Lattice Avant sysCONFIG User Guide

Preliminary Technical Note

FPGA-TN-02299-0.85

March 2024

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

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

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AES	Advanced Encryption Standard
CRC	Cyclic Redundancy Check
DDR	Double Data Rate
EBR	Embedded Block RAM
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	Hash-based Message Authentication Code
I^{2}C	Inter-Integrated Circuit; A synchronous, multi-controller, multi-target, packet switched, single-ended, serial bus
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Action Group
LMMI	Lattice Memory Mapped Interface
LSB	Least Significant Bit
LUT	Look Up Table
MIB	Memory Interface Block
MSB	Most Significant Bit
MSPI	Controller Serial Peripheral Interface
OTP	One Time Programmable
PCB	Printed Circuit Board
POR	Power On Reset
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
TCK	Test Clock Pin
TDI	Test Data Input
TDO	Test Data Output
TMS	Test Mode Select
xSPI	Expanded Serial Peripheral Interface

1. Introduction

Lattice Avant™ is a low-power mid-range general purpose FPGA platform optimized for a wide range of applications across multiple markets. It has many new and unique features that make it one of the best-in-class FPGAs in its logic density range. Avant FPGAs have one of the fastest boot up times including ultrafast I/O booting capability. It also offers advanced options to enable multi-boot features 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 the power is removed. This non-volatile memory or configuration engine supplies the configuration data to the Avant device when it powers up or anytime the device needs to be updated. The Avant device provides a rich set of features for configuring the FPGA or programming the external non-volatile memory. There are many options available for building the programming solution that fits a particular set of needs. Each of the options available is described in detail in this document. Waveforms presented in this document are for reference only; for detailed timing recommendations, refer to Lattice Avant Platform - Specifications (FPGA-DS-02112).

2. Features

Main programming and configuration features of the Avant device are:

- Ultrafast I/O configuration for instant-on support
- Fast device configuration with Controller SPI (x1, x2, x4) at 150 MHz; Octal SPI[1] (x8, dual transfer rate) at 180 MHz
- Bitstream dry-run support[1] to ensure bitstream integrity
- Enhanced and flexible Multi-boot[1] support (32-bit addressing support with jump-table support)
- Configuration bridging[1] for easy external SPI programming (SSPI to Controller SPI bridge, JTAG to Controller SPI Bridge, LMMI to Controller SPI Bridge)
- User-selectable Booting Sequence
- Bitstream Encryption/Decryption[1] – 256 bits AES. For detailed information about the cypher key handling and advanced security features, refer to Advanced Configuration Security Usage Guide for Avant Family (FPGA-TN-02295).
- Bitstream Authentication[1] – ECDSA and RSA. For detailed information, refer to Advanced Configuration Security Usage Guide for Avant Family (FPGA-TN-02295).
- Multiple programming and configuration interfaces:
  - 1149.1 JTAG
  - Controller SPI includes SERIAL, DUAL, QUAD mode and Octal SPI[1] (x8, dual transfer rate)
  - Target SPI includes SERIAL, DUAL, QUAD mode and Octal SPI1 (x8, dual transfer rate)
  - Configuration Daisy chaining[1]
- Ping-Pong boot[1]
- Transparent programming of external memory
- Readback security and encryption[1] for design protection
- Compression of bitstreams

Note:

1. Feature is not supported in Avant-AT-E. Available on Avant-AT-G and Avant-AT-X.

3. Definition of Terms

This document uses the following terms to describe common functions:

- Programming – Programming refers to the process used to alter the contents of the external configuration memory.
- Configuration – Configuration refers to a change in the state of the SRAM memory cells.
- Configuration Mode – The configuration mode defines the method the device uses to acquire the configuration data from the volatile memory.
- Configuration Data – This is the data read from the non-volatile memory and loaded into the FPGA’s SRAM configuration memory. This is also referred to as a bitstream, or device bitstream.
- BIT – The BIT file is the configuration data for the 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 Lattice programming tool.
- HEX – The HEX file is also a configuration data file for the device. It is in HEX format and is normally requested by third party programming vendors.
- Port – A port refers to the physical connection used to perform programming and some configuration operations. Ports on the Avant device include JTAG, SPI physical connections.
- User Mode – The device is in user mode when configuration is complete, and the FPGA is performing the logic functions it has been programmed to perform.
- Offline Mode – Offline mode is a term that is applied to SRAM configuration or external memory programming. When using offline mode, the FPGA no longer operates in user mode. The contents of the SRAM configuration memory or external memory device are updated. The FPGA does not perform logic operations until offline mode programming/configuration is complete.
- Direct Mode (Foreground Mode) – The device is in a configuration mode and all the I/O pins are kept tri-stated.
- Dual Boot – Supports two configuration patterns that reside in a SPI Flash device. Whenever loading failure occurs with the primary pattern, the FPGA device searches for and loads a so-called golden or fail-safe pattern. Both patterns come from an off-chip non-volatile SPI memory.
- Ping-Pong Boot – Utilize the Jump Table to select an image for booting without changing location of the image in SPI flash.
- Multi Boot – The FPGA device determines and triggers the loading of the next pattern after a prior successful configuration. Multiple patterns (that is, more than two patterns) are available for the FPGA device to choose to load on demand. All the patterns are stored in an external SPI Flash memory.
- Transparent Mode – Transparent mode, also referred to as Background Mode, is used to update the SRAM configuration while leaving the device in user mode and all I/O pins remain operational.
- 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
- SPI – Serial Peripheral Interface Bus. An industry standard, full duplex, synchronous serial data link that uses a four-wire interface. The interface supports a single controller and single or multiple targets.
- Controller SPI – A configuration mode where the FPGA drives the controller clock and issues commands to read the bitstream from an external SPI Flash device.
- Target SPI (SSPI) – A configuration mode where the CPU drives the clock and issues commands to the FPGA for writing the bitstream into the SRAM cells.
- Refresh – The process of re-triggering a bitstream write operation. It is activated by toggling the PROGRAMN pin or issuing a REFRESH command, which emulates the PROGRAMN pin toggling.
- Dry-Run – The process triggered by the LSC_DEVICE_CONTROL 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.

4. Configuration Details

The Avant configuration SRAM memory of contains the active configuration, which is essentially the *switches* that define the behavior of the FPGA. The active configuration is, in most cases, retrieved from an external non-volatile memory. The non-volatile memory holds the configuration data that is loaded into the FPGA's configuration SRAM.

4.1. Bitstream/PROM Sizes

The Lattice 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 is variable. It is based on the amount of logic available in the FPGA, and the number of pre-initialized Embedded Block RAM (EBR) components. The Lattice Avant design using the largest device, with every EBR pre-initialized with unique data values and generated without compression turned on requires the largest amount of storage.

Table 4.1. Maximum Configuration Bits

Device	Scenario	All Uncompressed	SPI Mode	
		Unencrypted/Encrypted Bitstream Size (Mb)[1]	Recommended SPI Flash Size (Mb)	Dual Boot Recommended SPI Flash Size (Mb)
LAV-AT-E/G/X30	No EBR	44	64	128
	MAX EBR	58.4	64	128
LAV-AT-E/G/X50	No EBR	67	128	256
	MAX EBR	89.7	128	256
LAV-AT-E/G/X70	No EBR	107	128	256
	MAX EBR	142.6	256	512

Note:

1. Both unencrypted and encrypted bitstreams are the same size. Compression ratio depends on bitstream, so only uncompressed bitstream data is provided.

4.2. sysCONFIG Ports

Table 4.2. Avant Programming and Configuration Ports

Interface	Port	Description
JTAG	JTAG (IEEE 1149.1)	4-wire JTAG Interface
sysCONFIG™	SSPI (SERIAL, DUAL, QUAD, Octal-DDR)	Target Serial Peripheral Interface (SPI)
	MSPI (SERIAL, DUAL, QUAD, Octal-DDR)	Controller Serial Peripheral Interface (SPI)

4.3. Configuration Ports Arbitration

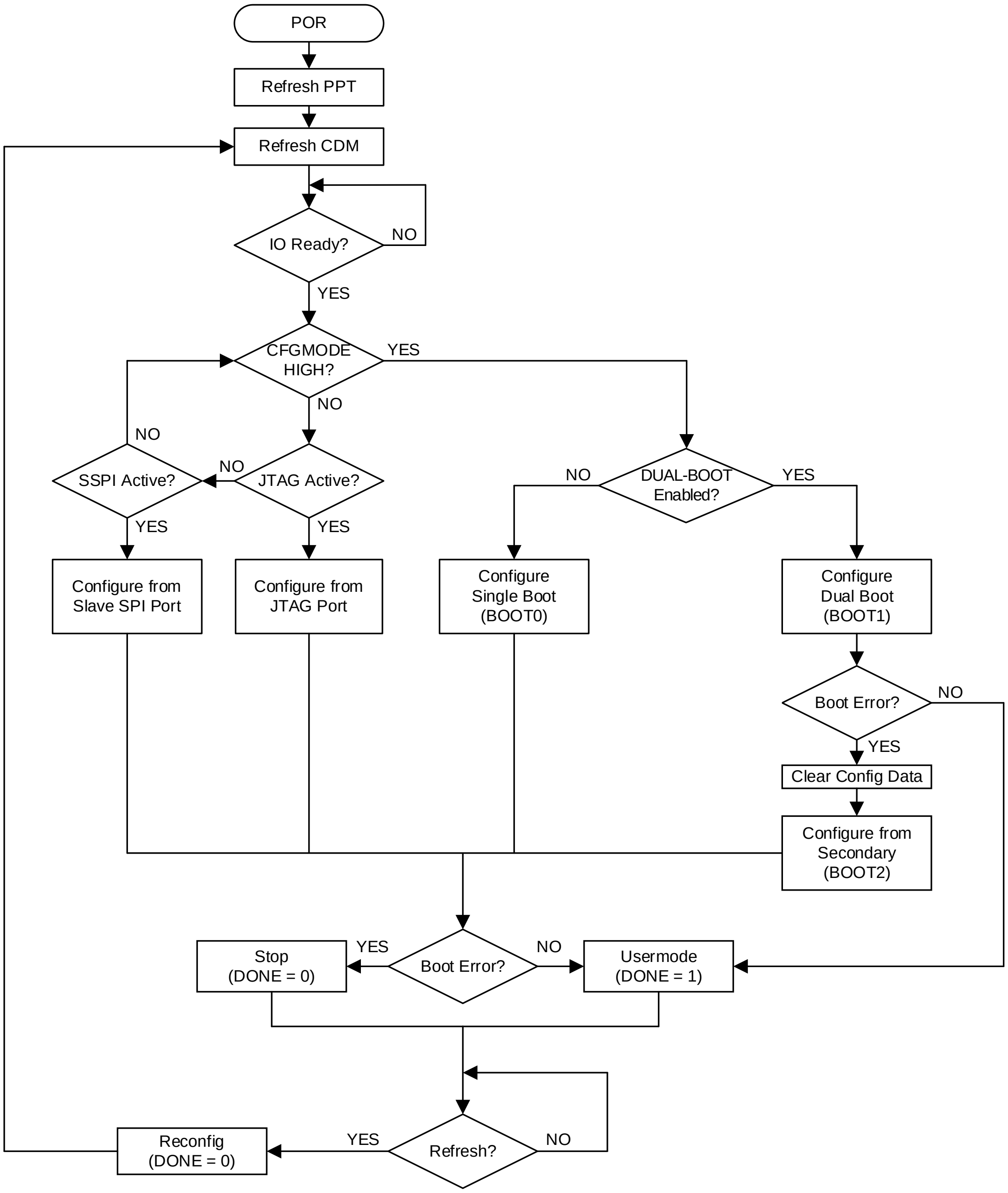
At Power Up (POR), PROGRAMN pin toggle (falling edge), or REFRESH command execution, the configuration logic performs CDM (Erase CRAM, Reset INIT registers, and optionally erase EBR depending on settings in Control Register (CR1). After performing CDM if the CFGMODE pin is *HIGH*, the device is put into controller auto-booting mode and the device starts the download from an external SPI boot PROM. If the CFGMODE pin is *LOW* after performing CDM, the device waits for a target request from either the JTAG or SSPI port. A port can request exclusive access using the PORT_REQUEST command as described in the PORT REQUEST Command section.

During configuration, the controller/target mode always follows the state of the CFGMODE pin. If the CFGMODE pin is toggled low during controller SPI booting the boot will be aborted, and the device will be in target mode. The target device would see that the boot was aborted by reading the status register. If the target device wants to reconfigure the part, it should first issue a refresh command to clear the configuration memory contents.

Target mode can only be enabled if the CFGMODE pin state is *LOW*. If the CFGMODE pin is toggled *HIGH* during target configuration, the target configuration is aborted, and the device is put in controller mode. To start the boot process over the Controller SPI port, a refresh event would be triggered by toggling the PROGRAMN pin.

The PROGRAMN and CFGMODE PINS are dedicated pins. After entering User Mode any dedicated or persisted target port can access config regardless of the CFGMODE pin state. The Target SPI can be a user GPIO port and therefore must be persisted if needed in User Mode.

The device configuration control flow is shown in Figure 4.1.



Note: Refresh is either programn toggle or REFRESH instruction.

Figure 4.1. Configuration Control Flow

4.4. sysCONFIG Pins

The Lattice Avant devices provide a set of sysCONFIG I/O pins that are used to program and configure the FPGA. The sysCONFIG pins are grouped together to create ports (such as JTAG, SSPI, MSPI) that are used to interact with the FPGA for programming, configuration, and access of resources inside the FPGA, as shown in Figure 4.2.

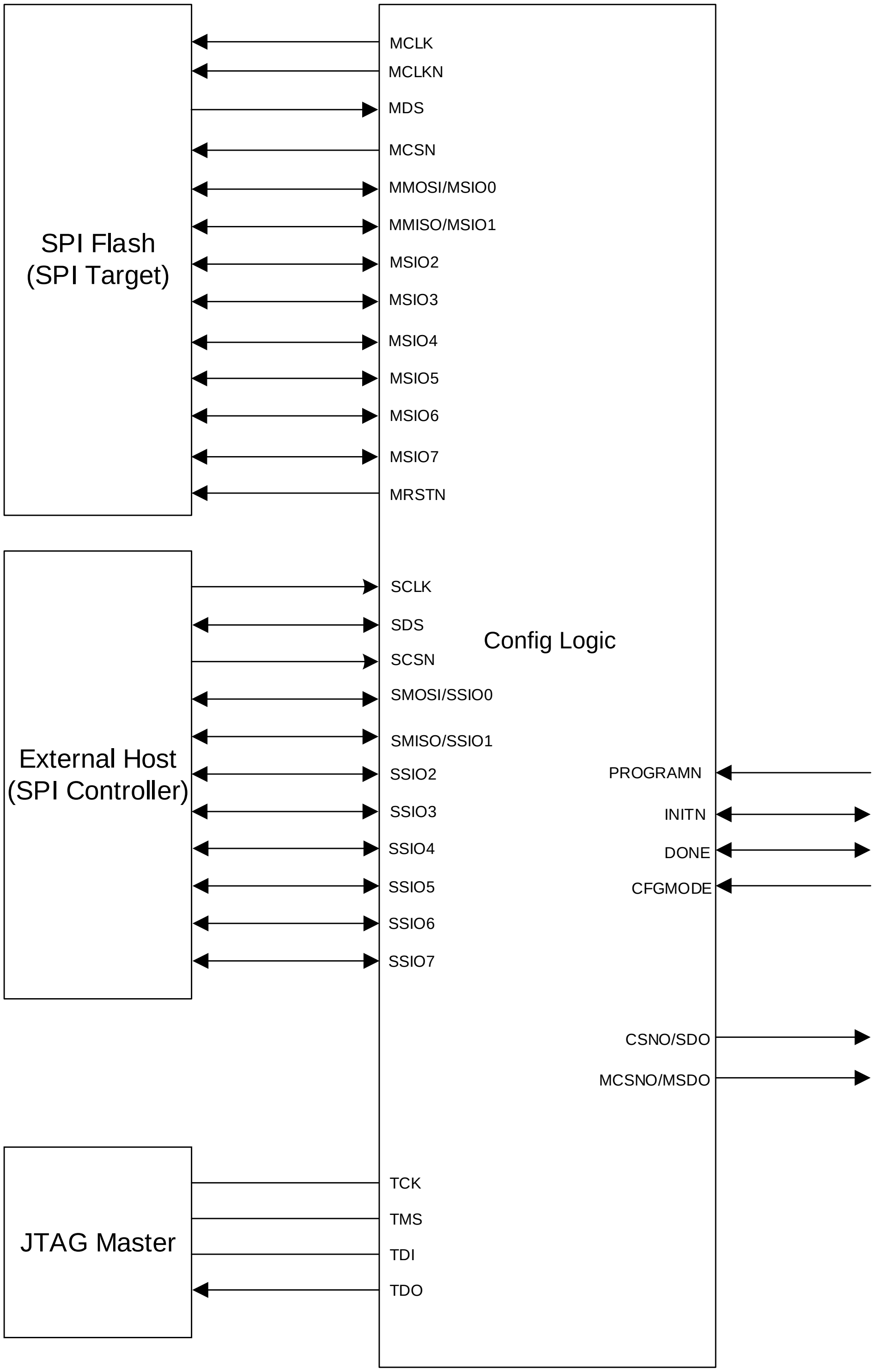


Figure 4.2. sysCONFIG Pins

The sysCONFIG pins in a configuration port group may be active, and used for programming the FPGA, or they can be reconfigured to act as general purpose I/O.

Recovering the configuration port pins for use as general purpose I/O requires adherence to the following guidelines:

- The unused port must be DISABLED. This can be accomplished by using the Lattice Radiant™ Device Constraint Editor under Global tab.
- External logic must be prevented from interfering with device programming.

Table 4.3 lists the shared sysCONFIG pins of the device, and the default state of these pins in user mode. After programming the Avant device, the default state of the SSPI sysCONFIG pins become general purpose I/O. This means that SSPI cannot be used to re-program the Lattice Avant devices when using the default sysCONFIG port settings. To retain the SSPI sysCONFIG pins in user mode, they must be ENABLED using the Lattice Radiant Device Constraint editor.

Unless specified otherwise, the sysCONFIG pins are powered by the VCCIO1 and VCCIO2 voltage. It is crucial for this to be taken into consideration when provisioning other logic attached to Bank 1 and Bank 2.

The function of each sysCONFIG pin is described in detail.

Table 4.3. Default State of the sysCONFIG Pins

Group	sysCONFIG Pins					Pull During Configuration	Configuration Modes		
	Name	Location	Type	Hardware Default	Software Default		MSPI	SSPI	JTAG
System	CFGMODE	Bank 2	Dedicated	CFGMODE	CFGMODE	UP	1’b1 1’bx	1’b0	1’b0
	PROGRAMN	Bank 2	Dedicated	PROGRAMN	PROGRAMN	UP	1’b1 1’b1	1’b0	1’b0
	INITN	Bank 2	Dedicated	INITN	INITN	UP	INITN		
	DONE	Bank 2	Dedicated	DONE	DONE	UP	DONE		
	MCSNO/MSDO	Bank 1	Shared[6]	MCSNO/MSDO	GPIO	UP	MCSNO/MSDO	—	—
	SCSNO/SSDO	Bank 2	Shared[6]	MCSNO/MSDO	GPIO	UP	—	SCSNO/SSDO	—
Controller SPI	MCLK[3]	Bank 1	Shared[6]	MCLK	GPIO	UP/DOWN[3]	MCLK	—	—
	MCLKN	Bank 1	Shared[6]	MCLKN	GPIO	UP/DOWN[4]		—	—
	MCSN[2]	Bank 1	Shared[6]	MCSN	GPIO	UP	MCSN	—	—
	MMOSI/MSIO0	Bank 1	Shared[6]	MOSI/MDO	GPIO	UP	MOSI/D0	—	—
	MMISO/MSIO1	Bank 1	Shared[6]	MISO/MD1	GPIO	UP	MISO/D1	—	—
	MSIO [2:7]	Bank 1	Shared[6]	MD2	GPIO	UP	D2	—	—
	MDS	Bank 1	Shared[6]	MD3	GPIO	DOWN	D3	—	—
	MRSTN	Bank 1	Shared[6]	MRSTN	GPOP	UP	MRSTN	—	—
Target SPI	SCLK	Bank 2	Shared[6]	SCLK	GPIO	UP/DOWN3	—	SCLK	—
	SCSN[1]	Bank 2	Shared[6]	SCSN	GPIO	UP	—	SCSN	—
	SMOSI/SSIO0	Bank 2	Shared[6]	SI/SD0	GPIO	UP	—	MOSI/D0	—
	SMISO/SSIO1	Bank 2	Shared[6]	SO/SD1	GPIO	UP	—	MISO/D1	—
	SSIO [2:7]	Bank 2	Shared[6]	SD2/SCL	GPIO	UP	—	D2	—
	SDS	Bank 2	Shared[6]	SD3/SDA	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. SCSN should have 4.7 kΩ pull-up resistor on-board for SSPI.
2. MCSN should have 4.7 kΩ pull-up on-board resistor for MSPI.
3. Inter 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. Inter 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. Shared between configuration and user GPIO. User selectable through Radiant Device Constraint Editor. Achieved through Nonvolatile EFUSE Feature setting.
6. Shared between configuration and user GPIO in User Function Mode. User-selectable through Radiant Device Constraint Editor. Achieved through SRAM fuse setting through bitstream.

4.4.1. 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 mode and starts a device configuration sequence at the Initialization phase, as described earlier. The PROGRAMN has a minimum pulse width assertion period ($t_{PROGRAMN}$) for it to be recognized by the FPGA. This minimum time is defined in the sysCONFIG Port Timing Specifications of Lattice Avant Platform - Specifications Data Sheet (FPGA-DS-02112).

Be aware of the following special cases when the PROGRAMN pin is active:

- If the device is currently being programmed via JTAG, PROGRAMN is ignored until the JTAG mode programming sequence is complete.
- Toggling the PROGRAMN pin during device configuration interrupts the process and restart the configuration cycle.
- PROGRAMN must not make a falling edge transition during the time the FPGA is in the Initialization state to avoid interrupting, restarting, or failing configuration. PROGRAMN must be asserted for a minimum low period of $t_{PROGRAMN}$ for it to be recognized by the FPGA. Failure to meet this requirement can cause the device to become non-operational, requiring power to be cycled.

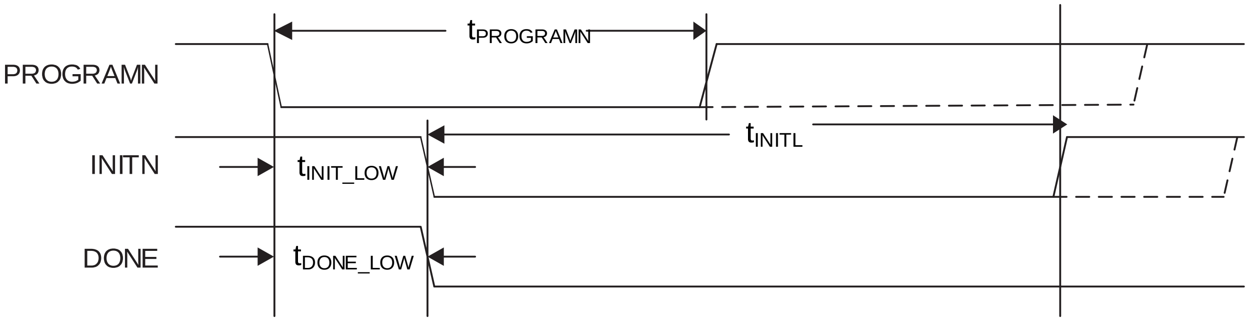


Figure 4.3. Configuration from PROGRAMN Timing

4.4.2. INITN

The INITN pin is a bidirectional open-drain control pin. It has the following functions:

- After power is applied, after a PROGRAMN assertion, or a REFRESH command it goes low to indicate the SRAM configuration memory is being erased. The low time assertion is specified with the t_{INIT_LOW} parameter.
- After the t_{INIT}L time period has elapsed the INITN pin is deserted (that is active high) to indicate the device is ready for its configuration bits. The device begins loading configuration data from an external SPI Flash.
- INITN can be asserted low by an external agent before the t_{INITL} time period has elapsed in order to prevent the FPGA from reading configuration bits. This is useful when there are multiple programmable devices chained together. The programmable device with the longest t_{INITL} time can hold all other devices in the chain from starting to get data until it is ready itself.
- The last function provided by INITN is to signal an error during the time configuration data is being read. Once t_{INITL} has elapsed and the INITN pin has gone high, any subsequent INITN assertion signals the device has detected an error during configuration.

The following conditions cause INITN to become active, indicating the Initialization state is active:

- Power has just been applied
- PROGRAMN falling edge occurred
- The REFRESH command is sent using a target configuration port (JTAG or SSPI). If the INITN pin is asserted due to an error condition, the error can be cleared by correcting the configuration bitstream and forcing the FPGA into the Initialization state.

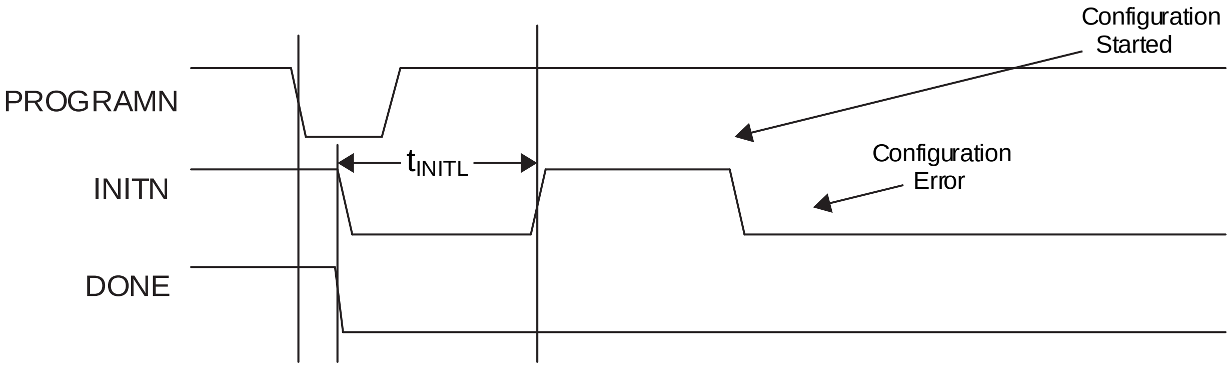


Figure 4.4. Configuration Error Notification

If an error is detected when reading the bitstream, INITN goes low, the internal DONE bit is not set, the DONE pin stays low, and the device does not wake up. The device configuration fails when the following happens:

- Failed (mismatch) on Device ID checking.
- The bitstream CRC error is detected.
- The invalid command error is detected.
- A preamble time out error is encountered when loading from the external Flash. This can occur when the device is in MSPI configuration mode and the SPI Flash device is not programmed.
- The program done command is not received when the end of on-chip SRAM configuration.

4.4.3. DONE

The DONE pin is a bi-directional open drain with a weak pull-up that signals the FPGA is in user mode. DONE is first able to indicate entry into user mode only after an internal DONE bit is asserted. The internal DONE bit defines the beginning of the FPGA Wake-Up state.

The FPGA can be held from entering User mode indefinitely by having an external agent keep the DONE pin asserted low. A common reason for keeping DONE driven low is to allow multiple FPGAs to be completely configured. As each FPGA reaches the DONE state, it is ready to begin operation. The last FPGA to configure can cause all FPGAs to start in unison.

The DONE pin drives low in tandem with the INITN pin when the FPGA enters Initialization mode. As described earlier, this condition happens when power is applied, PROGRAMN is asserted, or a Refresh command is received via an active target configuration port.

Sampling the DONE pin is a way for an external device to tell if the FPGA has finished configuration.

4.5. Controller SPI sysCONFIG Pins

The following is a list of the dual-purpose Controller SPI sysCONFIG pins. If any of these pins are used for configuration and for user I/O, the requirements listed at the start of the Configuration pin sections must be followed. These pins are powered by VCCIO1.

4.5.1. MCLK

The MCLK, when active, is a clock used to sequentially load the configuration data for the FPGA. When Controller Configuration mode is used, MCLK becomes an output clock with the programmed frequency. The output is used to drive to external memory device. The maximum MCLK frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing Specifications in Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112). MCLK actively drives until all the configuration data is received. When the device enters user mode the MCLK output tri-states. The MCLK is reserved for use in MSPI mode, in most post-configuration applications, as the reference clock for performing memory transactions with the external SPI PROM. See the Controller SPI Modes section for details.

The Avant device generates MCLK from an internal oscillator. The initial frequency of the MCLK is nominally 3.5 MHz. The MCLK frequency can be altered using the MCCLK_FREQ parameter. The MCCLK_FREQ parameter is selected using the Lattice Radiant Device Constraint Editor. For a complete list of the supported MCLK frequencies, see Table 4.4.

Table 4.4. Avant MCLK Valid Frequencies

MCLK Frequency (Based on 400 MHz/320 MHz) – Controller SPI
3.1
7.1
14.3
28.6
57.1
66.7
80.0
100.0
106.7
133.3
160.0

At startup, the lowest frequency MCLK 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 MCLK _FREQ value, the MCLK output begins driving the selected frequency. Make certain when selecting the MCCLK_FREQ not to exceed the frequency specification of the configuration memory or the PCB. When making MCCLK_FREQ decisions, review the sysCONFIG Port Timing specifications in Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

4.5.2. MCLKN

MCLKN – Invert of MCLK for differential clocking in Controller xSPI (Octal DDR) support. The remaining text and block diagrams in this document may only describe one clock (MCLK) signal for simplicity but, wherever MCLK is described or illustrated, a differential pair of MCLK, MCLKN may be used.

4.5.3. MCSN

MCSN – In MSPI mode, the MCSN becomes an active low Chip Select output that drives the SPI Serial Flash chip select. If SPI memory needs to be accessed using the SPI port while the part is in user mode (the DONE pin is high) then the MASTER_SPI_PORT= ENABLE, must be used to preserve this pin as MCSN. When the device is not in MSPI mode, the MCSN is a general purpose I/O with a weak pulldown. Adding a 4.7 kΩ to 10 kΩ pull-up resistor to MCSN pin on the Avant device is recommended. MCSN must ramp in tandem with the SPI PROM VCC input. It remains a general purpose I/O when the FPGA enters user mode.

4.5.4. MMOSI/MSIO0

MMOSI/MSIO0 – In Controller SPI configuration mode, the MOSI pin is the serial data output for SPI command and data. It becomes D0 of the data bus in Dual, Quad or Octal mode.

4.5.5. MMISO/MSIO1

MMISO/MSIO1 – In Controller SPI configuration mode, the MISO pin is the serial data input. It becomes D1 of the data bus in Dual, Quad or Octal mode.

4.5.6. MSIO [2:7]

MSIO [2:7] – In Controller SPI configuration mode, MSIO [2:3] becomes the D [2:3] of the data bus in Quad mode, or MSIO [2:7] becomes D [2:7] in Octal Mode.

4.5.7. MDS

MDS – In Controller xSPI configuration mode, this bit becomes the Data Strobe of the data bus in Octal mode.

4.5.8. MCSNO/MSDO

This is an output pin for configuration daisy chain support in Controller Mode, which has 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 part.

MSDO – MSDO pin is used in the Bypass mode. It will be the serial data output for the downstream device which supports the legacy SCM configuration mode.

4.6. Target SPI sysCONFIG Pins

4.6.1. SCLK

SCLK – In target SPI mode, this pin is the clock input for the target SPI configuration interface. The maximum CCLK frequency and the data setup/hold parameters can be found in the sysCONFIG Port Timing Specifications of Lattice Avant Platform – Specifications Data Sheet (FPGA-DS-02112).

4.6.2. SCSN

SCSN – In target SPI mode, this pin is the active low chip select input for the target SPI configuration interface. A 4.7 kΩ external pull-up resistor is recommended.

4.6.3. SMOSI/SSIO0

SMOSI/SSIO0 – In Target SPI configuration mode, the SI pin is the serial data input for SPI command and data. It becomes D0 of the data bus in Dual, Quad or Octal mode.

4.6.4. SMISO/SSIO1

SMIO/SSIO1 – In Target SPI configuration mode, the SO pin is the serial data output for SPI data. It becomes D1 of the data bus in Dual, Quad or Octal mode.

4.6.5. SSIO [2:7]

SD2 – In Target SPI configuration mode, SSIO [2:3] becomes the D [2:3] of the data bus in Quad mode, or SSIO [2:7] becomes D [2:7] in Octal Mode.

4.6.6. SDS

SDS – In Target xSPI configuration mode, this bit becomes the Data Strobe of the data bus in Octal mode.

4.6.7. SCNO/SSDO

This is an output pin for configuration daisy chain support in Target Mode, which has 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. The SCSNO of the first device drives the CSN of the second part.

SSDO – SSDO pin is used in the Bypass mode. It is the serial data output for the downstream device which supports the legacy SCM configuration mode.

4.7. JTAG Pins

4.7.1. TCK

TCK – If the JTAG port is enabled, this pin serves as the clock pin for the JTAG interface. The test clock pin (TCK) 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. The 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.

4.7.2. TMS

TMS – If the JTAG port is enabled, this pin serves as the Test Mode Select pin for the JTAG interface. The Test Mode Select (TMS) pin is an input pin that controls the progression through the 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 is present on TMS per the JTAG specification.

4.7.3. TDI

TDI – If the JTAG port is enabled, this pin serves as the Test Data Input (TDI) pin, which 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.

4.7.4. TDO

TDO – If the JTAG port is enabled, this pin serves as the Test Data Output (TDO) pin, which 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.

4.8. PERSISTENT

The internal PERSISTENT control bits are used to determine whether the dual-purpose Controller and Target sysCONFIG pins remain as sysCONFIG pins during normal operation, for example, to support transparent programming or configuration. The Lattice Avant devices have several PERSISTENT physical SRAM cells that determine the existence of the Controller SPI port or Target SPI port after entering user mode. These settings can be set in Lattice Radiant Device Constraint Editor under Global tab as shown in Table 4.5.

Table 4.5. sysCONFIG Pins Global Preferences[1]

Port Setting	Value	Pins Affected	Details
MASTER_SPI_PORT	SERIAL	MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1	If enabled, persisted for configuration purpose.
	DUAL	MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1	
	QUAD	MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1, MSIO [2:3]	
	OCTAL	MCLK, MCLKN, MCSN, MMOSI/MSIO0, MMISO/MSIO1, MSIO [2:7], MDS	
SLAVE_SPI_PORT	SERIAL	SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1	If enabled, persisted for configuration purpose.
	DUAL	SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1	
	QUAD	SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1, SSIO [2:3]	
	OCTAL	SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1, SSIO [2:7], SDS	

Note:

1. JTAG Persist follows JTAG enable pin.

5. Controller Configuration Process and Flow

Prior to becoming operational, the FPGA goes through a sequence of states, including initialization, configuration and wake-up.

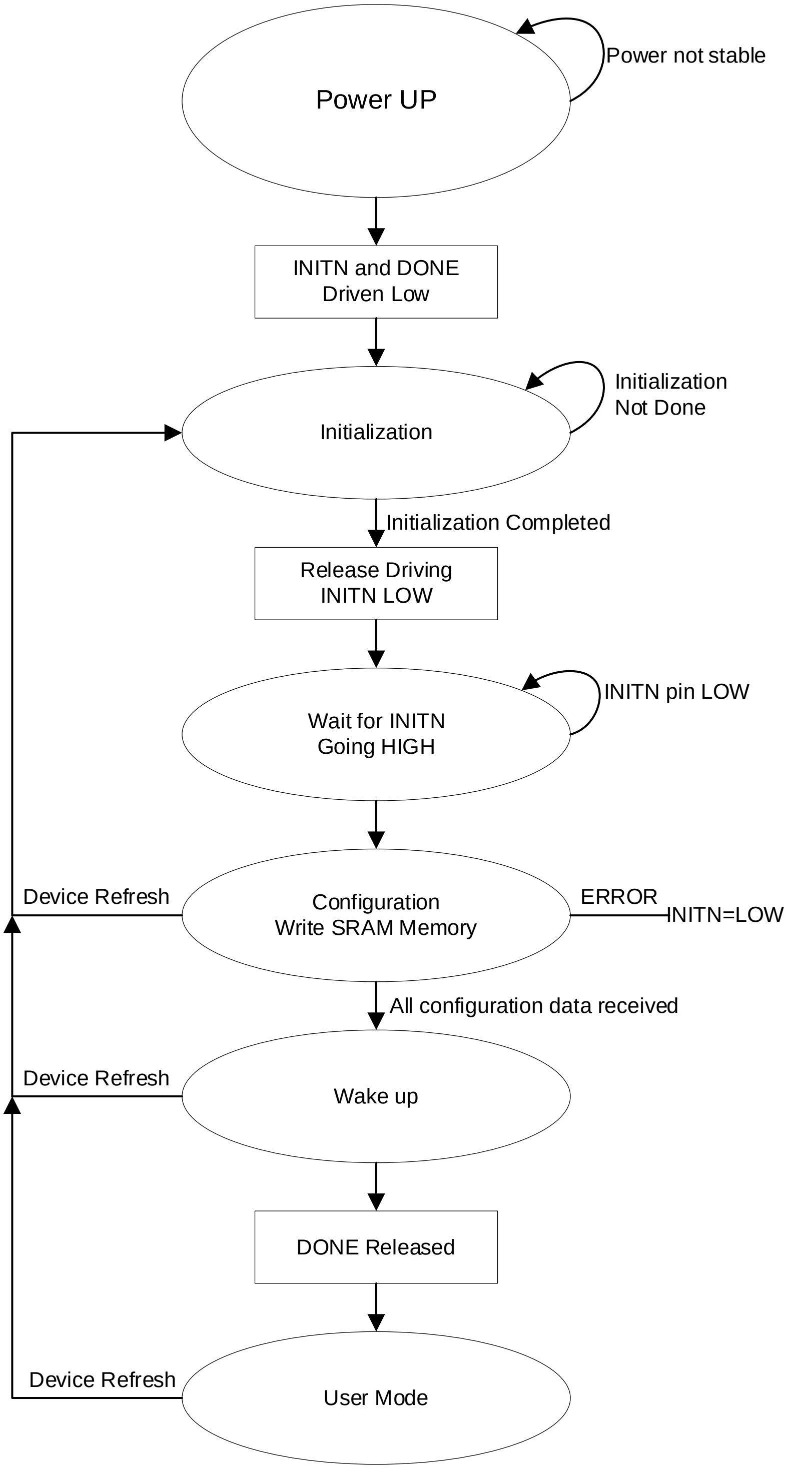


Figure 5.1. Controller Configuration Flow

The Lattice Avant device sysCONFIG ports provide industry standard communication protocols for programming and configuring the FPGA. Each of the protocols shown in Table 4.2 provides a way to access the configuration SRAM of the Lattice Avant device. The Configuration Ports Arbitration section provides information about the availability of each sysCONFIG port.

5.1. Power-up Sequence

In order for the Lattice Avant device to operate, power must be applied to the device. During a short period of time, as the voltages applied to the system rise, the FPGA stays in an indeterminate state.

As power continues to ramp, a Power-On-Reset (POR) circuit inside the FPGA becomes active. The POR circuit, once active, makes sure the external I/O pins are in a high-impedance state. It also monitors the VCC, VCCAUX, VCCIO0, and VCCIO1 input rails. When those power supplies reach the minimal operation level internally, the POR circuit releases an internal reset strobe, allowing the device to begin its initialization process. The Avant device asserts INITN active low, and drives DONE low. When INITN and DONE are asserted low, the device moves to the initialization state, as shown in Figure 5.1.

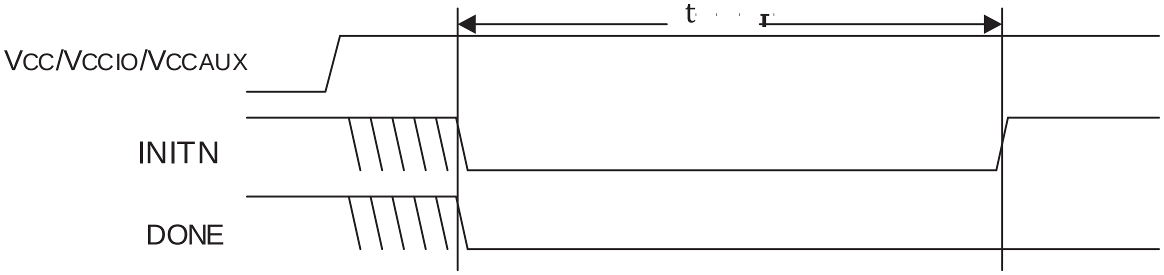


Figure 5.2. Configuration from Power-On-Reset Timing

5.2. Initialization

The Avant device enters the memory initialization phase immediately after the Power On Reset circuit drives the INITN and DONE status pins low. The purpose of the initialization state is to clear all the SRAM memory inside the FPGA.

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

- The t_{INITL} time period has elapsed.
- The PROGRAMN pin is deasserted.[1]
- The INITN pin is no longer asserted low by an external controller.

Note:

1. This condition is ignored by the Avant-AT-E70B devices. If desired, use INITN assertion (as described above) to delay the start of the configuration phase.

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

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

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, one or more INITN pins should be wire-ANDed. If one or more FPGAs or an external device holds INITN low, the FPGA remains in the initialization state.

The GPIO of the device at power-up defaults to tri-stated outputs with active weak pull-downs. After configuration, all GPIO included in the user design wake up in the user-defined condition. GPIO not defined in the user design remain output tri-stated and the input with a weak pull-down enabled. 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.

Before/during configuration, the dual-purpose sysCONFIG I/O have pull condition specified in Table 4.3, which are excluded from the GPIO default setting.

5.3. Configuration

The releasing HIGH on the INITN pin causes the FPGA to enter the configuration state. The FPGA can accept the configuration bitstream created by the Lattice Radiant Software.

Following power-up, the Avant device begins the external SPI flash boot process with the Signature Verification phase. The Avant 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 Avant device to proceed to the Preamble Verification phase immediately. Opposed to waiting for a fixed power-ramp timer to expire, this process allows for the fastest possible boot times.

The Signature Verification process verifies either the Lattice Specified LSCC signature or JEDEC Standard SFDP, depending on the value of Control Register 1, Bit 12 (CR1 [12]). If CR1 [12] is *0* (default), the Avant device reads the boot bitstream image from the base boot address (default is 0x00 but can be changed through OTP settings) and check for the LSCC signature (0x4C534343) using SPI flash command code 8'H03. If CR1 [12] is *1*, the Lattice Avant device performs a SFDP read (SPI flash command code 8'H5A) and checking for SFDP code (0x50444653) contained in SFPD compliant SPI flash devices. The Lattice Avant device retries the SFDP/LSCC signature read until three consecutive matches are found. When successful, the Lattice Avant device sets the internal Signature Successful Flag and proceeds to the Preamble Verification step. If the loop timer expires (600,000 SPI clock cycles, or about 150 ms) the device proceeds to the Preamble Verifications step without setting the Signature Successful Flag. When the internal Signature Successful Flag is set, the Signature Verification phase is bypassed for subsequent warm-boot events (for example, the PROGRAMN pin toggle or REFRESH command).

For proper bitstream data alignment, the bitstream Preamble must be detected once. If the preamble does not come before the preamble timer (counting for 600,000 SPI clock cycles) 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 memory. The Avant device does not leave the Configuration state if there is no valid configuration data.

INITN is used to indicate an error exists in the configuration data. When INITN is high, configuration is proceeding without issue. If INITN is asserted low, an error has occurred, and the FPGA does not operate.

5.4. Wake-up

Wake-up is the transition from configuration mode to user mode. The Lattice Avant device's fixed four-phase wake-up sequence starts when the device has correctly received all its configuration data. The order of the External DONE release is configurable to meet specific implementation requirements. When all configuration data is received, the FPGA asserts an internal DONE status bit. The assertion of the internal DONE causes a Wake-Up state machine to run those sequences four controls. The four control strobes are:

- Global Set/Reset (GSRN)
- Global Write Enable (GWE)
- Global Output Enable (GOE)
- External DONE

One phase of the Wake-Up process is for the FPGA to release the Global Output Enable. When it is asserted, permits the FPGA's I/O to exit a high-impedance state and take on their programmed output function. The FPGA inputs are always active. The input signals are prevented from performing any action on the FPGA flip-flops by the assertion of the Global Set/Reset (GSRN).

Other phases of the Wake-Up process release the Global Set/Reset and the Global Write Enable controls.

The Global Set/Reset is an internal strobe that, when asserted, 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/cleared per their hardware description language definition.

The Global Write Enable is a control that overrides the write enable strobe for all RAM logic inside the FPGA. As mentioned previously, the inputs on the FPGA are always active. Keeping GWE deasserted prevents accidental corruption of the instantiated RAM resources inside the FPGA.

Another phase of the Wake-Up process asserts the external DONE pin. The external DONE is a bi-directional, open-drain I/O only when it is enabled. An external agent that holds the external DONE pin low prevents the wake-up process of the FPGA from proceeding. Only after the external DONE, if enabled, is active high does the final wake-up phase complete. Wake-Up completes uninterrupted when the external DONE pin is not enabled.

Once the final wake-up phase is complete, the FPGA enters user mode. This process is detailed later in the document.

5.5. Early I/O Release

The Avant device supports an Early I/O Release feature, which allows the I/O that reside in the I/O Banks of the device, to assume user-defined drive states at the beginning of bitstream processing. The Early I/O Release feature releases the I/O after processing the I/O configuration for the left and right banks, which is located near the head of the bitstream data. Once data is programmed in the Memory Interface Block (MIB) the I/O is released to a predefined state. This feature is enabled by setting the EARLY_IO_RELEASE preferences to ON in 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 in Early-Release banks drive High-Z until full device configuration is complete. Be aware that some of the I/O in Bank 1, including the dual-purpose sysCONFIG I/O, cannot be utilized as Early Released I/O. Also, if the ECDSA bitstream authentication is enable for the Avant device, the Early I/O Release feature is not supported.

5.6. User Mode

The Avant device enters user mode immediately following the Wake-Up sequence has completed. User Mode is the point in time when the device begins performing the logic operations programmed by the user design. The device remains in this state until one of three events occurs:

- The PROGRAMN input pin is asserted.
- A REFRESH command is received via one of the configuration ports.
- Power is cycled or power supply levels drop below their specified trigger levels.
- User Watchdog Timer trigger

5.7. Clearing the Configuration Memory and Re-initialization

The current user mode configuration of the device remains in operation until they are actively cleared, or power is lost. Several methods are available to clear the internal configuration memory of the Lattice Avant device.

- Remove power and reapply power.
- Toggle the PROGRAMN pin.
- Execute the Refresh command. Any active configuration port can be used to send a Refresh command.

Invoking one of these methods causes the Avant device to drive INITN and DONE low. The device enters the initialization state as described earlier or Erase command is executed.

6. Device Configuration

The Lattice Avant device provides multiple options for loading the configuration SRAM from a non-volatile memory and provide target configuration capability for device setup program and configuration memory programming. The previous section describes the physical interface necessary to interact with the configuration logic of the Avant device. This section focuses on describing the functionality of each of the different configuration modes. Descriptions of important settings required in the Lattice Radiant Device Constraint Editor View are also discussed.

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 the well-defined configuration states.

6.1.1. Configuration Modal States Definition

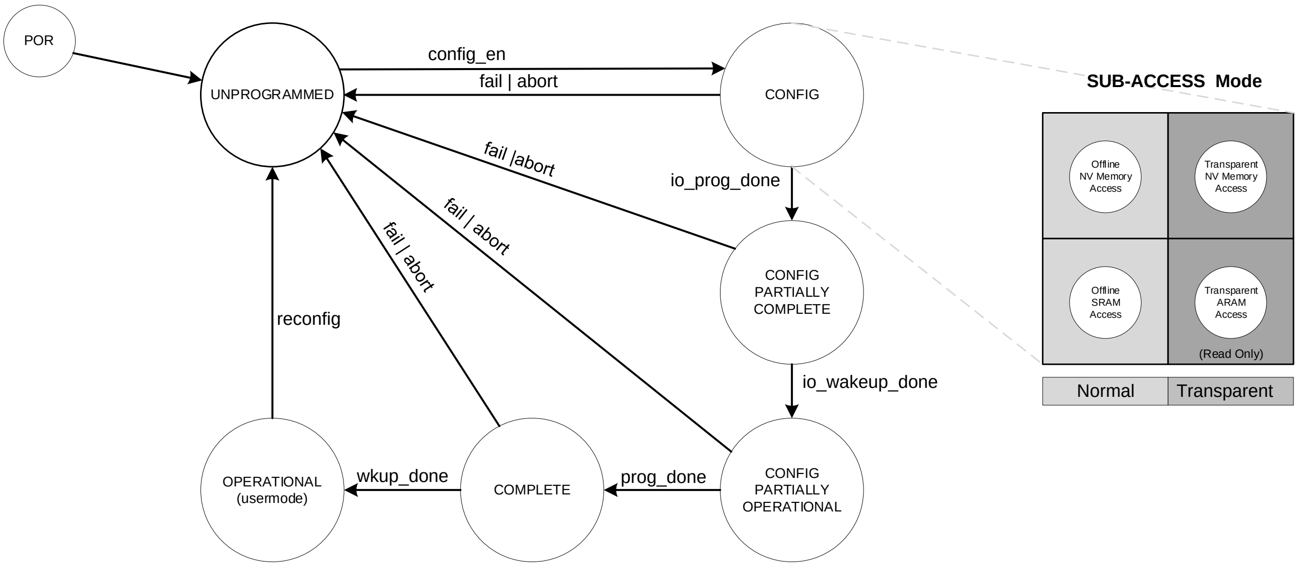


Figure 6.1. Configuration Modal States

- Unprogrammed – In this state the device may be blank or incompletely programmed. In this state the device's system I/O pins are disabled (Tri-Stated) such that they cannot contend with output drivers that may exist in other devices connected to our device. Any program qualified will have its operation nullified if executed in this state. Upon power-up, volatile devices, blank devices, and incompletely programmed nonvolatile 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 will 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 wakeup 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. In this state the device's system I/O pins take on their programmed values high, low, 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. Successfully programmed nonvolatile device, after automatic self-booting, should reside in this state. OTP Programming can be done in this state. Transparent programming can be done in this state.

6.1.2. Sub-Access Mode for Non-Volatile Memory Access

When accessing the NV Memory, all read command execution will capture the shadow register data for readout if the targeted NV Memory content has shadow register associated with it. Shadow registers can be modified with the PROG_OTP_SHADOW command depending on row access policy of the OTP.

6.2. Configuration Command Support

All commands except 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).

6.2.1. Command Format

The configuration command format is shown in Figure 6.2.

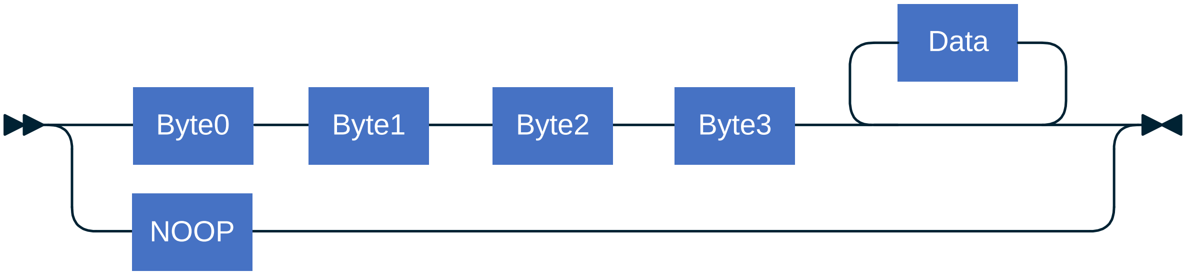


Figure 6.2. Configuration Command Format

6.2.2. Command Fields Definition

The configuration command fields are defined in Table 6.1.

Table 6.1. Command Fields Definition

Byte0								Byte1	Byte2	Byte3	Data (N Bytes)
7	6	5	4	3	2	1	0				
0	Opcode							Operand1	Operand2	Operand3	Optional
1	0	0	0	0	RD	Nbytes[9:8]		Nbytes[7:0]	Opcode	Operand	Optional

6.2.3. 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. The tables define the size of the result data for each command.

Because the latency for returning result data from a read command may be variable, each client interface (SSPI, JTAG) implements an interface-specific handshake protocol to indicate when the result data is present and valid.

6.2.3.1. JTAG Read

- Standard Read Commands
  - Host sends 4-byte read command through instruction shift register
  - The instruction shift register captures 0xFFFFFFFF until return data is ready.
  - Host shifts in NOOPs and observes captured data
  - Once return data is ready, shift register captures 0xFFFFFF00 and then captures the return data.
  - When return data is complete, the shift register captures 0xFFFFFFFF.
- CRAM Data Read
  - Return data behavior is same as Standard Read Commands
- Usage
  - Host observes captured data until it sees 0xFFFFFF00, then samples return data

  OR
  - Host can wait in run-test/idle state until maximum initial latency time is reached, then clocks at least one more time in run-test/idle before moving to select-DR-scan. Then captured data will be 0xFFFFFF00 followed by the return data.

6.2.3.2. SSPI Read

- Standard Read Commands
  - Controller can start sampling data immediately after command is transmitted.
  - SPI: Controller sends 0 on MOSI
  - DSPI/QSPI/xSPI: Controller tristate I/O pins
  - SPI: SSPI sends 1 on MISO until return data is ready
  - DSPI/QSPI/xSPI: SSPI tri-state I/O pins until return data is ready
    - Controller receives 0xFF because I/O pins are pulled up internally (weak pullups)
    - Data might be unstable for the first byte if clock is faster than the pullup time; Controller should ignore the first byte.
  - Minimum initial latency
    - SPI/DSPI/QSPI: minimum one byte of 0xFF output before data start byte (0x00)
    - xSPI: minimum of one full clock cycle before asserting DS signal and sending data start word (0xFF00)
  - When return data is ready,
    - SPI/DSPI/QSPI: output one byte of 0x00 before outputting data.
    - xSPI: output one word of 0xFF00 before outputting data.
  - SSPI tristate I/O pins when return data FIFO is empty or CSN is deserted.
- CRAM Data Read
  - Command is only supported in SPI (x1) mode. Not supported in DSPI / QSPI / xSPI.
  - Return data behavior is same as Standard Read Commands
- Usage
  - Commands with fixed initial latency:
    - Controller can ignore the return data until expected clock cycle for return data.
  - Commands with variable initial latency:
    - Controller can observe return data until it sees 0xFF -> 0x00 transition, then sample return data

    OR
    - Controller can send the read command, stretch the SSPI clock (hold it inactive) for a fixed amount of time (longer than the worst-case latency for the specified command), then start the SSPI clock again. Return data is valid at a fixed clock cycle after the clock resumes.

6.2.4. Configuration Command Table

6.2.4.1. Target Configuration Commands

Table 6.2. Target Configuration Commands

Command Name	Byte0	Byte1	Byte2	Byte3	# DATA BYTES	# RETURN BYTES	PROG Qualify	Description
BITSTREAM_BURST	0x05	0x00	0x00	0x00	—	—	Y	Configure the Device by burst in bitstream file in client mode
CRAM_READ_INC	0x0a	Level, Zone	NFRAMES [15:8]	NFRAMES [7:0]	—	variable	—	Read back the configuration memory data frames selected by the address register and post increment the address. # return bytes = $\sum_{i=1}^{NFRAMES}(DSRlen_i)$
CRAM_DATA_SHIFT	0x0b	Level, Zone	0x00	0x00	DSR_len	DSR_len	Y	Shift Data into and out of DSR for debugging
CRAM_READ_STATUS	0x0d	Level, Zone	0x00	0x00	2	4	—	Reads back status from CRAM zone controller. Used for debugging.
INIT_READ	0x0f	Mode	NFRAMES [15:8]	NFRAMES [7:0]	4	BYTES	—	Read back data through INIT Bus
READ_BUILD_ID	0x01	0x00	0x00	0x00	—	4	—	Read RTL Build Timestamp. For debug/emulation only.
READ_FW_REVISION	0x01	0xff	0x00	0x00	—	4	—	Read Config and Security firmware revision numbers. For debug/emulation only.
READ_IDCODE_PUB	0x01	0x01	0x00	0x00	—	4	—	Read 32-bit Public IDCODE of the device
READ_IDCODE_PRV	0x01	0x02	0x00	0x00	—	4	—	Read 32-bit Private IDCODE for the device
READ_UIDCODE_PUB_L	0x01	0x03	0x00	0x00	—	4	—	Read bits [31:0] of UNIQUE_ID[63:0]
READ_UIDCODE_PUB_H	0x01	0x04	0x00	0x00	—	4	—	Read bits [63:32] of UNIQUE_ID[63:0]
READ_SECURE_UID	0x84	0x00	0x28	0x00	—	8	—	Read the device 256-bit Secure Unique ID
READ_USERCODE	0x01	0x05	0x00	0x00	—	4	—	Read 32-bit User Code Register
READ_DR_USERCODE	0x01	0x06	0x00	0x00	—	4	Y	Read Dry-Run User Code Shadow Register
READ_STATUS0	0x01	0x07	0x00	0x00	—	4	—	Read Status Register 0
READ_STATUS1	0x01	0x08	0x00	0x00	—	4	—	Read Status Register 1
READ_STATUS2	0x01	0x09	0x00	0x00	—	4	—	Read Status Register 2
READ_STATUS3	0x01	0x0a	0x00	0x00	—	4	—	Read Status Register 3
READ_STATUS4	0x01	0x0e	0x00	0x00	—	4	—	Read Status Register 4
CHECK_BUSY	0x01	0x0b	0x00	0x00	—	4	—	Read Busy flag from Status Register 0
DEVICE_CTRL	0x80	0x00	0x01	CMD	—	—	—	Device Control
READ_CNTRL0	0x84	0x00	0x04	0x00	—	4	—	Read Control Register 0
READ_CNTRL1	0x84	0x00	0x06	0x00	—	4	—	Read Control Register 1
READ_UMR	0x84	0x00	0x08	0x00	—	16	—	Read User Mode Configuration Register
DRY_RUN_CTRL	0x80	0x00	0x09	MODE	—	—	—	Dry Run Control
REFRESH	0x80	0x00	0x0a	0x00	—	—	—	Equivalent to toggle the PROGRAMN pin.
SSPI_MODE	0x80	0x00	0x0f	MODE	—	—	—	Change Target SPI Mode
MSPI_BRIDGE	0x16	0x00	0x00	0x00	variable	—	—	Bridge Data to Controller SPI
PORT_REQUEST	0x7a	0x00	0x00	TYPE	—	—	—	Port Ownership Request
PORT_STATUS_READ	0x7b	0x00	0x00	0x00	—	4	—	Port Ownership Status
READ_SED_CRC	0x01	0x0c	0x00	0x00	—	4	Y	Read expected SED 32-bit CRC
READ_SED_CRC_CALC	0x01	0x0d	0x00	0x00	—	4	Y	Read 32-bit SED CRC calculated by the device
BBRAM_ERASE_ALL	0x80	0x00	0x4B	0x00	—	—	Y	Erase All BBRAM Data
READ_OTP	0x84	0x00	0x4E	ROW	—	4	—	Read User OTP

6.2.4.2. PORT REQUEST Command

The PORT_REQUEST command Requests exclusive access for the port it iss sent to or releases exclusive access. This command is supported by all client interfaces (SSPI, JTAG, and LMMI). When a port is granted exclusive access, all input data from other client interfaces is ignored.

The PORT_REQUEST command format and operand description are shown in Table 6.3 and Table 6.4.

Table 6.3. PORT_REQUEST Command Format

Byte0	Byte1	Byte2	Byte3
Opcode	**Operand1**	**Operand2**	**Operand3**
0x7A	0x00	0x00	REQ_Option

Table 6.4. PORT_REQUEST Command Operand Definition

Operand3	Bits	Description
REQ_Option	[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

6.2.4.3. MSPI_MODE Command

The MSPI_MODE command is a bitstream only command. This command selects the mode and frequency to boot from Controller SPI (MSPI). MSPI will always come up in x1 mode and can switch modes with this command. If the requested MODE is the same as the current mode, then this command acts like a NOOP. If the requested mode is different than the current mode, then the device stops reading from MSPI, optionally sends a mode transition command to the flash device, changes MSPI to the new mode, and then executes a REFRESH command internally.

If CMD_LEN is in the range 1 to 9, then the device sends the WREN command (0x06) to the flash device in one SPI transaction, followed by CMD_LEN number of command bytes from CMD_DATA0-8 in a second transaction. If CMD_LEN is 0 or is greater than 9, then CMD_DATA0-8 is ignored, and no commands are sent to the flash device.

The MSPI_MODE command format and operand description are shown in Table 6.5 and Table 6.6.

Table 6.5. MSPI_MODE Command Format

Byte0	Byte1	Byte2	Byte3	Data
Type/Rd/Nbytes[9:8]	Nbytes[7:0]	Opcode	Operand	
0x80	0x0C	0x0B	0x00	MODE[15:8], MODE[7:0], CMD_LEN, CMD_DATA0, …, CMD_DATA8

Table 6.6. MSPI_MODE Command Data Field Definition

Data	Bits	Description
MODE	[15:9]	SPI Mode 0: Slow Read: 1s-1s-1s (default) 1: Fast Read: 1s-1s-1s 2: Dual Read: 1s-2s-2s 3: Quad Read: 1s-4s-4s 4: xSPI Profile 1 Read: 8d-8d-8d 5: xSPI Profile 2 Read; 8d-8d-8d
	[8]	Clock Mode 0: Single-ended clock (SPI, DSPI, QSPI, xSPI) 1: Differential clock (xSPI only)
	[7:2]	Clock divider (Setup by Radiant)
	[1:0]	Clock Source (Fosc) 00: 400 MHz 01: 320 MHz 10: 266 MHz 11: 213 MHz)
CMD_LEN	[7:0]	Command Data Length (0-9)
CMD_DATA0-8	[7:0]	Command Data to send to flash device for mode transition

6.2.4.4. SSPI_MODE Command

The SSPI_MODE command changes the Target SPI operating mode.

The SSPI_MODE command format and operand description are shown in Table 6.7 and Table 6.8.

Table 6.7. SSPI_MODE Command Format

Byte0	Byte1	Byte2	Byte3
Type/Rd/Nbytes[9:8]	Nbytes[7:0]	Opcode	Operand
0x80	0x00	0x0F	OPR

Table 6.8. SSPI_MODE Command Operand Definition

Data	Bits	Description
OPR	7:2]	Reserved
	[1:0]	Target SPI Mode 00 = x1 (SPI) 01 = x2 (DSPI) 10 = x4 (QSPI) 11 = x8DDR (xSPI 8D-8D-8D)

6.3. Controller SPI Modes

The Controller SPI port is considered an intelligent port and it is the default mode before any other target configuration port is activated. It can perform read and write actions based on the command shifted into the device. All commands are built inside the bitstreams. In Controller SPI mode (MSPI), the Avant device begins retrieving configuration data from the SPI Flash when power is applied, a REFRESH command is received, or the PROGRAMN pin is asserted and released (or simply asserted, with the Avant-AT-E70B variant). The MCLK I/O takes on the Controller Clock (MCLK) function and begins driving a nominal 3.5 MHz clock to the SPI Flash's SCLK input. MCSN is asserted low, commands are transmitted to the PROM over the MMOSI/MSIO0 output, and data is read from the PROM on the MMISO/MSIO1 input pin. When all the configuration data is retrieved from the PROM, the MCSN pin is deserted, and the MSPI output pins are tri-stated.

Table 6.9. Controller SPI Configuration Port Pins

Pin Name	Function	Direction	Description
MCLK	CLK	Output	Controller clock used to time data transmission/reception from the Avant device Configuration Logic to a target SPI PROM.
MCLKN	MCLKN	Output	Optional complementary Controller clock output for differential clock support in xSPI (OCTAL, DDR) mode.
MCSN[1]	CSN	Output	Chip select used to enable an external SPI PROM containing configuration data.
MMOSI/MSIO0	MOSI	Output	Carries output data from the Avant device Configuration Logic to the target SPI PROM.
	SIO0	Input	Input pin for bitstream data as DUAL, QUAD or OCTAL mode
MMISO/MSIO1	MISO	Input	Carries input data from the target SPI PROM to the Avant device Configuration Logic.
	SIO1	Input	Input pin for bitstream data as DUAL, QUAD or OCTAL mode.
MSIO [2:3]	SIO [2:3]	Input	This is the input pins for bitstream data from SPI Flash, used only in QUAD or OCTAL mode.
MSIO [4:7]	SIO [4:7]	Input	This is the input pins for bitstream data from xSPI Flash, used only in OCTAL mode.
MDS	DS	Input/Output	xSPI Data Strobe

Note:

1. Use 4.7 kΩ pull-up resistor.

The MCLK frequency always starts downloading the configuration data at the nominal 3.5 MHz frequency. The MCCLK_FREQ parameter, accessed using Device Constraint Editor, can be used to increase the configuration frequency. The configuration data in the PROM has some padding bits, and then the data altering the MCLK base frequency is read. The Avant device reads the remaining configuration data bytes using the new MCLK frequency. For higher MCLK frequency with fast slew rate, a 33 Ω damping resistor is recommended.

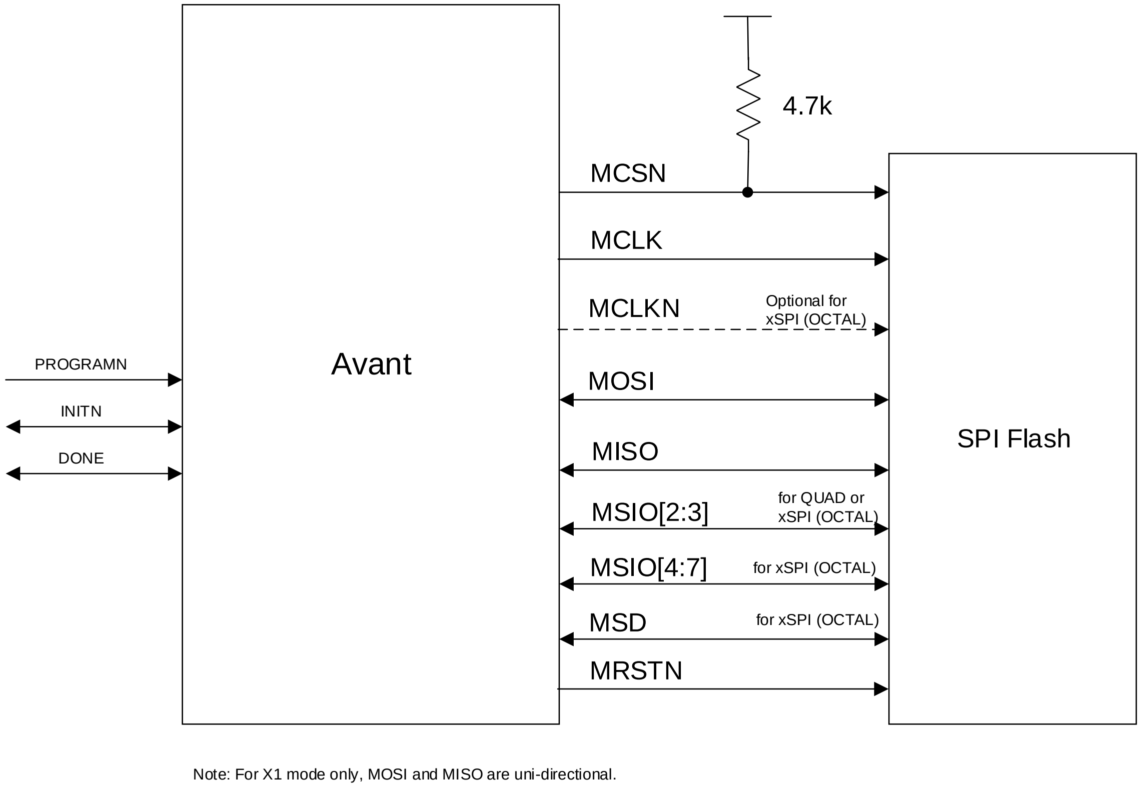


Figure 6.3. Avant Controller SPI Port with SPI Flash

Once the SPI Flash contains the configuration data, the configuration can be tested by either asserting the PROGRAMN pin, transmitting a REFRESH command, or cycling power to the board. The Lattice Avant device then configures from the external SPI Flash.

6.3.1. Method to Enable the Controller SPI Port

The Controller SPI port is enabled by driving CFGMODE pin high.

- When the device is powered up, or when the PROGRAMN pin is toggled, or when the REFRESH command is executed, the Controller SPI port is selected as the configuration port by default. A port is said to be a configuration port when it can execute both bitstream write, read commands, and port control commands which could invoke DUAL, QUAD and OCTAL read mode from SPI Flash.
- Enabling Controller SPI port persistence
  The MSPI port could be persisted in user mode by setting the desired Value (SERIAL, DUAL, QUAD, or OCTAL) for MASTER_SPI_PORT in 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 persistent bits to determine if the Controller SPI port is to remain operational once in user mode. This selection is independent of the CFG port arbitration during configuration phase. A port enabled by persistence is a Transparent Mode port. It reserves those pins from being occupied by user logic.
  Note that both the DONE pin and the INITN pin must be high. If not, then the device is not in user mode. The persistent bits have no affect when the device is not in user mode.

6.3.2. Dual-Boot and Multi-Boot Configuration Modes

Both the primary and the golden (fail-safe) configuration data is stored in external SPI memory. The fail-safe pattern is available in case the primary pattern would fail to load. The primary image can fail in one of the following reasons:

- A time-out error is encountered while waiting for PREAMBLE code.
- Device ID checking failed at the beginning of the bitstream.
- An illegal command is encountered.
- A bitstream CRC error is detected.
- A time-out error is encountered when loading from the primary pattern stored in the external memory.

A CRC error is caused by incorrect data being written into the SRAM configuration memory. Data is read from the external 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 external golden pattern.

It is possible for the data to be correct from a CRC calculation perspective, but not be functionally correct. In this instance the internal DONE bit never becomes active. The device counts the number of controller clock pulses it provided after the Power On Reset signal is released. When the count expires without DONE becoming active, the FPGA attempts to get its configuration data from the external golden pattern.

Dual boot configuration mode typically requires two configuration data files. One of the two configuration data files is a fail-safe image that is rarely, if ever, updated. The second configuration data file is a working image that is routinely updated. Both the working (primary) image and the fail-safe image are stored in the external SPI memory. One Lattice Radiant project can be used to create both the working and the fail-safe configuration data files. 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 MSPI mode. If the primary pattern 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 a SPI Flash device to be split into two sections, the first containing a *golden boot* file, and a second updatable *primary boot* file, which can be erased and reprogrammed. By default, the FPGA loads the primary boot file in block 0 (0x000000), can be changed through OTP setting). If the FPGA fails in configuration, it automatically loads the golden boot file in the last block (0xFFFF00, can be changed through PROG_SEC_BOOT command). 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 program correctly.

Up to five different design revisions can be stored in external Flash. Multi-boot, or the ability to dynamically reconfigure from multiple design bitstreams, is similar to the dual-boot where there is one primary (working) bitstream and up to four alternate bitstreams.

For multi-boot operation, the next target address is set in memory that was loaded during the current configuration memory load. Initiating reprogramming by toggling the PROGRAMN pin or issuing a REFRESH via any sysCONFIG port causes the device to load from the defined SPI Flash address. For detail information regarding multi-boot operation, refer to Lattice Avant Multi-Boot User Guide (FPGA-TN-02314).

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

6.3.3. Ping-Pong Boot

The Ping-Pong boot mode utilize the JUMP table to select an image for booting without changing 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 were to become corrupted by a power failure. A backup JUMP instruction should be programmed at offset 0xFFFF00 pointing to the golden boot image. Refer to Figure 6.4 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 offset 0xFFFF00, and the primary bitstream is selected with the JUMP command in the jump table. In order to swap the order of booting these three instructions are all that needs to be re-programmed.

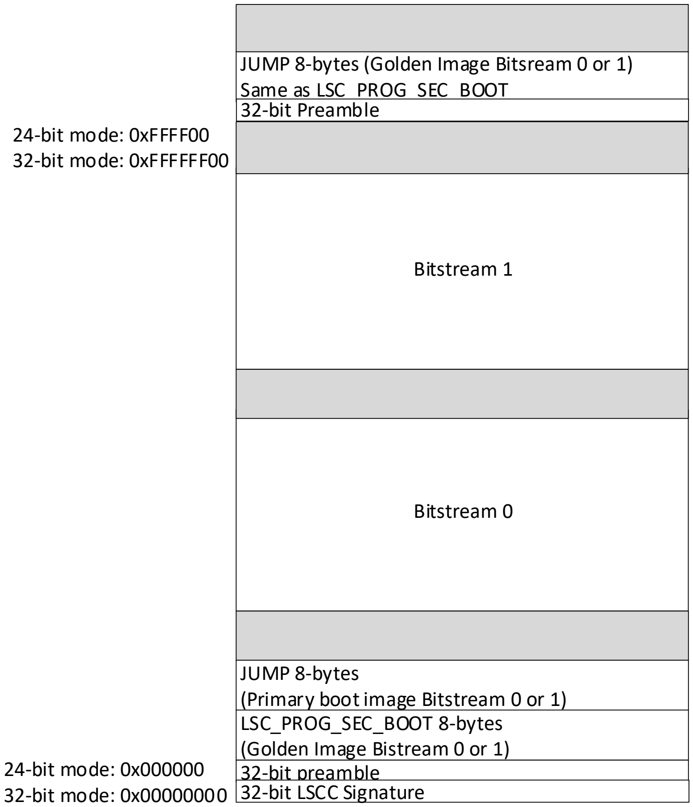


Figure 6.4. Jump Table

6.3.4. Dual, Quad and OCTAL Controller SPI Read Mode

The controller SPI configuration mode in the Avant device is expanded to support new industry standard Quad I/O SPI Flash memory and Octal I/O xSPI Flash memory. The support of (Serial Multi I/O) Flash memory enables fast parallel read.

A typical SPI Flash interface uses either 4 or 6 interface signals to the FPGA. The Standard SPI flash uses CLK, CS, SI, and SO while Quad SPI Flash uses CLK, CS, I/O0, I/O1, I/O2 and I/O3, maintaining function and pin-out compatibility with the standard SPI Flash devices, while adding Dual-I/O and Quad-I/O SPI capability. Also, the Avant device support the FLASH Memory with xSPI interface which could achieve 8bits (OCTAL) data throughput. All SPI modes are sub-modes of CONTROLLER SPI, therefore there is no longer a separate CFGMDN pin decode for each SPI mode.

To change the SPI read mode to fast read, dual, quad or octal read, the Deployment Tool must be used to generate the hex file used for programming the SPI flash device. Lattice Radiant flow only generates bitstreams with default SPI read mode, which is slow serial read (03h) mode. The sysCONFIG option in Lattice Radiant Device Constraint Editor is for the software to preserve appropriated pins to help customer design flow. It does not serve the purpose of enabling SPI port nor setting appropriate bits in the PROM file.

Inside the bitstream, the MSPI_MODE is utilized to invoke different sub-modes of CONTROLLER SPI. See the MSPI_MODE Command section for details.

6.4. Target SPI Mode

The Avant device provides a Target SPI (SSPI) 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. The external Flash memory updates are implemented using either offline or transparent operations. It is necessary to send a REFRESH command to load a new external Flash image into the configuration SRAM. When the device is in Transparent Mode, only read type commands for the Configuration SRAM are supported, allowing for device verification or debugging. The command format and definition are described in the Configuration Command Support section. The Target SPI port is a byte bounded port; all input and output data must be byte bounded.

In the Target SPI mode, the SCLK pin serves as Target SPI Clock. Input data is read into the device on the SMOSI pin at the rising edge of SCLK. Output data is valid on the SMISO pin at the falling edge of SCLK. The SCSN acts as the chip select signal. When SCSN is high, the SSPI interface is deselected and the SMISO pin is tri-stated. Commands can be written into, and data read from the device when SCSN is asserted.

The SSPI port can be used when CFGMODE is LOW before the device enters user mode. The SSPI port can be persisted in user mode by setting the desired Value (SERIAL, DUAL, QUAD or OCTAL) for SLAVE_SPI_PORT in Lattice Radiant Device Constraint Editor.

Using the SSPI port simplifies the Avant device configuration process. Lattice provides *C* source code called sspiembedded simplify the process of programming the Avant device. Refer to the Lattice Radiant online help about sspiembedded.

Table 6.10. Target SPI Configuration Port Pins

Pin name	Function	Direction	Description
SCLK	SCLK	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. SN is an active low input. High to Low transition: reset the device, prepare it to receive commands. Low to High transition: Completes or terminates the current command.
SMOSI/SSIO0	MOSI	Input	Carries output data from the external SPI controller to the Avant device Configuration Logic
	SIO0	Input/Output	D0 of the data bus for DUAL and QUAD mode
SMISO/SSIO1	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.
	SIO1	Input/Output	D1 of the data bus for DUAL and QUAD mode
SSIO [2:3]	SIO [2:3]	Input/Output	D [2:3] of the data bus for QUAD mode and OCTAL mode.
SSIO [4:7]	SIO [4:7]	Input/Output	D [4:7] of the data bus for OCTAL mode.
SDS	DS	Input/Output	xSPI Data Strobe

Note:

1. Use external 4.7 kΩ pull-up resistor.

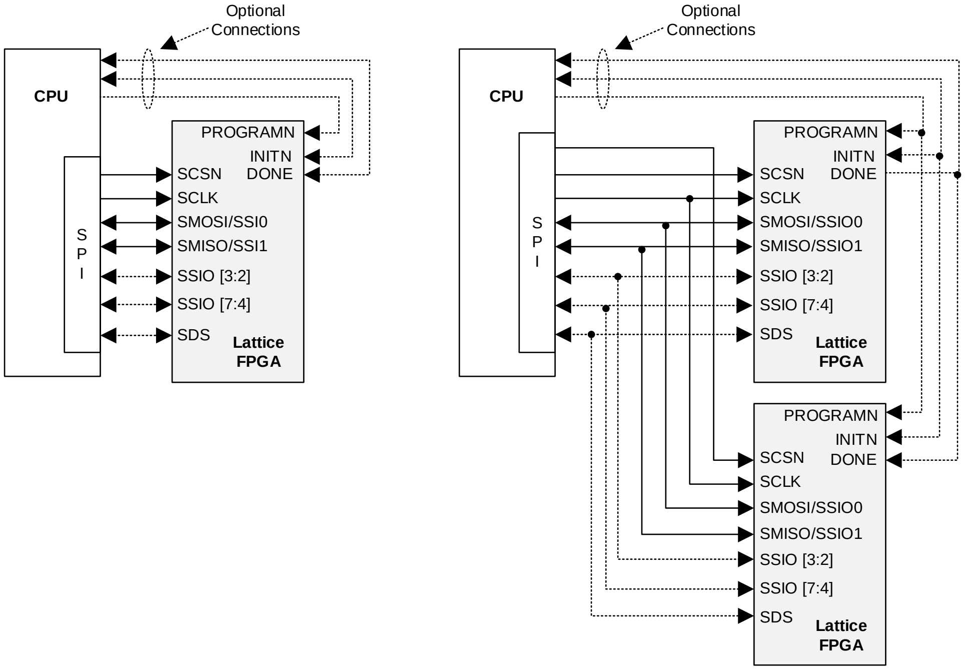


Figure 6.5. Avant Target SPI Port with CPU and Single or Multiple Devices

Notes:

- The dotted lines indicate optional connections.
- The wake-up time of the device does vary 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.

6.4.1. Method to Enable the Target SPI Port

The Target SPI port is enabled by drive CFGMODE pin low.

Similar to all configuration ports, the Target SPI port is enabled by the two standard methods:

- Enable the Target SPI Configuration Port in Configuration Mode.
- At Power Up or PROGRAMN pin toggle LOW (longer than tPROGRAMN) or REFRESH command execution, holding the CFGMODE low to stop the controller auto-booting event. Then drive the SCSN of the target SPI port and shift in the PORT_REQUEST command. See the PORT REQUEST Command section for details.
- Enabling Target SPI port persistence in user mode.
- The SSPI port could be persisted in user mode by setting the desired Value (SERIAL, DUAL or QUAD) for SLAVE_SPI_PORT in 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 persistent bits to determine if the TARGET SPI port is to remain operational once in user mode. This selection is independent of the CFG port arbitration during configuration phase. A port enabled by persistence is a Transparent Mode port. It reserves those pins from being occupied by user logic. To gain the configuration access to the device, the PORT_REQUEST command should be used. See the PORT REQUEST Command section for PORT_REQUEST command detail.
  Note that both the DONE pin and the INITN pin must be high to qualify the Target SPI port as a read back port. If not, then the device is not in user mode. The persistent bits have no affect when the device is not in user mode.

6.4.2. Specifications and Timing Diagrams – Target SPI Port Waveforms

Data and commands shift into the MOSI pin on the rising edge of clock. Data is shifted out of the MISO pin on the falling edge of clock. Only a read command causes the MISO pin to be enabled for data read out.

The Target SPI read and write waveforms are shown in Figure 6.6 and Figure 6.7.

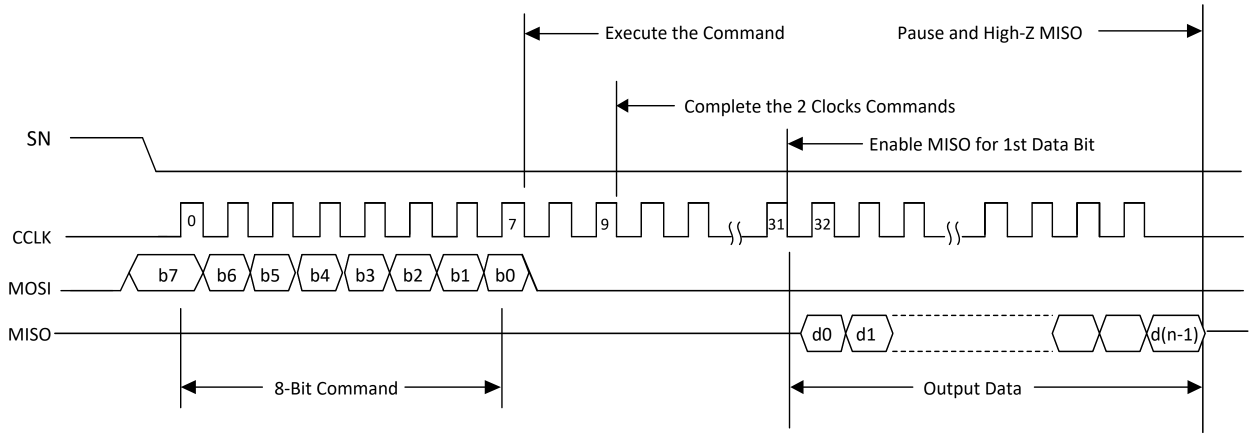
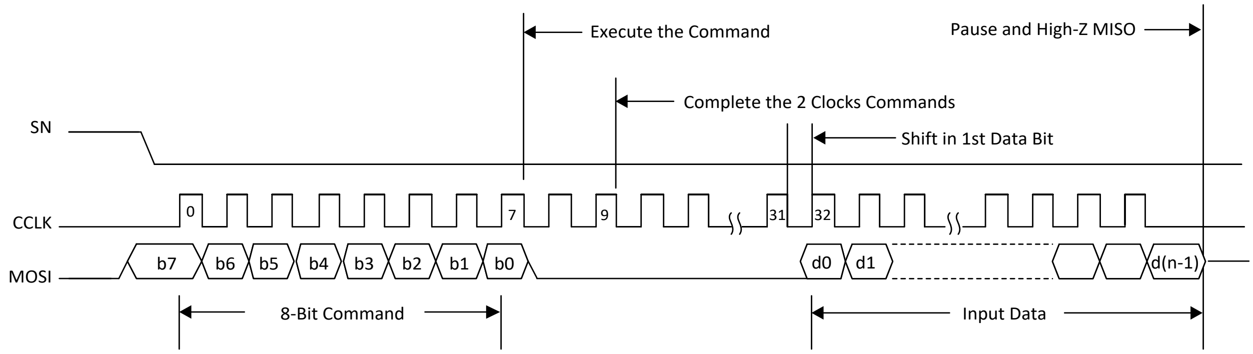


Figure 6.6. Target SPI Read Waveforms



Note: The bitstream is transferred starting with the first byte of the data file, starting with the MSB of the byte.

Figure 6.7. Target SPI Write Waveforms

6.4.3. Target SPI Port AC Timing Requirements

The Target SPI port maximum operation frequency requirement is shown in Table 6.11.

Table 6.11. Target SPI Port AC Timing Requirements

Description	Parameter	Min	Max	Unit
SCLK Frequency	f_{CCLK}	—	180	MHz

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

6.4.4. Dual, Quad Octal Target SPI Port

By default, the SPI port data width is x1 (by 1). That is, there is only one bit of input and one bit of output. However, to allow faster loading of configuration information, some devices support wider versions of the SPI interface. For this reason, the Target SPI configuration port of the Avant device also support x2 (Dual), x4 (Quad) and x8 (Octal, xSPI DDR) modes of operation.

For target SPI, the Dual, Quad and Octal modes are switched by SSPI_MODE command. See the SSPI_MODE Command section for SSPI_MODE command detail.

At POR or in the event of PROGRAMN pin toggle (low) or REFRESH command execution, the Target SPI port is reset to default serial (X1) mode.

6.4.5. Target SPI Configuration Flow Diagrams

The Target SPI port supports the regular Avant device bitstream and the encrypted bitstream (SSPI Mode).

The Target SPI configuration flow diagram is shown in Figure 6.8 and the highlights are listed below.

- The bitstream file is a stand-alone file. It is not part of the driver or system code. This provides seamless system integration and flexible file management. For example, the bitstream can be switched on the fly without changing a single line of system code.
- The Avant device is capable to wake up the I/O Banks on the left and right side of the device to user specified value at the beginning of the bitstream.
- The Avant FPGA Fabric wakes up and enter user mode once it reads in the entire bitstream. If it is necessary to delay the wake-up, the simplest method is by using the DONE pin. Wake-up can be delayed by holding the open-drain DONE pin low until the wake-up is desired.

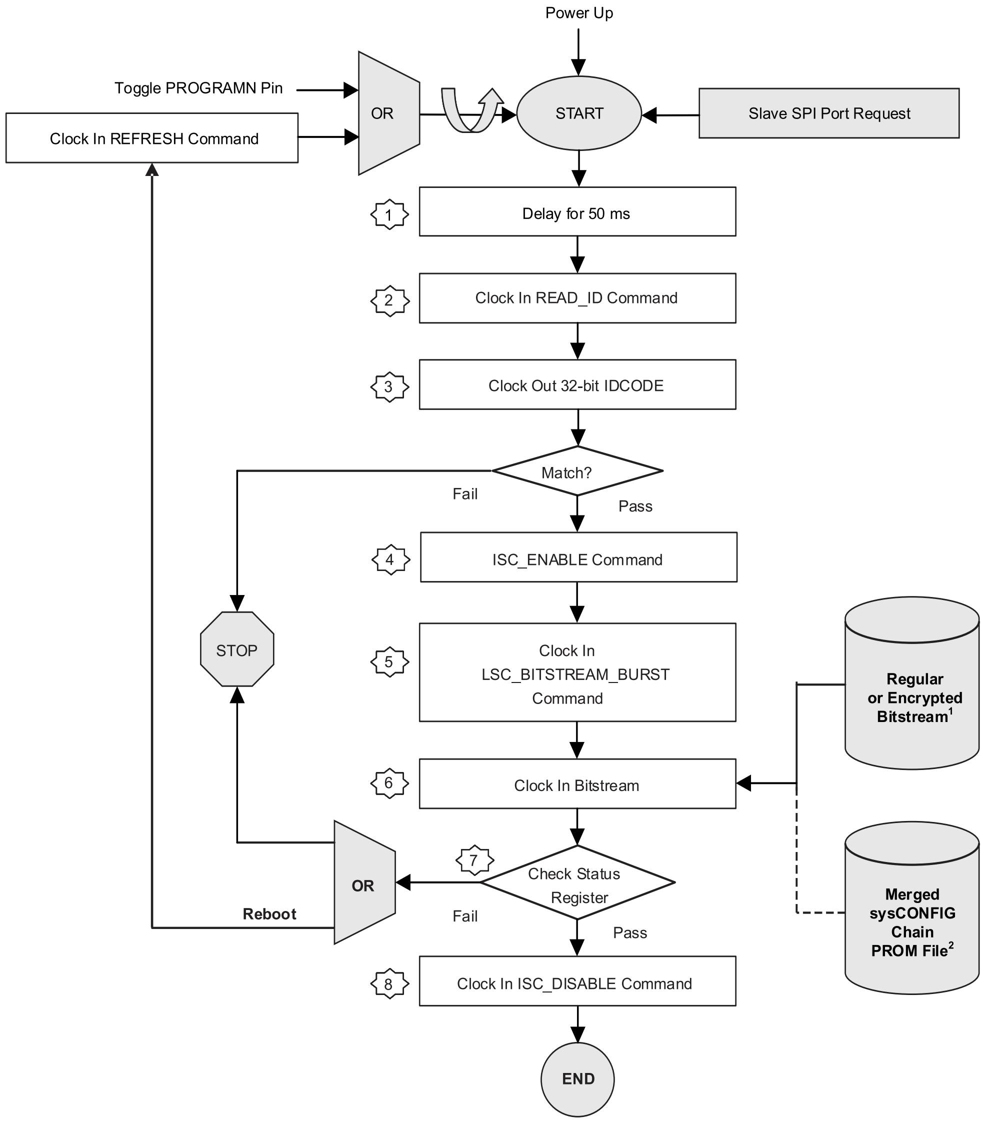


Figure 6.8. Target SPI Configuration Flow

Notes:

1. For a single Avant 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 PROM file.

6.4.6. Command Waveforms

6.4.6.1. Class A Command Waveforms

The Class A commands are ones that read data out from the Lattice Avant device. Bit 0 of the data or bitstream is read out first. After the 32 bits command (Opcode + Operand), SSPI sends 1 on MISO until return data is ready with one byte 0 as data valid flag.

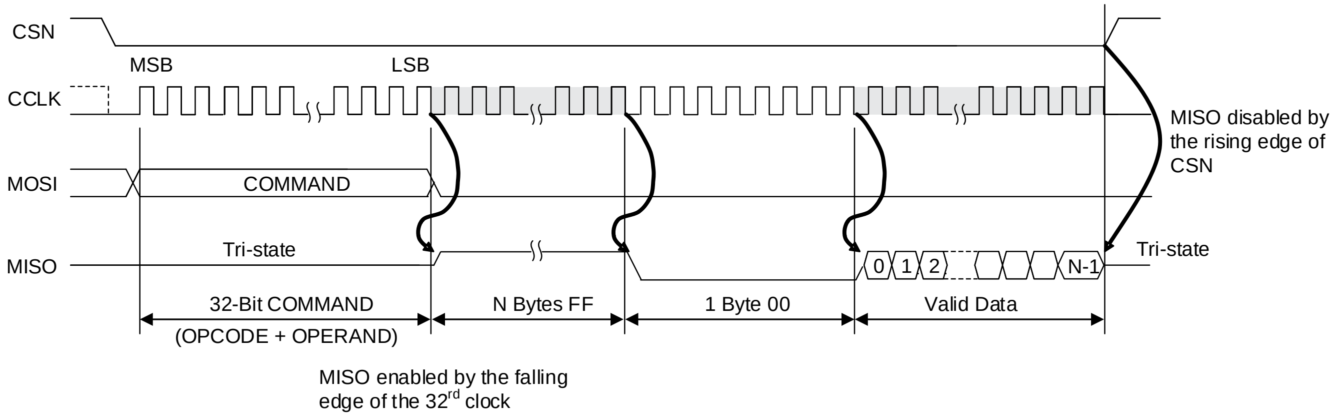


Figure 6.9. Class A Command Waveforms

6.4.6.2. Class B Command Waveforms

The Class B commands are used to shift data into the port. Bit 0 of the data or bitstream is shifted in first. The twenty-four (24) dummy clocks provide the device the necessary delay time to execute the command properly.

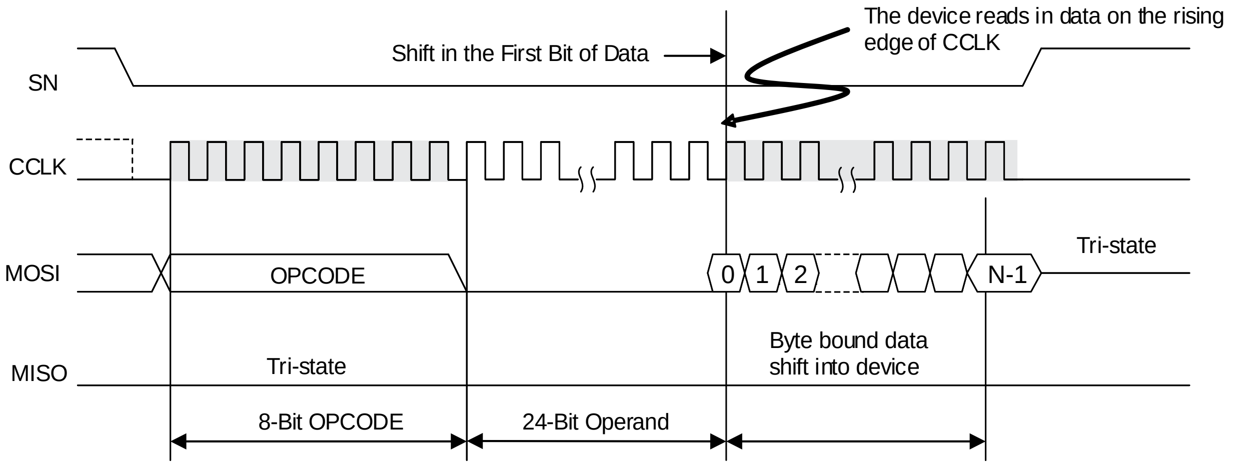

Shift in the First Bit of Data
The device reads in data on the rising edge of CCLK
SN
CCLK
MOSI
MISO
OPCODE
0
1
2
N-1
Tri-state
Tri-state
Byte bound data shift into device
8-Bit OPCODE
24-Bit Operand

Figure 6.10. Class B Command Waveforms

6.4.6.3. Class C Command Waveforms

The Class C commands do not require any data to be shifted in or out. The twenty-four (24) dummy clocks provide the device the necessary delay for the proper execution of the command. Even if extra dummy clocks are presented, the device ignores them.

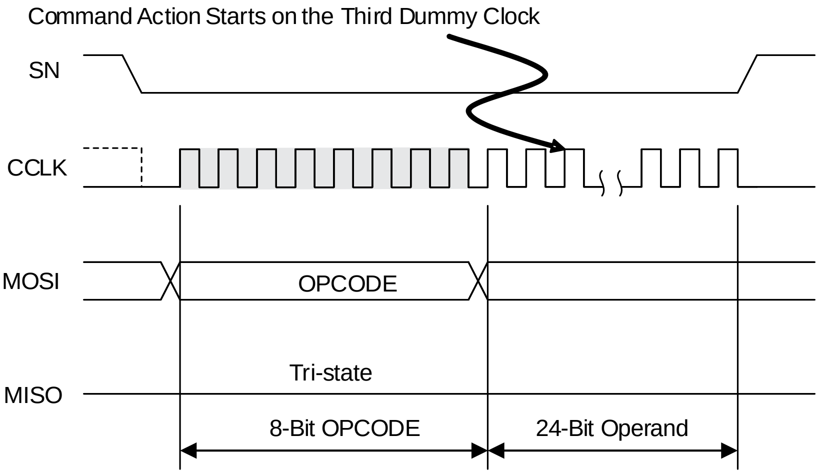


Figure 6.11. Class C Command Waveforms

6.4.6.4. Class D Commands Waveforms

The Class D commands do not need to shift data in or out but still require a delay to execute the action associated with the command. This type of command cannot terminate the action of any commands including itself. After the 24th dummy clock, continuing to clock or suspending the clock or driving the SN pin high does not terminate the action. The action ends when it is complete. This class of commands is defined particularly for the benefit of the two unique commands: CLEAR and REFRESH.

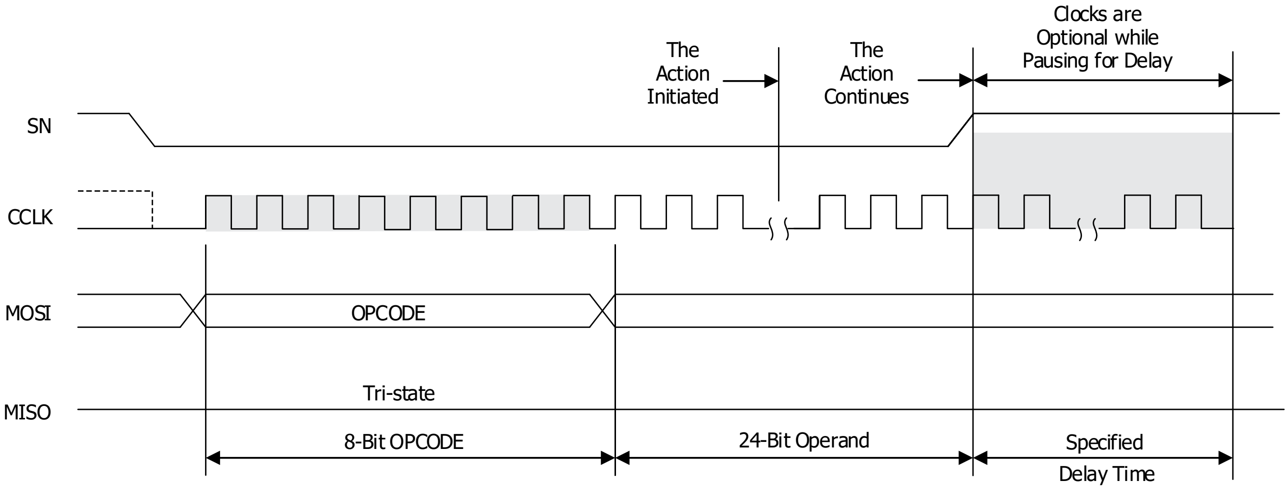


Figure 6.12. Class D Command Waveforms

6.4.7. Target SPI to Controller SPI Bridge

As the Controller SPI port and Target SPI port on the Lattice Avant device are allocated at different set of pins, the Lattice Avant device provides the Target SPI to Controller SPI Bridge, which enables programming the external SPI Flash through the Avant Target SPI configuration port.

MSPI_BRIDGE command is used to enable this bridging. Once this bridging is enabled, any data following this command on SSPI interface is directed to MSPI interface. Once this access is terminated by the external Host, by deasserting the Chip-Select (CSN) signal of SSPI, the bridging function is disabled and normal target access for Configuration continues. The functional diagram of the SSPI to MSPI Bridge is shown in Figure 6.13.

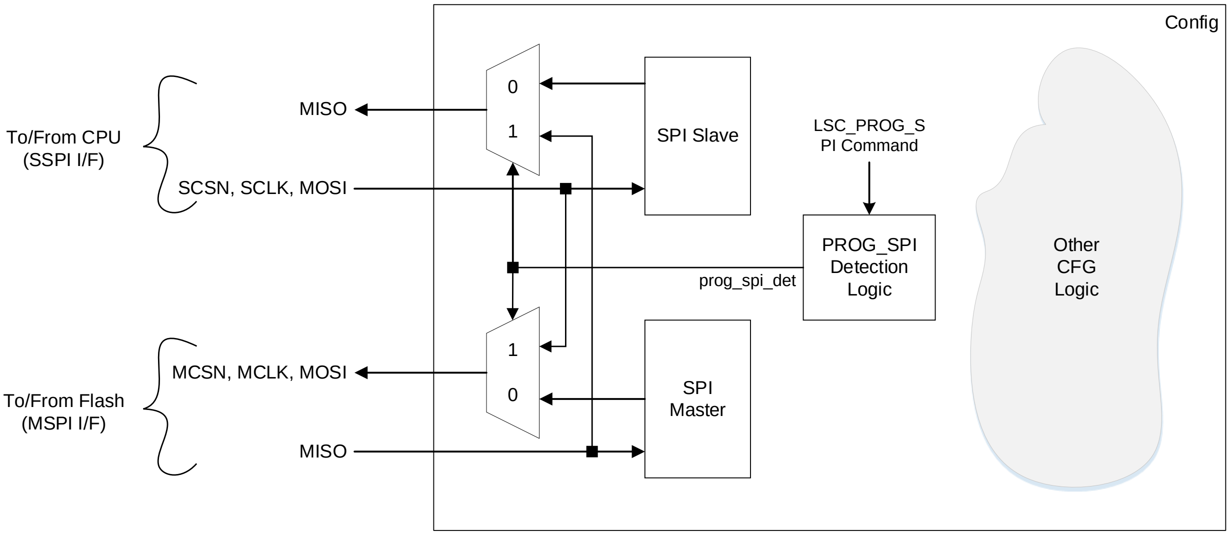


Figure 6.13. SSPI to MSPI Bridge Functional Diagram

The default support for this bridging is serial data mode only. Based on the devices I/O support, dual and quad can be supported, while serial SSPI to MSPI bridging is supported by default. As an example, the diagram for SSPI to MSPI Bridge utilization is shown in Figure 6.14.

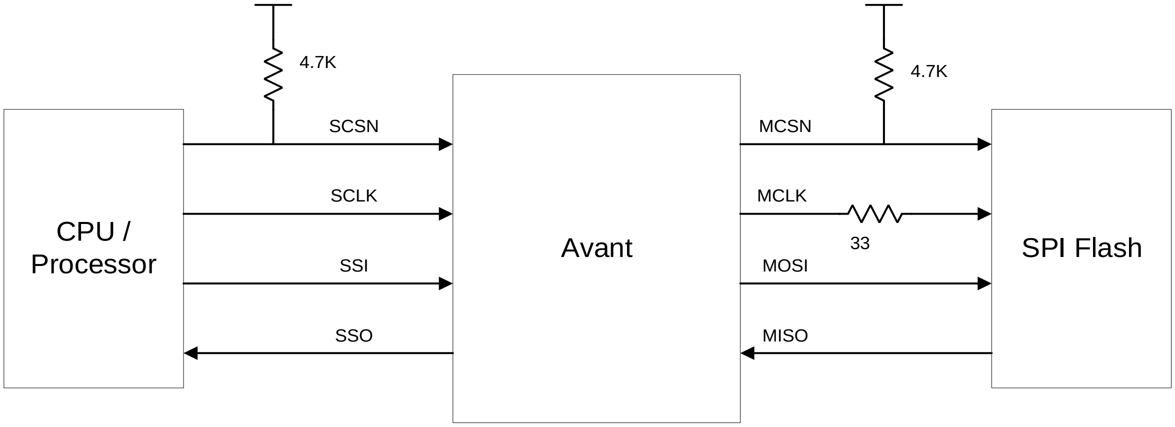


Figure 6.14. SSPI to MSPI Bridge Block Diagram

6.5. JTAG Mode

The Avant device provides a four-pin JTAG engine, which is fully compliance with IEEE 1149 (Standard Test Access Port and Boundary-Scan Architecture). The JTAG port on Avant devices provides:

- Offline external Flash memory programming
- Background external Flash memory programming
- Direct SRAM configuration
- Full access to the Avant Device Configuration Logic
- Device chaining
- IEEE 1149.1 testability
- IEEE 1532 Compliant programming

The advantages of keeping the JTAG port available include:

- Multi-chain Architectures – The JTAG port is the only configuration and programming port that permits the Lattice 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 which instruments the user design inside the Lattice Avant device, similar to how an external logic analyzer permits analysis of board level logic. Reveal access is only available via the JTAG port.
- SRAM Readback – The JTAG port can access the configuration SRAM. It is occasionally necessary to perform failure analysis for SRAM based FPGAs. A key component for failure analysis is reading the configuration SRAM.
- Boundary Scan Testability – Board level connectivity testing performed using IEEE 1149.1 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.

6.5.1. Method to Enable the JTAG Port for Configuration

The JTAG port is enabled by drive CFGMODE pin low.

The four pins for the JTAG port on Avant devices are dedicated I/O. It can enable the JTAG port at any time, no matter whether the device is in configuration mode or user functional mode. To gain the configuration access to the device, the PORT_REQUEST command should be used. See the PORT REQUEST Command section for PORT_REQUEST command detail.

6.5.2. JTAG Port AC Timing Requirements

The JTAG port AC timing requirements are listed in Table 6.12.

Table 6.12. JTAG AC Timing Requirements

Description	Parameter	Min	Max	Unit
TCK Frequency	f_{MAX}	—	25	MHz
TCK Frequency – JTAG to Controller SPI Bridge (all devices, except-E70B)	f_{MAX}	—	25	MHz
TCK Frequency – JTAG to Controller SPI Bridge (-E70B)	f_{MAX}	—	1.5	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).

6.5.3. JTAG to Controller SPI Bridge

The Lattice Avant devices provide the JTAG to Controller SPI Bridge, which enables programming the external SPI Flash through the Avant JTAG port.

MSPI_BRIDGE instruction is used to enable this bridging. There is a 16-bit TDR between TDI and TDO for this instruction. Upon the TAP controller going through the Update-DR state first time, the internal TDI signal, the gated TCK signal, the Shift-DR signal, and the internal TDO signal are unconditionally connected respectively to the MOSI/MD0 pin, the MCLK pin, the MCSN pin, and the MISO/MD1 pin.

As an example, the diagram for JTAG to MSPI Bridge utilization is shown in Figure 6.15.

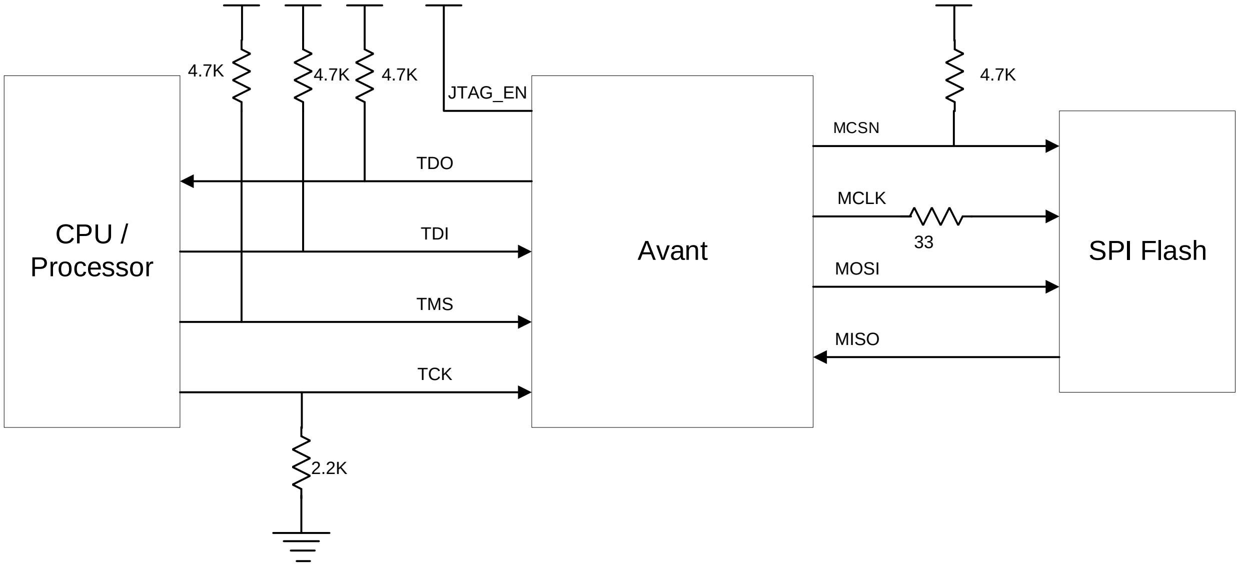


Figure 6.15. JTAG to MSPI Bridge Block Diagram

6.5.4. JTAG ispTRACY™/Reveal Support

The JTAG ispTRACY feature supported in FPGAs is the optional addition of internal logic analyzers (ILA). These ILAs have features similar to external logic analyzers such as programmable event and trigger conditions and deep trace memory. Logic analyzer IP (two current versions are currently supported: ispTRACY and REVEAL) consists of one ispJTAG core plus numerous ispLA logic analyzer cores (device limited). The IP block ispJTAG implements two embedded registers to support logic analyzers. To implement the ispJTAG core, the configuration block has an 11-port I/O interface to the core logic as seen in Figure 6.16.

In order to be IEEE 1149.1 compliant, there is a default 16-bit register implemented for registers ER1 and ER2 in order to allow the size of this register to be a fixed length whenever these registers are not implemented in FPGA logic. Whenever these registers are implemented in FPGA logic, these 16-bit registers are replaced with a new fixed length register that is based on the customer application – in order to allow for IEEE 1149.1 compliance, the BSDL file needs to be modified by the customer to define the lengths of these new ER1 and/or ER2 registers. This new implementation is different from that of all previous device architectures.

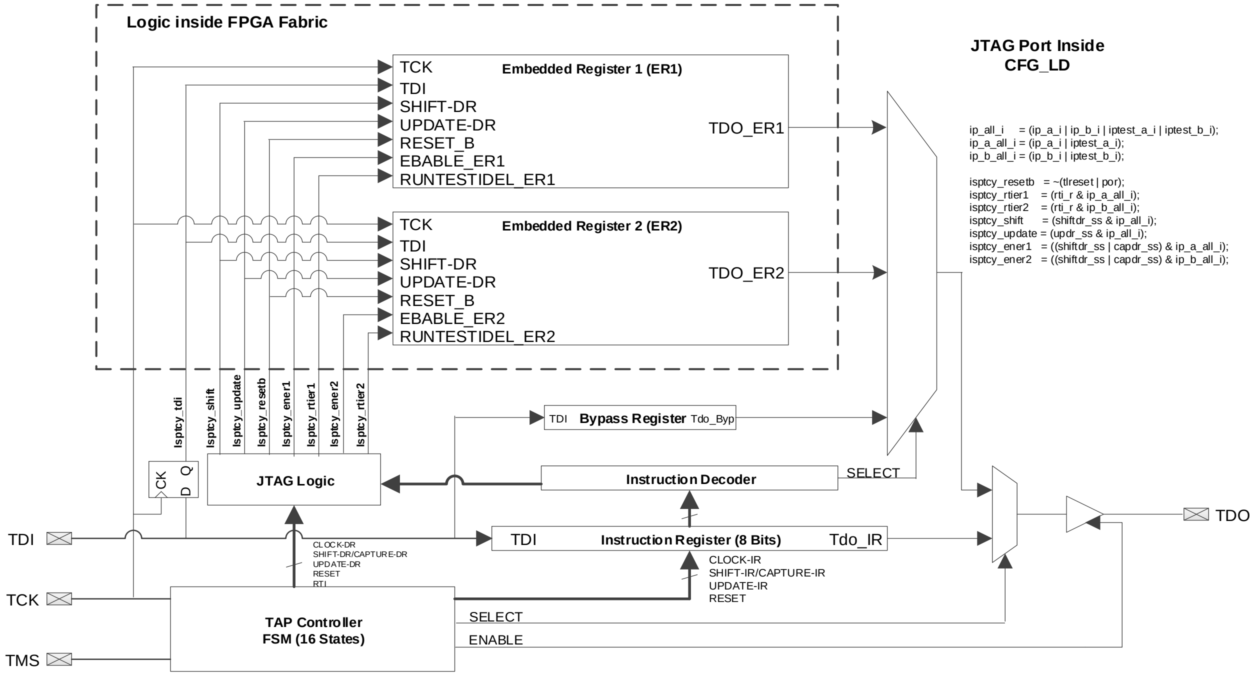

Logic inside FPGA Fabric
TCK
TDI
SHIFT-DR
UPDATE-DR
RESET_B
EBABLE_ER1
RUNTESTIDEL_ER1
Embedded Register 1 (ER1)
TDO_ER1
TCK
TDI
SHIFT-DR
UPDATE-DR
RESET_B
EBABLE_ER2
RUNTESTIDEL_ER2
Embedded Register 2 (ER2)
TDO_ER2
JTAG Port Inside
CFG_LD
ip_all_i = (ip_a_i | ip_b_i | iptest_a_i | iptest_b_i);
ip_a_all_i = (ip_a_i | iptest_a_i);
ip_b_all_i = (ip_b_i | iptest_b_i);
isptcy_resetb = ~(tlreset | por);
isptcy_rtier1 = (rti_r & ip_a_all_i);
isptcy_rtier2 = (rti_r & ip_b_all_i);
isptcy_shift = (shiftdr_ss & ip_all_i);
isptcy_update = (updr_ss & ip_all_i);
isptcy_ener1 = ((shiftdr_ss | capdr_ss) & ip_a_all_i);
isptcy_ener2 = ((shiftdr_ss | capdr_ss) & ip_b_all_i);
Isptcy_tdi
Isptcy_shift
Isptcy_update
Isptcy_resetb
Isptcy_ener1
Isptcy_rtier1
Isptcy_ener2
Isptcy_rtier2
CK
Q
D
JTAG Logic
TDI
Bypass Register
Tdo_Byp
Instruction Decoder
SELECT
TDI
Instruction Register (8 Bits)
Tdo_IR
CLOCK-DR
SHIFT-DR/CAPTURE-DR
UPDATE-DR
RESET
RTI
CLOCK-IR
SHIFT-IR/CAPTURE-IR
UPDATE-IR
RESET
TAP Controller
FSM (16 States)
SELECT
ENABLE
TDI
TCK
TMS
TDO

Figure 6.16. JTAG ispTRACY Interface

In user functional mode, the four JTAG pins located in I/O BANK1 can have their functionality changed from user function to JTAG function by bringing JTAG_EN pin to VCCIO1. Hence, the TMS/TDI/TCK pins need to be configured as INPUT or BI-DIRECTIONAL in user mode.

The JTAG driver to Avant needs to match the JTAG pins I/O type. For example, if four JTAG pins in Bank 1 use LVCMOS33 then the JTAG driver needs to be 3.3 V.

When JTAG_EN is high in user mode, the four signals need to be HIGHZ from other devices beside the JTAG driver.

After the JTAG_EN goes low in user mode, the drive strength of the four pins remains at 8 mA until 200 ms later and need dynamic switching to return to user mode drive strength. If these four pins stay static after JTAG_EN goes low, they remain at 8 mA drive strength until switching.

6.6. Device Status Register

The Lattice Avant User Status Register (USR) is used to provide the information of the device during or after the configuration process to user through the READ_STATUS [0/1] commands. The default value of the Status Register is '0' and it is reset by powering up, PROGRAMN pin toggle low, or the REFRESH command execution. There are four user status registers that can be read individually with the commands READ_STATUS[0/1].

Note: If authentication is enabled the BSE Error status will be disabled.

Table 6.13. STATUS0 Register

Bit	Description	Mask	Definition	Reset Condition
31	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 ID Error	POR Last command
30				
29				
28	BSE Secondary	BS_AUTH	0000 No error has occurred.	POR Refresh

Bit	Description	Mask	Definition	Reset Condition
27 26 25	Boot Error Code		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 Mode 0101 Bitstream Engine execution was aborted by the user 0110 Bitstream Engine last SRAM address programmed, but did not terminate 0111 There was a CRC checksum error 1000 Authentication Error 1001 Authentication Setup Error	
24 23 22 21	BSE Primary Boot Error Code	BS_AUTH	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 Mode 0101 Bitstream Engine execution was aborted by the user 0110 Bitstream Engine last SRAM address programmed, but did not terminate 0111 There was a CRC checksum error 1000 Authentication Error 1001 Authentication Setup Error	POR Refresh
20	reserved	reserved	reserved	—
19	SPI Controller Primary Boot Failed	—	Status bit shows that the primary booting has failed, even if the secondary booting is successful when dual boot is enabled. Once set, this bit can only be cleared by REFRESH command execution, or PROGRAMN pin toggle low.	Refresh
18 17	Preamble	—	Indicates what preamble was receive through the last bitstream segment. 00-Plain, 01-Auth Header, 10-Authenticated, 11-Encrypted + Authenticated	POR Refresh
16	reserved	reserved	reserved	—
15	UID_EN	—	Status bit indicates that the UID EN bit inside the OTP is programmed. The IDCODE_PUB command will read out the User ID, which is the 32 bits 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.	Mirror OTP shadow
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.	Mirror OTP shadow (Sticky)
13	reserved	reserved	reserved	—
12	Fail Flag	—	Indicates last command failed. Valid in command mode.	Last Command
11	Busy Flag	—	Config engine is busy	Mirror state
10	INITN Pin State	—	INITN pin state. 0=low, 1=high	Mirror state
9	Auth DONE	—	Authentication is done, 1=authentication done	POR Refresh,
8	DONE Pin State	—	Done pin state. 0=low, 1=high	POR, Refresh, Mirror config mode
7	DONE	—	Device done bit is set from PROG_DONE command. When device wakeup is complete the done pin will go high.	POR, Refresh
6	Read Enable	—	Configuration memory read is enabled, qualified with security state. See Note1	—
5	Write Enable	—	Configuration memory write is enabled, qualified with security state. See Note2	PROG Disable

Bit	Description	Mask	Definition	Reset Condition
4	Erase Enable	—	Configuration memory erase is enabled, qualified with security state. See Note3	PROG Disable
3	reserved	reserved	reserved	—
2	reserved	reserved	reserved	—
1	bitstream enable	—	Device is in bitstream mode.	—
0	Transparent Mode Enabled	—	Device is in transparent programming mode.	Mirror state

Table 6.14. STATUS1 Register

Bit	Description	Mask	Definition	Reset Condition
31	USER OTP PROG Lock	—	PROG_OTP and LOCK OTP command locked from external ports USER_OTP_HLOCK | USER_OTP_PROG_SLOCK	Mirror OTP Shadow
30	USER OTP READ Lock	—	READ_OTP command locked form external ports USER_OTP_HLOCK | USER_OTP_READ_SLOCK	Mirror OTP Shadow
29	AES Write Lock	—	OTP AES Key is locked for Write AES_KEY_HLOCK | AES_KEY_PROG_SLOCK	Mirror OTP Shadow
28	PUB Key Write Lock	—	OTP PUB Key is locked for Write PUB_KEY_HLOCK | PUB_KEY_PROG_SLOCK	Mirror OTP Shadow
27	JTAG Test Only	—	JTAG Port is in test only mode. Only commands that can be executed are Boundary scan and ISP tracy commands.	Mirror OTP Shadow
26	SSPI -> MSPI Bridge Lock	—	Target SPI to Controller SPI bridge is locked.	Mirror OTP Shadow
25	JTAG -> MSPI Bridge Lock	—	JTAG to Controller SPI bridge is locked.	Mirror OTP Shadow
24	LMMI -> MSPI Bridge Lock	—	LMMI to Controller SPI bridge is locked.	Mirror OTP Shadow
23	BS Auth Mode	—	0x0 - None 0x1 - AES256-GMAC 0x2-3 - reserved2 0x4 - ECDSA256 (digest = SHA2-256 or AES256-GCM1) 0x5 - ECDSA384 (digest = SHA2-384 or AES256-GCM1) 0x6 - ECDSA521 (digest = SHA2-512 or AES256-GCM1) 0x7-9 - reserved2 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-0xF - reserved2	Mirror OTP Shadow
22				
21				
20				
19	Device State (SNOP, SOP, SD, SNOPR)	—	00-SNOP, 01-SOP, 10-SD, 11-SNOPR	Mirror state
18				
17	ATM Global Alarm	—	There was a system ATM alarm trigger event.	Error cleared
16	PUF_Enrolled	—	0- PUF Not enrolled, 1-PUF enrolled from OTP	—
15	reserved	—	reserved	—
14	reserved	—	reserved	—
13	reserved	reserved	reserved	—
12	reserved	reserved	reserved	—
11	User WDT Reboot	—	Device was rebooted because the user watchdog timed out.	POR, normal refresh
10	User WDT Busy	—	User watch dog timer is running	Mirror WDT enable
9	SFDP Timeout	—	Boot timed out reading the flash signature. SFDP/LSCC	POR Refresh

Bit	Description	Mask	Definition	Reset Condition
8	Target SPI Timeout	—	The target SPI timed out before receiving the done command, while sending a bitstream.	POR, Refresh
7	Daisy Chain Flow Through Mode	—	Device has completed configuration and is in daisy-chain flow-through state.	Mirror state
6	Daisy Chain Bypass Mode	—	Device has completed configuration and is in daisy-chain bypass state.	Mirror state
5	Dry Run BSE Error Code	BS_AUTH	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 Mode 0101 Bitstream Engine execution was aborted by the user 0110 Bitstream Engine last SRAM address programmed, but did not terminate 0111 There was a CRC checksum error 1000 Authentication Error 1001 Authentication Setup Error 1010 Bitstream Engine Timeout Error	POR Refresh
4				
3				
2				
1	Dry Run Auth Done	—	The dry-run bitstream has completed authentication.	POR Refresh,
0	Dry Run Done	—	The dry-run bitstream done command was received and no errors.	POR Refresh,

6.7. Control Register 0 (CR0)

The 32 bits Control Register 0 is used to control configuration logic behavior during and after configuration. User could Write the CR0 through PROG_CNTRL0 command and read back the CR0 content through READ_CNTRL0 command. Also, the CR0 could be written through the PROG_CNTRL0 command in bitstream file, and it becomes bitstream only once authentication is enabled. The bit definition of the CR0 is shown in Table 6.15.

Note: When BS_AUTH is enabled TRAN_CRAM and TRAN_INIT is disabled.

Table 6.15. Bit Definition for Control Register 0

BIT	Name	Default	Description
31:26	reserved	0	reserved
25	TRAN_CRAM	0	Enable Transparent CRAM Programming
24	TRAN_INIT	0	Enable Transparent INIT Bus Programming
23	TRAN_SEC_ENGINE	0	Enable Transparent Security Engine Access
22	PRESERVE_REG	0	Preserve register values on refresh.
21	NO BOOT	0	Do not boot on refresh
20	reserved	0	reserved
19	MULTIBOOT_EN	0	Enable the Multi-Boot feature
[18:17]	DAISYCHAIN CTRL	0	Enable daisy-chain mode for upstream device
[16:15]	INTN OPT	0	Over-ride INITN
[14:13]	DONE OPT	0	Over-ride DONE
[12:10]	reserved	0	reserved
9	WAIT_DONE	0	Wait for done pin before wakeup for daisy chain upstream
8	Fast slew rate enable	0	Forces fast slew rate
7:0	reserved	0	reserved

Bit [7:0]: Reserved

Bit [8]: Force fast slew rate when clock frequency is below 25 MHz.

Bit [9]: WAIT_DONE

This bit is used for daisy chain configuration mode and is set in upstream devices to hold the completion of device wakeup. In daisy-chain mode the wakeup should be held off until the last device in the chain has successfully finished configuration by releasing its done pin.

Bit [12:10]: reserved Bit

[14:13]: DONE OPT

If bit 14 is set to "1", the DONE pin is overridden by the bit 13.

Bit [16:15]: INITN OPT

If bit 16 is set to "1", the INITN pin is overridden by bit 15.

Bit [18:17]: DAISY_CHAIN_CTRL

Selects daisy chain mode for upstream device as shown in Table 6.16.

Table 6.16. Daisy Chain Mode Selection

Bit [18:17]	Behavior
10	Daisy Chain Bypass Mode
11	Daisy Chain Flow-Through Mode
0X	No Daisy Chain (Default)

Bit [19]: MULTIBOOT_EN

Control bit for Controller SPI boot address selection. If set to "1", use boot address from LMMI register mspim_addr[47:0] or usermode register mc_mspi_addr[47:0], depends on mc_source_sel.

Bit [20]: reserved

Bit [21]: NO BOOT

This bit disables the automatic booting activity for the next programn pin toggle event or lsc_refresh command execution.

Bit [22]: PRESERVE_REG

This bit when set preserves CR1, and feature row settings. When set, these registers do not revert to the default values.

Bit [23]: TRAN_SEC_ENGINE

Bit to request access to security engine while in usermode.

Bit [24]: TRAN_INIT

Control bit to enable access to the INIT registers in transparent mode.

Bit [25]: TRAN_CRAM

Control bit to enable the write operation to the SRAM array in transparent mode.

Bit [31:26]: reserved

6.8. Control Register 1 (CR1)

The 32 bits Control Register 1 is used to control EFUSE/OTP access control logic behavior during and after configuration. You can write the CR1 through PROG_CNTRL1 command and read back the CR1 content through READ_CNTRL1 command. Also, the CR1 could be written through the PROG_CNTRL1 command with the Bitstream file. The bit definition of the CR1 is shown in Table 6.17.

Table 6.17. Bit Definition for Control Register 1

Bit	Definition	Default	Description
[31:30]	Configuration Core clock select	0	Sets the config clock divider (default divide by 1)
[29:27]	Controller Signature Timer count value	0	Signature counter timeout value
26	MSPI_CPHA	0	Invert Controller SPI CPHA
25	MSPI_CPOL	0	Invert Controller SPI CPOL

Bit	Definition	Default	Description
24	MSPI_TX Edge	0	Invert Controller SPI Transmit data edge
23	MSPI_RX Edge	0	Invert Controller SPI Receive data edge
22	MSPI_LSBF	0	Change Controller SPI Bit shift order to LSB first
21	SSPI_CPHA	0	Invert Target SPI CPHA
20	SSPI_CPOL	0	Invert Target SPI CPOL
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
13	EBR Erase Disable	0	Disable the erase of EBR contents on PROGRAMN or refresh.
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 SPIM address	0	Enable 32-bit Controller SPI address
8	32-bit SPIM commands	0	Enable 32-bit Controller SPI commands
7	Disable I/O glitch filter	0	Disable the I/O glitch filter
[6:3]	Target Idle Timer count value	0	—
[2:0]	Controller Preamble Timer count value	0	—

Bit [2:0]: Controller Preamble Timer Count Value

Time to get a valid preamble:

0	200 ms
1	100 ms
2	50 ms
3	40 ms
4	20 ms
5	1 ms
6	500 µs
7	100 µs

Bit [6:3]: Target Idle Timer Count Value

0	disabled (default) – bitstream would be 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
11	100 ms
12	75 ms
13	50 ms
14	25 ms
15	10 ms

Bit [7]: Disable I/O Glitch Filter

Disable I/O Glitch Filter during config.

Note: Controls glitch filter in WRIO over INIT bus.

Bit [8]: 32-bit MSPI commands

Send 32-bit command set on MSPI, default 24-bit command set.

Bit [9]: 32-bit MSPI addressing

Enable 32-bit MSPI addressing mode, default 24-bit addressing.

Bit [10]: Signature infinite retry

Retry the SFDP/LSCC signature an infinite number of times.

Bit [11]: Signature disable

If sfdp_en=0 do not read LSCC signature from flash, go directly to reading preamble.

1=skip checking signature when sfdp_en=0, 0=signature enabled (default 0).

Bit [12]: SFDP Enable

Enables checking the SFDP signature during controller SPI booting. Should be enabled for flash devices that support SFDP.

1-enable reading SFDP signature from flash SFDP header.

0-enable reading LSCC signature from flash starting at address 0x000

Bit [13]: erase_disable

Control bit to disable erase on refresh.

Bit [16:14]: SSPI Auto

This control is to enable the 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 DDRx8 mode

Bit [17]: SSPI_LSBF

Control bit for SSPI to receive commands and address least significant bit first.

Bit [18]: SSPI_RX_EDGE

Control bit to adjust RX clock edge.

Bit [19]: SSPI_TX_EDGE

Control bit to adjust TX clock edge.

Bit [20]: SSPI_CPOL

This control bit selects an inverted or non-inverted clock

0 = Active high clocks selected. In idle state clock is low

1 = Active low clocks selected. In idle state clock is high

Bit [21]: SSPI_CPHA

This Control bit is used to select the clock format.

0 = sampling of data occurs at the rising edge of the clock

1 = sampling of data occurs at the falling edge of the clock

Bit [22]: MSPI_LSBF

Control bit for MSPI to shift least significant bit first.

Bit [23]: MSPI_RX_EDGE

Control bit to adjust RX clock edge.

Bit [24]: MSPI_TX_EDGE

Control bit to adjust TX clock edge.

Bit [25]: MSPI_CPOL

This control bit selects an inverted or non-inverted clock
0 = Active high clocks selected. In idle state clock is low
1 = Active low clocks selected. In idle state clock is high

Bit [26]: MSPI_CPHA

This Control bit is used to select the clock format.
0 = sampling of data occurs at the rising edge of the clock
1 = sampling of data occurs at the falling edge of the clock

Bit [29:27]: Controller Signature Timer Count Value

Signature_count: time to get a valid LSCC/SFDP signature.

0	200 ms
1	100 ms
2	50 ms
3	40 ms
4	20 ms
5	1 ms
6	500 µs
7	100 µs

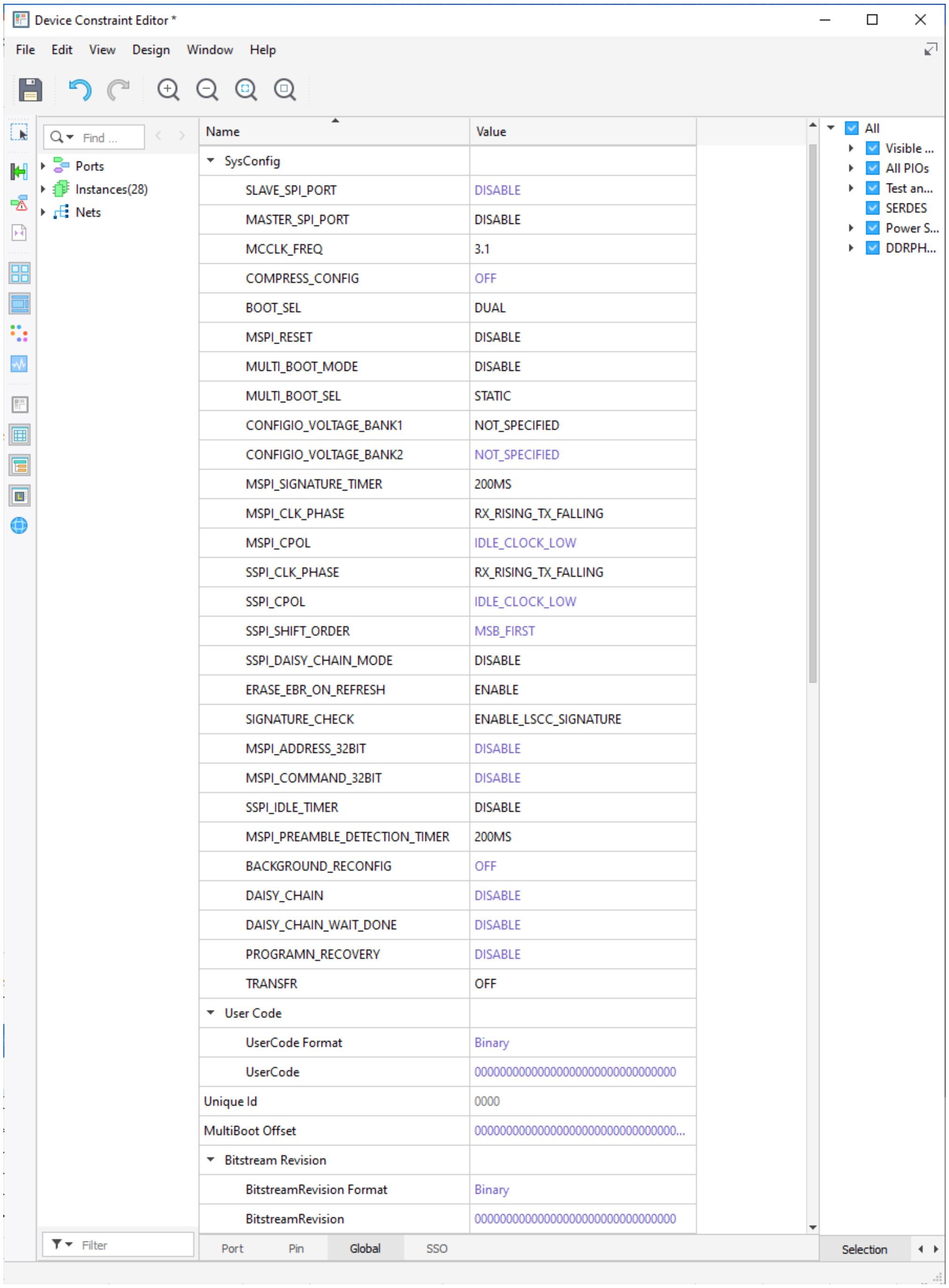
Bit [31:30]: Configuration Core CLK (smclk) SEL

Control bits to set the Configuration Engine clock frequency with respect to the internal oscillator frequency.
0 – divide by 1
1 – divide by 2
2 – divide by 3
3 – divide by 4

7. Software Selectable Options

The operation of the Lattice Avant device configuration logic is managed by options selected in the Lattice Radiant design software. The Lattice Avant devices use the built-in arbitration logic, as described in the Configuration Ports Arbitration section, to select the configuration port. User can set the configuration port persistence after device enters user function mode, and the system Configuration Pins (PROGRAMN pin, INITN pin, and DONE pin) persistence using the Lattice Radiant Device Constraint Editor.

The configuration logic preferences are accessed using Device Constraint Editor. Click the Global tab and look for the sysCONFIG tree. The sysCONFIG section is shown in Figure 7.1.



Name	Value
SysConfig	
SLAVE_SPI_PORT	DISABLE
MASTER_SPI_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_CLK_PHASE	RX_RISING_TX_FALLING
MSPI_CPOL	IDLE_CLOCK_LOW
SSPI_CLK_PHASE	RX_RISING_TX_FALLING
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
PROGRAMN_RECOVERY	DISABLE
TRANSFR	OFF
User Code	
UserCode Format	Binary
UserCode	00000000000000000000000000000000
Unique Id	0000
MultiBoot Offset	00000000000000000000000000000000...
Bitstream Revision	
BitstreamRevision Format	Binary
BitstreamRevision	00000000000000000000000000000000

Figure 7.1. sysCONFIG Preferences in Global Tab, Lattice Radiant Device Constraint Editor

Table 7.1. sysCONFIG Options[1]

Option Name	Default Setting	All Settings
SLAVE_SPI_PORT	DISABLE	DISABLE, SERIAL, DUAL, QUAD, xSPI[2]
MASTER_SPI_PORT	DISABLE	DISABLE, SERIAL, DUAL, QUAD, xSPI, xSPI_DIFF_CLK[2]
MCCLK_FREQ	3.1	3.1, 7.1, 14.3, 28.6, 57.1, 66.7, 80.0, 100.0, 133.3
COMPRESS_CONFIG	OFF	OFF, ON[2]
BOOT_SEL	DUAL	DUAL, SINGLE
MSPI_RESET	DISABLE	DISABLE, ENABLE[2]
MULTI_BOOT_MODE	DISABLE	DISABLE, ENABLE[2]
MULTI_BOOT_SEL	STATIC	STATIC, DYNAMIC[2]
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_CLK_PHASE	RX_RISING_TX_FALLING	RX_RISING_TX_RISING[2], RX_RISING_TX_FALLING, RX_FALLING_TX_RISING[2], RX_FALLING_TX_FALLING[2]
MSPI_CPOL	IDLE_CLOCK_LOW	IDLE_CLOCK_LOW, IDLE_CLOCK_HIGH[2]
SSPI_CLK_PHASE	RX_RISING_TX_FALLING	RX_RISING_TX_RISING[2], RX_RISING_TX_FALLING, RX_FALLING_TX_RISING[2], RX_FALLING_TX_FALLING[2]
SSPI_CPOL	IDLE_CLOCK_LOW	IDLE_CLOCK_LOW, IDLE_CLOCK_HIGH[2]
SSPI_SHIFT_ORDER	MSB_FIRST	MSB_FIRST, LSB_FIRST[2]
SSPI_DAISY_CHAIN_MODE	DISABLE	DISABLE, SCM[2], AUTO[2]
ERASE_EBR_ON_REFRESH	DISABLE	DISABLE, ENABLE[2]
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[2], ON[2]
DAISY_CHAIN	DISABLE	DISABLE, BYPASS[2], FLOW_THROUGH[2]
DAISY_CHAIN_WAIT_DONE	DISABLE	DISABLE, ENABLE[2]
PROGRAMN_RECOVERY	DISABLE	DISABLE, ENABLE[2]
TRANSFR	OFF	OFF, ON
UserCode_Format	BINARY	BINARY, HEX, ASCII, AUTO
UserCode	32’b0	32-bit User Code (User Electronic Signature)
Unique_ID	16’b0	Upper 16-bit user-defined code for TraceID.
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

Notes:

1. The CONFIG_MODE option in Lattice Radiant software Global Preference is for the software to preserve appropriated pins to help customer design flow. It does not serve the purpose of enabling SPI port nor setting appropriate bits in the PROM file.
2. The software limited setting for initial silicon.

7.1. SLAVE_SPI_PORT

The SLAVE_SPI_PORT preference controls the behavior of the Target SPI configuration port after the device enters user mode. There are five states to which the SLAVE_SPI_PORT preference can be set:

- DISABLE – This setting disconnects the SPI port pins from the configuration logic.
- SERIAL – This setting preserves the standard serial SPI port I/O (SCLK, SCSN, SMOSI, SMISO) when the device is in user mode. When the pins are preserved, an external SPI controller can interact with the configuration logic. The preference also prevents user from over-assigning I/O to those pins.
- DUAL – This setting preserves the SPI port I/O (SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1) in DUAL mode when the device is in user mode. When the pins are preserved, an external SPI controller can interact with the configuration logic. The preference also prevents user from over-assigning I/O to those pins.
- QUAD – This setting preserves the SPI port I/O (SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1, SSIO[2:3]) in QUAD mode when the device is in user mode. When the pins are preserved, an external SPI controller can interact with the configuration logic. The preference also prevents user from over-assigning I/O to those pins.
- xSPI – This setting preserves the SPI port I/O (SCLK, SCSN, SMOSI/SSIO0, SMISO/SSIO1, SSIO[2:7], SDS) in OCTAL mode when the device is in user mode. When the pins are preserved, an external SPI controller can interact with the configuration logic. The preference also prevents user from over-assigning I/O to those pins.

The SLAVE_SPI_PORT could be enabled at the same time as the MASTER_SPI_PORT, because the Target SPI port and Controller SPI port are in different I/O banks, this makes the Avant device possible to support the Target SPI to Controller SPI bridging functionality. The default setting for this preference is DISABLE.

7.2. MASTER_SPI_PORT

The MASTER_SPI_PORT preference controls the behavior of the Controller SPI configuration port after the device enters user mode. There are six states to which the MASTER_SPI_PORT preference can be set:

- DISABLE – This setting disconnects the Controller SPI port pins from the configuration logic. The Controller SPI pins (MCLK, MCLKN, MCSN, MMOSI/MSIO0, MMISO/MSIO1, MSIO[2:7]) could be general purpose user I/O.
- SERIAL – This setting preserves the SPI port I/O (MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1) in serial mode when the device is in user mode. The preference also prevents user from over-assigning I/O to those pins.
- DUAL – This setting preserves the SPI port I/O (MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1) in dual mode when the device is in user mode. The preference also prevents user from over-assigning I/O to those pins.
- QUAD – This setting preserves the SPI port I/O (MCLK, MCSN, MMOSI/MSIO0, MMISO/MSIO1, MSIO [2:3]) in quad mode when the device is in user mode. The preference also prevents user from over-assigning I/O to those pins.

The default setting for this preference is DISABLE.

7.3. MCLK_Frequency

The MCCLK_FREQ preference controls the MCLK frequency used to retrieve data from an external SPI Flash when using EXTERNAL or Dual Boot configuration modes. The Avant device uses a nominal 3.5 MHz (±15%) clock frequency to begin retrieving data from the external SPI Flash. The MCCLK_FREQ value is stored in the incoming configuration data. The Avant device reads a series of padding bits, a *start of data* word (0xBDB3) and a control register value. The control register contains the new MCCLK_FREQ value. The device switches to the new clock frequency shortly after receiving the MCCLK_FREQ value. The MCCLK_FREQ has a range of possible frequencies available from 3.5 MHz up to 180 MHz. Take care not to exceed the maximum clock rate of the SPI Flash, or of the printed circuit board.

7.4. COMPRESS_CONFIG

The COMPRESS_CONFIG preference alters the way files are generated with compressed FPGA data frames. The default setting of COMPRESS_CONFIG is OFF.

7.5. BOOT_SEL

The BOOT_SEL preference selects the device booting mode, either DUAL boot or SINGLE boot. The default setting of BOOT_SEL is DUAL.

7.6. MSPI_RESET

The MSPI_RESET preference enables the hardware reset signal to the flash device. The default setting of this preference is DISABLE.

7.7. MULTI_BOOT_MODE

The MULTI_BOOT_MODE preference enables the multiple boot functionality of the device. The default setting of this preference is DISABLE.

7.8. MULTI_BOOT_SEL

The MULTI_BOOT_SEL preference selection the booting address for the multiple boot event when MULTI_BOOT_MODE is set to ENABLE.

- STATIC – Using MULTI_BOOT_OFFSET parameter as booting address.
- DYNAMIC – Using the user setting through CONFIG_LMMI interface.

The default setting for this preference is STATIC.

7.9. CONFIGIO_VOLTAGE_BANK1

The CONFIGIO_VOLTAGE_BANK1 preference specifies the VCCIO level for IO BANK1 to guide bank1 IO placement for PAR. The default setting for this preference is NOT_SPECIFIED.

7.10. CONFIGIO_VOLTAGE_BANK2

The CONFIGIO_VOLTAGE_BANK2 preference specifies the VCCIO level for IO BANK2 to guide bank2 IO placement for PAR. The default setting for this preference is NOT_SPECIFIED.

7.11. MSPI_SIGNATURE_TIMER

Time to get a valid LSCC/SFDP signature. Once time out before signature detected, the configuration logic will declare signature detection failure.

7.12. MSPI_CLK_PHASE

Controller SPI clock phase selection to setup the data receiving and transmitting edge.

- RX_RISING_TX_RISING – Set the Controller SPI transceiver receiving data on rising edge of MCLK, transmitting data on rising edge of MCLK.
- RX_RISING_TX_FALLING – Set the Controller SPI transceiver receiving data on rising edge of MCLK, transmitting data on falling edge of MCLK.
- RX_FALLING_TX_RISING – Set the Controller SPI transceiver receiving data on falling edge of MCLK, transmitting data on rising edge of MCLK.
- RX_FALLING_TX_FALLING – Set the Controller SPI transceiver receiving data on falling edge of MCLK, transmitting data on falling edge of MCLK.

The default setting for this preference is RX_RISING_TX_RISING.

7.13. MCSPI_CPOL

Controller SPI clock polarity selection to choose inverted or non-inverted clock, which determines the clock idle state when the port is not active.

- INACTIVE_LOW – MCLK stay at LOW when idle.
- INACTIVE_HIGH – MCLK stay at HIGH when idle.

The default setting for this preference is INACTIVE_LOW.

7.14. SSPI_CLK_PHASE

Target SPI clock phase selection to setup the data receiving and transmitting edge.

- RX_RISING_TX_RISING – Set the Target SPI transceiver receiving data on rising edge of SCLK, transmitting data on rising edge of SCLK.
- RX_RISING_TX_FALLING – Set the Target SPI transceiver receiving data on rising edge of SCLK, transmitting data on falling edge of SCLK.
- RX_FALLING_TX_RISING – Set the Target SPI transceiver receiving data on falling edge of SCLK, transmitting data on rising edge of SCLK.
- RX_FALLING_TX_FALLING – Set the Target SPI transceiver receiving data on falling edge of SCLK, transmitting data on falling edge of SCLK.

The default setting for this preference is RX_RISING_TX_RISING.

7.15. SSPI_CPOL

Target SPI clock polarity selection to choose inverted or non-inverted clock, which determines the clock idle state when the port is not active.

- INACTIVE_LOW – SCLK stay at LOW when idle.
- INACTIVE_HIGH – SCLK stay at HIGH when idle.

The default setting for this preference is INACTIVE_LOW.

7.16. SSPI_SHIFT_ORDER

Target SPI transceiver shifting direction selection, Lest Significant Bit First (LSB_First) or Most Significant bit first (MSB_First).

- LSB_First – Least Significant Bit First shifting.
- MSB_First – Most Significant Bit First shifting.

The default setting for this preference is LSB_First.

7.17. SSPI_DAISY_CHAIN_MODE

Enable device as a downstream device in a daisy chain.

- DISABLE – 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 downstream device in a daisy chain following the SLAVE_SPI_PORT setting.

The default setting for this preference is DISABLE.

7.18. ERASE_EBR_ON_POR

Enable/Disable the erase of EBR contents on the refresh event triggered by PROGRAMN toggle or REFRESH command execution.

- DISABLE – Disable the erase of EBR contents on PROGRAMN or refresh.
- ENABLE – Enable the erase of EBR contents on PROGRAMN or refresh.

The default setting for this preference is DISABLE.

7.19. SIGNATURE_CHECK

This preference Enable/Disable the signature checking for the flash device before performing the bitstream booting. It also chose the signature checking mode.

- ENABLE_LSCC_SIGNATURE – Enable the signature checking for LSCC signature.
- DISABLE – Disable the signature checking.
- ENABLE_SFDP_SIGNATURE – Enable the signature checking for SFDP signature.

The default setting for this preference is ENABLE_LSCC_SIGNATURE.

7.20. MSPI_ADDRESS_32BIT

This option setup the address format for SPI Flash memory:

- DISABLE – Using 24-bit Controller SPI address.
- ENABLE – Enable 32-bit Controller SPI address.

The default setting for this preference is DISABLE.

7.21. MSPI_COMMAND_32BIT

This option setup the command format for SPI Flash memory:

- DISABLE – Using 24-bit Controller SPI command.
- ENABLE – Enable 32-bit Controller SPI command.

The default setting for this preference is DISABLE.

7.22. TARGET_IDLE_TIMER

Target Idle Timer setting to prevent system lock when performing segmented bitstream burst.

- DISABLE – Disable the Timer.
- 200S – Set the timer to 200 seconds.
- 100S – Set the timer to 100 seconds.
- 50S – Set the timer to 50 seconds.
- 25S – Set the timer to 25 seconds.
- 10S – Set the timer to 10 seconds.
- 5S – Set the timer to 5 seconds.
- 1S – Set the timer to 1 second.
- 750MS – Set the timer to 750 milliseconds.
- 500MS – Set the timer to 500 milliseconds.
- 250MS – Set the timer to 250 milliseconds.
- 100MS – Set the timer to 100 milliseconds.
- 75MS – Set the timer to 75 milliseconds.
- 50MS – Set the timer to 50 milliseconds.
- 25MS – Set the timer to 25 milliseconds.
- 10MS – Set the timer to 10 milliseconds.

The default setting for this preference is DISABLE.

7.23. MSPI_PREAMBLE_DETECTION_TIMER

Timer setting for preamble detection. Once time out before preamble been detected, the configuration logic declares PREAMBLE failure.

- 200MS – Set the timer to 100 milliseconds.
- 100MS – Set the timer to 100 milliseconds.
- 50MS – Set the timer to 50 milliseconds.
- 20MS – Set the timer to 20 milliseconds.
- 1MS – Set the timer to 1 millisecond.
- 500US – Set the timer to 500 microseconds.
- 100US – Set the timer to 100 microseconds.

The default setting for this preference is 200MS.

7.24. BACKGROUND_RECONFIG

The BACKGROUND_RECONFIG preference controls the behavior regarding transparent access mode for the next PROGRAMN pin toggle or REFRESH command execution event. There are four states to which BACKGROUND_RECONFIG preference can be set:

- OFF – This setting causes the device going to OFFLINE access mode for the next PROGRAMN pin toggle or REFRESH command execution event.
- ON – This setting causes the device going to transparent access mode with SRAM, EBR and IP Write capability for the next PROGRAMN pin toggle or REFRESH command execution event.
- SRAM_ONLY – This setting causes the device going to transparent access mode with SRAM Write capability for the next PROGRAMN pin toggle or REFRESH command execution event.

The default setting for this preference is OFF.

7.25. DAISY_CHAIN

The DAISY_CHAIN preference enable/disable the sysCONFIG daisy chain, and choose the daisy chain mode:

- DISABLE – This setting disables the sysCONFIG daisy chain.
- BYPASS – This setting enables the sysCONFIG daisy chain in BYPASS mode.
- FLOW_THROUGH– This setting enables the sysCONFIG daisy chain in FLOW_THROUGH mode.

The default setting for this preference is DISABLE.

7.26. DAISY_CHAIN_WAIT_DONE

The DAISY_CHAIN_WAIT_DONE preference selects the wakeup mode for the device in sysCONFIG daisy chain:

- DISABLE – Device wakeup after been successfully configured without waiting for the external DONE pin goes high.
- ENABLE – Device waits for the external DONE pin goes high before entering user functional mode.

The default setting for this preference is DISABLE.

7.27. PROGRAMN_RECOVERY

The PROGRAMN_RECOVERY preference recoveries the PROGRAMN as user GPIO:

- DISABLE – PROGRAMN pin serves as sysCONFIG functional pin.
- ENABLE – PROGRAMN pin serves as user GPIO.

The default setting for this preference is DISABLE.

7.28. TRANSFR

The TRANSFR Feature ON/OFF selection for non-disturb IO transfer during re-configuration:

- OFF – Disable TRANSFR.
- ON – Enable TRANSFR.

The default setting for this preference is OFF.

7.29. UserCode_Format

The UserCode_Format preference selects the format for the data field used to assign a value in the USERCODE preference. The USERCODE_FORMAT has four options:

- Binary – USERCODE is set using 32 1 or 0 characters.
- Hex – USERCODE is set using eight hexadecimal digits, that is 0-9A-F.
- ASCII – USERCODE is set using up to four ASCII characters.
- Auto – USERCODE is automatically created by the software. The upper 16-bit is Unique ID and the lower 16 bits are sequentially increased automatically for every bitstream generation.

The default setting for this preference is Binary.

7.30. UserCode

The Lattice Avant device contains a 32-bit register for storing a user-defined value. The default value stored in the register is 0x00000000. Using the USERCODE preference, this register can be initialized with any 32-bit value specified by the user. Suggested uses include the configuration data version number, a manufacturing ID code, date of assembly, or the JEDEC file checksum.

The format of the USERCODE field is controlled using the USERCODE_FORMAT preference with Hex format as default. Data entry can be performed in either Binary, Hex, ASCII or Auto formats.

7.31. Unique_ID

The Lattice Avant device contains a 16-bit register for storing a user-defined value. The default value stored in the register is 0x0000. Unique ID can only be set when USERCODE_FORMAT is Auto.

7.32. MultiBoot_Offset

The Lattice Avant device contains a 48-bit register for storing the booting address (Multiple Boot) for external Flash Memory. The default value stored in the register is 0x000000000000.

7.33. BitstreamRevision_Format

The BitstreamRevision_Format preference selects the format for the data field used to assign a value in the BitstreamRevision preference. The BitstreamRevision_Format has four options:

- Binary – BitstreamRevision is set using 32 1 or 0 characters
- Hex – BitstreamRevision is set using eight hexadecimal digits, that is 0-9A-F
- ASCII – BitstreamRevision is set using up to four ASCII characters
- TIMESTAMP – BitstreamRevision is automatically created by the Time Stamp.

The default setting for this preference is Binary.

7.34. BitstreamRevision

The Avant device contains a 32-bit register for storing the bitstream revision information. The default value stored in the register is 0x00000000. Using the BitstreamRevision preference, this register can be initialized with any 32-bit value you specified.

The format of the BitstreamRevision field is controlled using the BitstreamRevision_Format preference with Binary format as default. Data entry can be performed in either Binary, Hex, ASCII or use TimeStamp.

8. Wake-up Sequence

When configuration is complete (the SRAM has been loaded), the device wakes up in a predictable fashion. If the Avant device is the only device in the chain, or the last device in a chain, the wake-up process should be initiated by the completion of configuration. Once configuration is complete, the internal DONE bit is set and then the wake-up process begins. Figure 8.1 shows the wake-up sequence with DISABLE_DONE_SYNC option.

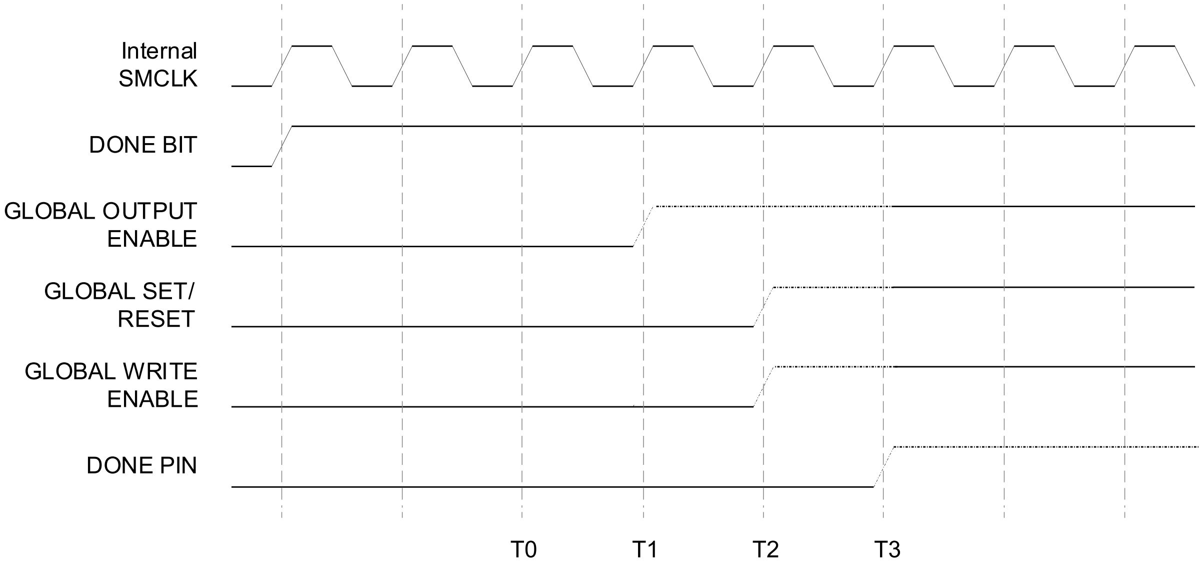


Figure 8.1. Wake-up Sequence Using Internal Clock

8.1. Wake-up Signals

Three internal signals GSRN, GWE, and GOE determine the wake-up sequence.

- GSRN is used to set and reset the core of the device. GSRN is asserted (low) during configuration and de-asserted (high) in the wake-up sequence.
- When the GWE signal is low, it safeguards the integrity of the RAM Blocks and LUTs in the device. This signal is low before the device wakes up. This control signal does not control the primary input pin to the device but controls specific control ports of EBR and LUTs.
- When low, GOE prevents the device's I/O buffers from driving the pins. The GOE only controls output pins. Once the internal DONE is asserted the Avant device responds to input data.
- When high, the DONE pin indicates that configuration is complete and that no errors were detected.
- Before DONE pin goes high, all signals going to EBR should hold steady, otherwise, it might impact the EBR initialization.

The available wake-up sequences are shown in Table 8.1. A wake-up sequence is the order in which the signals change. The phase transition is based on a wakeup clock, as discussed below. The exact timing relationship between the internal signals and the wakeup clock varies and is not specified.

Table 8.1. Wake-up Sequences

Sequence	Phase T0	Phase T1	Phase T2	Phase T3
ENABLE_DONE_SYNC	DONE	GOE	GWE, GSRN	—
DISABLE_DONE_SYNC	—	GOE	GWE, GSRN	DONE

8.2. Wake-up Clock

The Lattice Avant device always uses the internal clock to perform the device wake up sequence. Once the device is configured, it enters the wake-up state, which is the transition from the configuration mode to user mode. This sequence is synchronized to the internal SMCLK, which is derived from internal oscillator.

9. 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 only Flow Through chaining method with first FPGA being target. The *Synchronous to External DONE pin* option (DONE_EX) must be enabled in Lattice Radiant Device Constraint Editor. Based on the configuration ports arrangement, Avant devices can only be the first device inside a configuration daisy chain.

9.1. Flow Through Option

The Flow Through option for the Lattice Avant device can only be used with serial daisy chains, and only support the Lattice FPGA with Target SPI, which has the automatic bitstream mode.

When the first device completes configuration and a flow through command is included with the bitstream, the MCSNO pin is driven low. Once the flow through option starts, the device remains in flow through until the wake-up sequence completes.

An example of the Lattice Avant device in a configuration daisy chain with Flow Through option is shown in Figure 9.1.

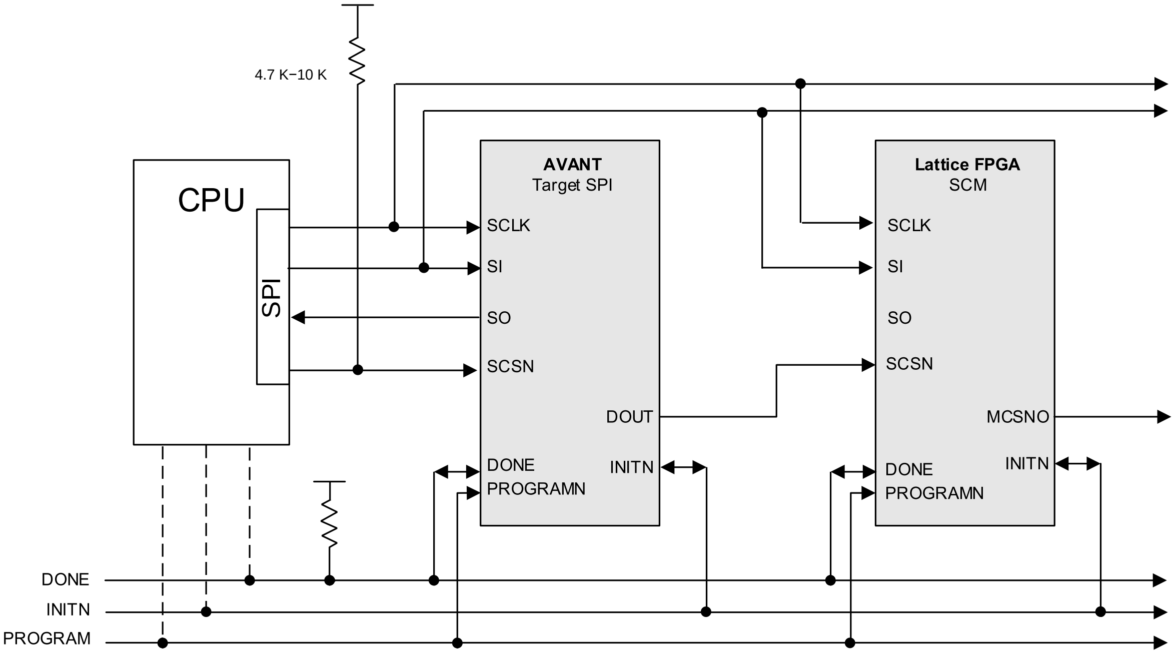


Figure 9.1. Lattice Avant in Configuration Daisy Chain with Target SPI in Flow Through Mode

Appendix A. Avant Target SPI Programming Guide

The 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 -> SISPI
- ISPEN -> SN
- TCK -> SCLK
- TDO -> SPISO

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 for 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 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

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 and Reference Designs for Avant-E
- Development Kits and Boards for Avant-E

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)

Other references:

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

Technical Support Assistance

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

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

Revision History

Revision 0.85, March 2024

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

Section	Change Summary
	- 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. - 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.

LATTICE SEMICONDUCTOR®

www.latticesemi.com