCLKP frequency to 3.1 MHz. • Removed 33-Ω damping resistor recommendation for higher MCLKP frequencies with a fast slew rate. • Updated Figure 6.2. Avant Controller SPI Port with SPI Flash. • In Table 6.1. Controller SPI Configuration Port Pins: • Updated description for MCLKP. • Updated function and description for MDQ[2:3] and MDQ[4:7]. • Updated function, direction, and description for MMOSI/MDQ0 and MMISO/MDQ1. • Moved Table 6.1. Controller SPI Configuration Port Pins into new Controller SPI Configuration Port Pins section. • Updated description and reorganized content from previous Method to Enable the Controller SPI Port section into Enabling Controller SPI Port in Configuration Phase and Enabling Controller SPI Port Persistence in User Function Mode sections. • Updated description in the Dual-Boot and Multi-Boot and Ping-Pong Boot sections. • Updated Figure 6.3. Jump Table in the Ping-Pong Boot section. • Updated section title and description in the Dual, Quad, and xSPI Controller SPI Read Mode section. • In the Target SPI Mode section: • Updated description. • In Table 6.2. Target SPI Configuration Port Pins: • Updated description for SCSN. • Updated function for SMOSI/SDQ0, SMISO/SDQ1, SDQ[2:3], and SDQ[4:7].

Section	Change Summary
	• Moved Table 6.2. Target SPI Configuration Port Pins into new Target SPI Configuration Port Pins section. • Updated description and reorganized content from previous Method to Enable the Target SPI Port section into Enabling Target SPI Port in Configuration Phase and Enabling Target SPI Port Persistence in User Function Mode sections. • Updated Figure 6.5. Target SPI Read Waveforms and Figure 6.6. Target SPI Write Waveforms in the Specifications and Timing Diagrams – Target SPI Port Waveforms section. • Updated section title and description in Dual, Quad, and xSPI Target SPI Port section. • In the Command Waveforms section: • Removed Class A Command Waveforms, Class B Command Waveforms, Class C Command Waveforms, and Class D Command Waveforms sections. • Added Read Command Waveforms, Data Read Command Waveforms, Data Write Command Waveforms, Immediate Action (IMM) Command Waveforms, and Delayed Action Command Waveforms sections. • In the Target SPI to Controller SPI Bridge section: • Updated description. • Updated Figure 6.12. Target SPI to Controller SPI Bridge Functional Diagram and Figure 6.13. Target SPI to Controller SPI Bridge Block Diagram. • In the JTAG Mode section: • Updated description including removing *IEEE 1532 compliant programming* from listing. • Removed Method to Enable the JTAG Port for Configuration section. • Removed TCK Frequency - JTAG to Controller SPI Bridge entries in Table 6.4. JTAG AC Timing Requirements in the JTAG Port AC Timing Requirements section. • In the JTAG to Controller SPI Bridge section: • Updated description. • Updated Figure 6.14. JTAG to Controller SPI Bridge Block Diagram. • Removed ispTRACY information, revised section title to JTAG Reveal Support, and updated description. • Added JTAG Operating Modes section. • In the Status Registers section: • Updated section title and description. Reworked section into subsections. • Removed note on authentication. • In Table 6.6. Status Register 0 and Table 6.7. Status Register 1: • Removed Mask and Reset Condition columns. • Renamed columns from Bit to Field, Description to Name, and Definition to Description. • Modified formatting of bit fields. • Updated various registers, names, and descriptions. • Added registers. • Added Table 6.8. Status Register 2. • In the Control Registers section: • Reworked section into subsections and updated descriptions. • Updated and merged register descriptions into tables. • In Table 6.9. Control Register 0: • Updated table title. • Changed column name from BIT to Field. • Added and updated registers. • Updated descriptions for various registers. • In Table 6.10. Control Register 1: • Updated table title. • Changed column name from BIT to Field and Definition to Name.

Section	Change Summary
	• Added and updated registers. • Updated names and descriptions of various registers. • Added User Mode Register section. • In the Configuration Commands section: • Reorganized content. • Updated descriptions in the Read Commands section and subsections. • Removed PORT REQUEST Command, MSPI_MODE Command, and SSPI_MODE Command sections. • In Table 6.12. Target Configuration Commands in Target Configuration Command Set section: • Added note for PROG_QUALIFY. • Updated command list and various command descriptions. • Added Target Configuration Command Details section.
Software Selectable Options	• Updated descriptions and reorganized content into the Accessing Software Selectable sysCONFIG Options and sysCONFIG Options sections. • Updated Figure 7.1. sysCONFIG Options in Global Page of Device Constraint Editor and its figure title in the Accessing Software Selectable sysCONFIG Options section. • In Table 7.1. sysCONFIG Options: • Added the options MSPI_RESET_PORT, MSPI_TX_EDGE, MSPI_RX_EDGE, MSPI_CPHA, SSPI_TX_EDGE, SSPI_RX_EDGE, and SSPI_CPHA. • Removed MSPI_CLK_PHASE, SSPI_CLK_PHASE, PROGRAMN_RECOVERY, and Unique_ID options. • Removed note to table on CONFIG_MODE option. • Updated descriptions and reorganized content into tables for all sections from SLAVE_SPI_PORT to BitstreamRevision. • Added the sections MSPI_RESET_PORT, MSPI_TX_EDGE, MSPI_RX_EDGE, MSPI_CPHA, SSPI_TX_EDGE, SSPI_RX_EDGE, and SSPI_CPHA. • Updated ERASE_EBR_ON_REFRESH and SSPI_IDLE_TIMER section titles. • Removed MSPI_CLK_PHASE, SSPI_CLK_PHASE, PROGRAMN_RECOVERY, and Unique_ID sections.
Daisy Chaining	• Updated description. • In the Flow-Through Mode section: • Updated section title. • Updated description. • Updated Figure 8.1. Lattice Avant in Configuration Daisy Chain in Flow-Through Mode. • Added Bypass Mode and SCM Mode with External Host sections.
Appendix A. Avant Target SPI Programming Guide	Updated target SPI pin names in relation of the Lattice programming cable.
Appendix B. Avant Device Information	• Renamed section to *Avant Device Parameters* and reorganized content into Device ID section. • Added Configuration Memory section.
Appendix C. Modifying User OTP Settings	Added new appendix.
Appendix D. Configuration Access from User Logic	Added new appendix.
References	Added and rearranged items.

Revision 0.85, March 2024

Section	Change Summary
All	Removed Appendix B. Avant Bitstream File format section.
Controller Configuration	Added footnote for PROGRAMN in Initialization section.

Section	Change Summary
Process and Flow	
Device Configuration	• Added Avant-AT-E70B information in Controller SPI Modes. • Updated Table 6.12. JTAG AC Timing Requirements to add TCK Frequency – JTAG to Controller SPI Bridge rows.

Revision 0.84, December 2023

Section	Change Summary
All	• Indicated Avant-AT-E specific information and added Avant-AT-G/X support. • Applied inclusive language across the document.
Disclaimers	Updated this section.
Inclusive Language	Newly added section.
Introduction	Updated section to change external configuration engine to *external controller*.
Configuration Details	• Updated Table 4.1. Maximum Configuration Bits to update uncompressed bitstream size data, include Avant-AT-G/X values, and update device name of Avant-AT-E. • Updated Table 4.2. Avant Programming and Configuration Ports to remove 1532 support. • Updated Configuration Ports Arbitration section to change text to At Power Up (POR), PROGRAMN pin toggle (falling edge), or REFRESH command execution, the configuration logic will perform CDM (Erase CRAM, Reset INIT registers, and optionally erase EBR depending on settings in Control Register 1 (CR1). • Removed the Holding the PROGRAMN pin low information in PROGRAMN. • Updated Figure 4.2. sysCONFIG Pins to apply inclusive language and Pin directions. • Updated Table 4.3. Default State of the sysCONFIG Pins to update Hardware and Software Default columns for Target SPI (SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1) and added MRSTN pin in the Controller SPI group. • Updated Table 4.4. Avant MCLK Valid Frequencies to add 106.7 and 160.0 frequency.
Controller Configuration Process and Flow	• Replaced the sentence *You can arrange the order of these four phases is configurable to meet specific implementation requirements with the order of the External DONE release is configurable to meet specific implementation requirements* in Wake-up section. • Updated Early I/O Release section to remove left and right I/O references.
Device Configuration	• Replaced *This is the only method with which you can perform DUAL, QUAD and OCTAL read from SPI Flash* with *and port control commands which could invoke DUAL, QUAD and OCTAL read mode from SPI Flash* in Method to Enable the Controller SPI Port section. • Deleted *Dual boot can also be deployed with multi boot, allowing a golden (fail safe) design (or sixth design) to be available in the external Flash* in Dual-Boot and Multi-Boot Configuration Modes section. • Updated Table 6.6. MSPI_MODE Command Data Field Definition to change clock source values of Bit[1:0] to 400, 320, 266, and 213. • Updated Figure 6.3. Avant Controller SPI Port with SPI Flash to add MRSTN pin, add footnote, and change directions for MISO and MSIO.
Software Selectable Options	• Updated Figure 7.1. sysCONFIG Preferences in Global Tab, Lattice Radiant Device Constraint Editor. • Updated the following in Table 7.1. sysCONFIG Options: • Changed option name from MASTER _SIGNATURE_TIMER to *MSPI_SIGNATURE_TIMER*. • Added TransFR row. • Changed SLAVE_IDLE_TIMER to *SSPI_IDLE_TIMER*. • Changed MASTER_PREAMBLE_DETECTION_TIMER to *MSPI_PREAMBLE_DETECTION_TIMER*. • Removed MSPI_SHIT_ORDER row. • Updated the following section names: • MASTER _SIGNATURE_TIMER to MSPI_SIGNATURE_TIMER. • SLAVE_IDLE_TIMER to TARGET_IDLE_TIMER.

Section	Change Summary
	• MASTER_PREAMBLE_DETECTION_TIMER to MSPI_PREAMBLE_DETECTION_TIMER. • Added TRANSFR section. • Removed MSPI_SHIFT_ORDER section.
Daisy Chaining	Updated Figure 9.1. Lattice Avant in Configuration Daisy Chain with Target SPI in Flow Through Mode to change to Target SPI.
Appendix B. Avant Bitstream File Format	Removed references to sections that are not available in the document in Encrypted and Authenticated Bitstream Format.
Appendix C. Avant Device ID	Added values of Logic Capacity and 32-bit IDCODE for G70, G50, G30, X70, X50, X30 and updated Logic Capacity and 32-bit IDCODE for E70, E50, E30 in Table C.1. Lattice Avant Device ID.
References	Added this section.

Revision 0.83, December 2022

Section	Change Summary
Device Configuration	Corrected block in Figure 6.14. SSPI to MSPI Bridge Block Diagram and Figure 6.15. JTAG to MSPI Bridge Block Diagram to Avant.

Revision 0.82, November 2022

Section	Change Summary
All	• Updated the document to provide detailed information on Avant-E features only. • Updated titles and links of referenced documents.
Features	Updated list of features.
Configuration Details	• Updated device names in Table 4.1. Maximum Configuration Bits. • Updated location of INIT, DONE, and MCSNO/MSDO in Table 4.3. Default State of the sysCONFIG Pins. • General update to Table 4.4. Avant MCLK Valid Frequencies.
Device Configuration	• Removed the Bitstreams Command, the Config Usermode Registers, and the TransFR sections. • Updated Table 6.2. Target Configuration Commands to show Avant-E data only.
Software Selectable Options	• Updated Figure 7.1. sysCONFIG Preferences in Global Tab, Lattice Radiant Device Constraint Editor. • General update to the subsections that describe the configuration options in detail.
Technical Support Assistance	Added reference to the Lattice Answer Database on the Lattice website.

Revision 0.81, October 2022

Section	Change Summary
Appendix C. Avant Device ID	Updated devices to 500E, 300E, and 200E.

Revision 0.80, May 2022

Section	Change Summary
All	Preliminary release.

www.latticesemi.com