# Lattice Avant Configuration Access through CONFIG_LMMIC Primitive

## *Preliminary* Reference Design

FPGA-RD-02334-0.80

January 2026

## Disclaimers

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

## Inclusive Language

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

# Contents

# Figures



# Tables

# Abbreviations in This Document

A list of abbreviations used in this document.

| Abbreviation | Definition |
|---|---|
| AS2 | Alpha Sample 2 |
| CFG | Configuration |
| FPGA | Field Programmable Gate Array |
| HDL | Hardware Description Language |
| IP | Intellectual Property |
| JTAG | Joint Test Action Group |
| LED | Light Emitting Diode |
| LMMI | Lattice Memory Mapped Interface |
| NOOP | No Operation |
| OSC | Oscillator |
| RTL | Register Transfer Level |
| USB | Universal Serial Bus |

# 1. Introduction

This reference design demonstrates using the Lattice memory mapped interface (LMMI) host logic to drive the LMMI interface of the CONFIG_LMMIC primitive. The LMMI protocol is used to perform LMMI read operations. When accessing any configuration (CFG) internal register or any other CFG data, the non-JTAG target command format must be followed. The supported commands are listed in the CONFIG_LMMIC Supported Commands section in the Lattice Avant sysCONFIG User Guide (FPGA-TN-02299).

By default, the reference design performs two LMMI read operations. These read operations demonstrate the READ_IDCODE_PUB command to read the 32-bit public IDCODE of the device and the READ_USERCODE command to read the 32-bit user code set through the Global tab of the Device Constraint Editor in the Lattice Radiant™ software. However, the reference design provides the flexibility to define the read operation commands as parameters in cmd_list.v.

**Note:** This reference design is developed using the Lattice Avant™ LAV-AT-X70 device but can be applied to other Avant devices.

## 1.1. Quick Facts

Download the reference design files from the Lattice reference design web page.

**Table 1.1. Summary of the Reference Design**

| | | |
|---|---|---|
| **General** | Target Devices | LAV-AT-X70-3LFG1156I |
| | Source code format | Verilog |
| **Simulation** | Functional simulation | Not performed |
| | Timing simulation | Not performed |
| | Test bench | Not available |
| | Test bench format | Not available |
| **Software Requirements** | Software tool and version | Lattice Radiant™ Software 2024.2.1 |
| | IP version | OSC Module v2.1.0<br>CONFIG_LMMIC Primitive |
| **Hardware Requirements** | Board | Avant-X Versa Board |
| | Cable | USB-A to Mini-B programming cable |

## 1.2. Feature

Key features of the Configuration Access through CONFIG_LMMIC Primitive reference design include:

- Ability to interface with the CONFIG_LMMIC primitive using the LMMI host logic
- Supports read operation for non-JTAG target commands

## 1.3. Naming Conventions

### 1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog hardware description language (HDL).

### 1.3.2. Signal Names

Signal names that end with:

- *_n* are active low signals (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals

# 2. Directory Structure and Files

Figure 2.1 shows the directory structure.

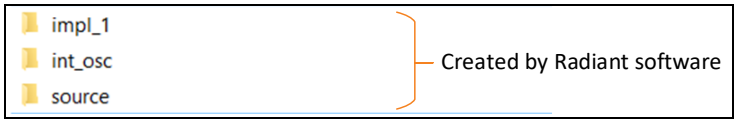



**Figure 2.1. Directory Structure**

The Configuration Access through CONFIG_LMMIC Primitive reference design includes the bitstream file, instantiated intellectual property (IP) files, and register transfer level (RTL) source codes.

Table 2.1 shows the list of directories included in the reference design package.

**Table 2.1. Directory List**

| Directory | Description |
|---|---|
| impl_1 | Contains the bitstream (.bit) file. |
| int_osc | Contains the generated Oscillator Module package, which includes files as shown in Table 2.2. |
| source | Contains the RTL source code. |

Table 2.2 shows the list of files included in the reference design package for the generated IP.

**Table 2.2. File List**

| Attribute | Description |
|---|---|
| <Component name>.ipx | Contains the information on the files associated with the generated IP. |
| <Component name>.cfg | Contains the parameter values used in the IP configuration. |
| component.xml | Contains the ipxact: component information of the IP. |
| design.xml | Documents the configuration parameters of the IP in IP-XACT 2014 format. |
| rtl/<Component name>.v | Provides an example RTL top file that instantiates the module. |
| rtl/<Component name>_bb.v | Provides the synthesis closed box. |
| misc/<Component name>_tmpl.v<br>misc /<Component name>_tmpl.vhd | Provide instance templates for the module. |

# 3. Functional Description

Figure 3.1 shows the top-level block diagram of the reference design. The blocks shown are the fundamental components of the reference design.

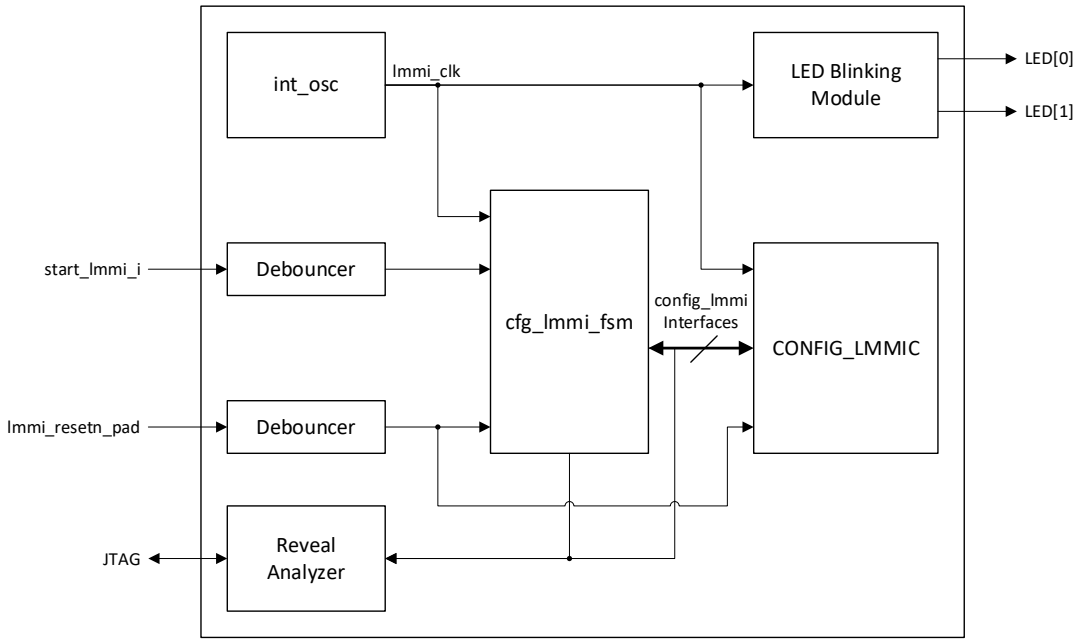



**Figure 3.1. Reference Design Block Diagram**

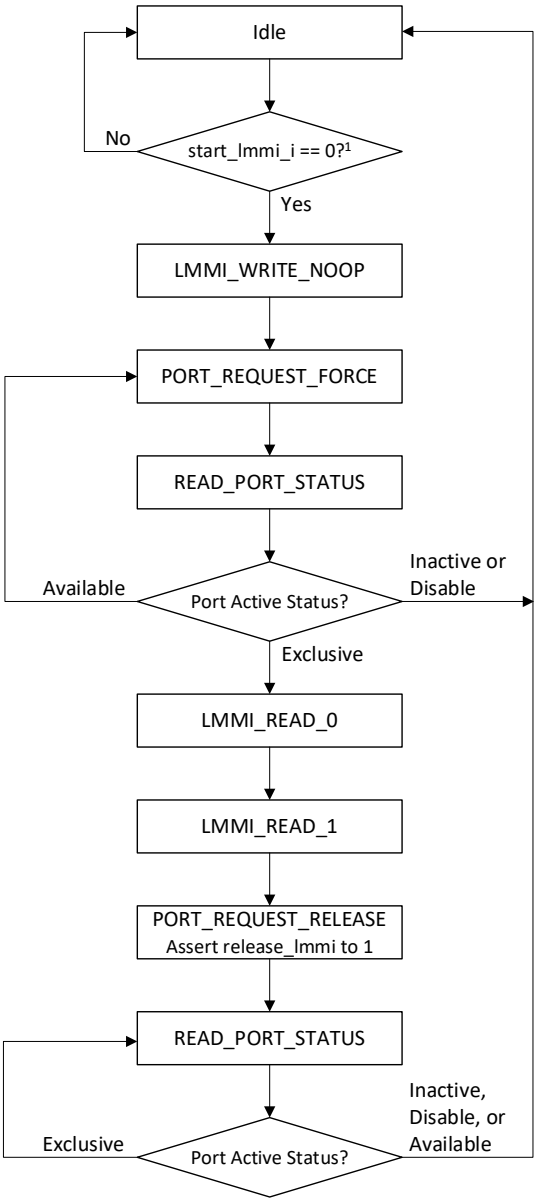
## 3.1. Design Components

The Configuration Access through CONFIG_LMMIC Primitive reference design includes the following blocks:

- int_osc: An oscillator (OSC) module that generates the clock source (lmmi_clk) for the cfg_lmmi_fsm module, CONFIG_LMMIC primitive, and the light emitting diode (LED) blinking module. lmmi_clk runs at 25 MHz. Refer to the Lattice Avant OSC Module User Guide (FPGA-IPUG-02184) for more information.
- LED blinking module: This module implements a counter to blink the two LEDs on the board.
- Reveal analyzer : The Reveal analyzer is used to view the LMMI interfaces and selected signals in the cfg_lmmi_fsm module to monitor LMMI operations and data values of read operations.
- Debouncer: This block eliminates noise or fluctuations (also known as bouncing) in a digital input signal, which can occur when a mechanical switch is pressed or released.
- CONFIG_LMMIC: LMMI interface to the configuration block. Refer to the CONFIG_LMMIC page in the Lattice Radiant Software Help for more information.
- cfg_lmmi_fsm: This module implements a state machine controller to send the necessary commands to the CONFIG_LMMIC primitive to perform the two LMMI read operations. The following describes the LMMI operation flow as shown in Figure 3.2:
  - IDLE: Waits for the input signal (start_lmmi_i) to start LMMI operations.
  - LMMI_WRITE_NOOP: Executes 32 bytes of no operation (NOOP) command. For additional information on this operation, refer to the LMMI_WRITE_NOOP section. Note that this operation is only required for the Avant alpha sample 2 (AS2) device and is optional for production devices.
  - PORT_REQUEST_FORCE: Executes a forced port request to obtain exclusive access to the configuration engine. This ensures that the port request always succeeds, even if another port previously held exclusive access. To obtain exclusive access for other request types such as Nowait or Wait, refer to the Lattice Avant sysCONFIG User Guide (FPGA-TN-02299) for more information. For additional information on this operation, refer to the PORT_REQUEST_FORCE section.

- READ_PORT_STATUS: Reads the port status (LMMI_CFG_PORT_STATUS[15:0]) for the CONFIG_LMMI interface. Proceeds to the next state once exclusive access is obtained. If the port is available, returns to the PORT_REQUEST_FORCE state to obtain exclusive access. If the port is inactive or disabled, returns to the idle state. For additional information on this operation, refer to the READ_PORT_STATUS for Exclusive Access section.
- LMMI_READ_0: Performs an LMMI read operation using the READ_IDCODE_PUB command to read the 32-bit public IDCODE of the device. Sends the 32-bit command in two LMMI clock cycles. For additional information on this operation, refer to the LMMI_READ_0 section.
- LMMI_READ_1: Performs an LMMI read operation using the READ_USERCODE command to read the 32-bit user code set through the Global tab of the Device Constraint Editor in the Lattice Radiant software. Sends the 32-bit command in two LMMI clock cycles. For additional information on this operation, refer to the LMMI_READ_1 section.
- PORT_REQUEST_RELEASE: Releases exclusive access to the configuration engine to ensure the flow gracefully exits the active state. For additional information on this operation, refer to the PORT_REQUEST_RELEASE section.
- READ_PORT_STATUS: Reads the port status (LMMI_CFG_PORT_STATUS[15:0]) for the CONFIG_LMMI interface. If the port is available and release_lmmi = 1, the LMMI port has been released and the flow returns to the idle state. If the port is inactive or disabled, the flow returns to the idle state. For additional information on this operation, refer to the READ_PORT_STATUS for Released Exclusive Access section.

Refer to Table 6.12. Target Configuration Commands and Table D.2. LMMI CFG Offset List in the Lattice Avant sysCONFIG User Guide (FPGA-TN-02299) for more information.




**Note:**
1. The start_lmmi_i signal is controlled by the SW13 momentary-contact push button.

**Figure 3.2. LMMI Operation Flow Chart**

# 4. Reference Design Parameter Description

The Configuration Access through CONFIG_LMMIC Primitive reference design includes the parameters shown in Table 4.1. You can modify the parameters by editing the cmd_list.v file as needed, such as to execute the desired LMMI read operation.

**Table 4.1. Parameters in cmd_list.v**

| Parameter | Default Value | Description |
|---|---|---|
| COMMAND_WORD_H_0 | 16'h0101 | Upper 16 bits of the 32-bit target command. The default value is the READ_IDCODE_PUB command. |
| COMMAND_WORD_L_0 | 16'h0000 | Lower 16 bits of the 32-bit target command. The default value is the READ_IDCODE_PUB command. |
| DATA_BYTES_0 | 8'h04 | Number of returned data bytes for LMMI_READ_0 operation. |
| COMMAND_WORD_H_1 | 16'h0105 | Upper 16 bits of the 32-bit target command. The default value is the READ_USERCODE command. |
| COMMAND_WORD_L_1 | 16'h0000 | Lower 16 bits of the 32-bit target command. The default value is the READ_USERCODE command. |
| DATA_BYTES_1 | 8'h04 | Number of returned data bytes for LMMI_READ_1 operation. |

# 5. Signal Description

The input/output interface signals for the top-level unit (top_wrapper.v) are shown in Table 5.1.

**Table 5.1. Primary I/O**

| Port Name | I/O | Width | Description |
|---|---|---|---|
| LED[0] | Out | 1 | Output to blink LED D45. |
| LED[1] | Out | 1 | Output to blink LED D46. |
| start_lmmi_i | In | 1 | Active low input port. Toggle the SW13 push button to start LMMI operations. |
| lmmi_resetn_pad | In | 1 | Active low asynchronous reset port. Toggle the SW12 push button to assert the reset. |

# 6. Running the Reference Design

This section describes how to run the Configuration Access through CONFIG_LMMIC Primitive reference design using the Lattice Radiant software. For more details on the Lattice Radiant software, refer to the Lattice Radiant Software User Guide.

## 6.1. Compiling the Reference Design

The reference design package includes the pre-compiled files and bitstream (.bit) to facilitate a quick start. However, you can recompile the reference design through the standard compilation flow in a newer Radiant software version or after modifying the user logic (Verilog code).

## 6.2. Generating the Bitstream File

This section provides the procedure of creating your field programmable gate array (FPGA) bitstream file using the Lattice Radiant Software.

To create the FPGA bitstream file, follow the steps below.

1. Open the Lattice Radiant software, as shown in Figure 6.1.

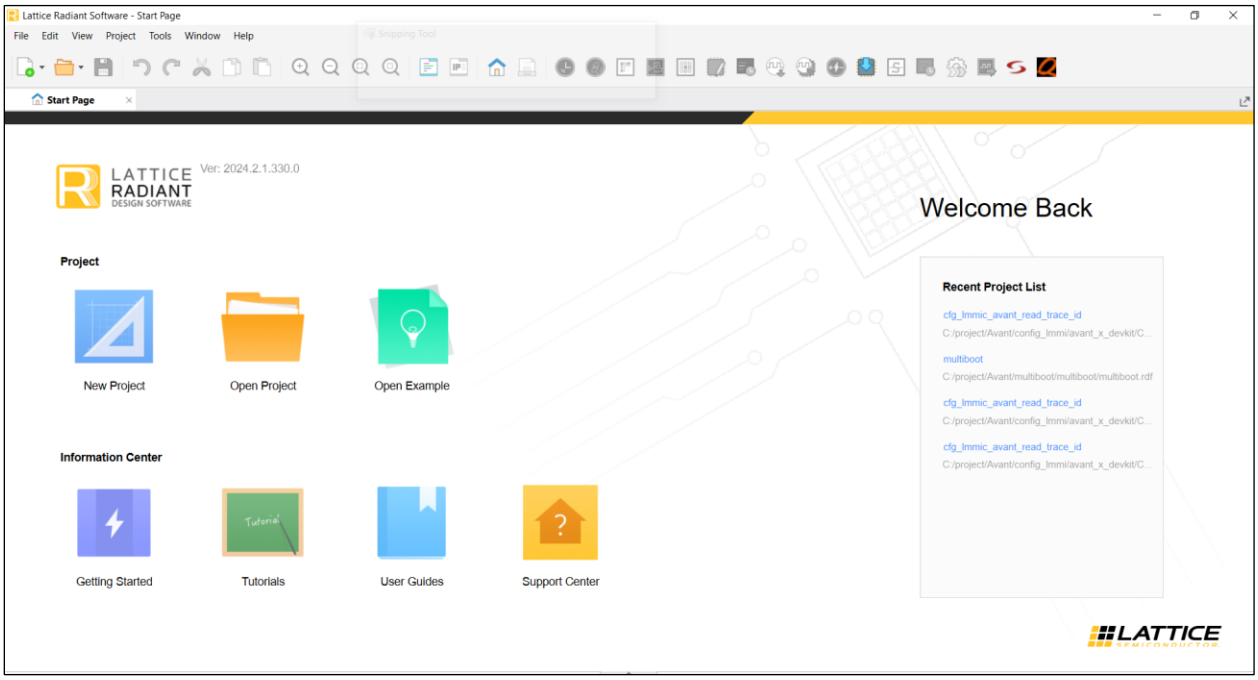



**Figure 6.1. Lattice Radiant Software**

2. Click **File > Open Project**. From the project database, navigate to the cfg_lmmic_avant folder of the reference design project and open the Lattice Radiant project file cfg_lmmic_avant.rdf as shown in Figure 6.2.

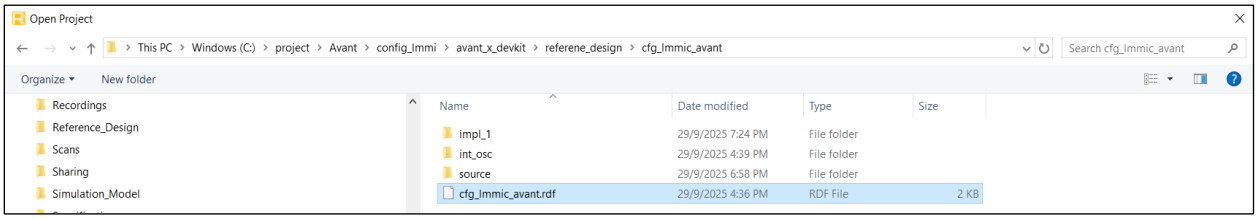



**Figure 6.2. Open Project File**

3. Click **Export Files** to generate the bit file. View the log message in the Export Reports folder for the generated bitstream. Once compilation is successful, you can find the generated bit file in the project implementation folder (for example, cfg_lmmic_avant\impl_1\ cfg_lmmic_avant_impl_1.bit).
   **Note:** Repeat this step if you need to recompile the design for other design implementations.

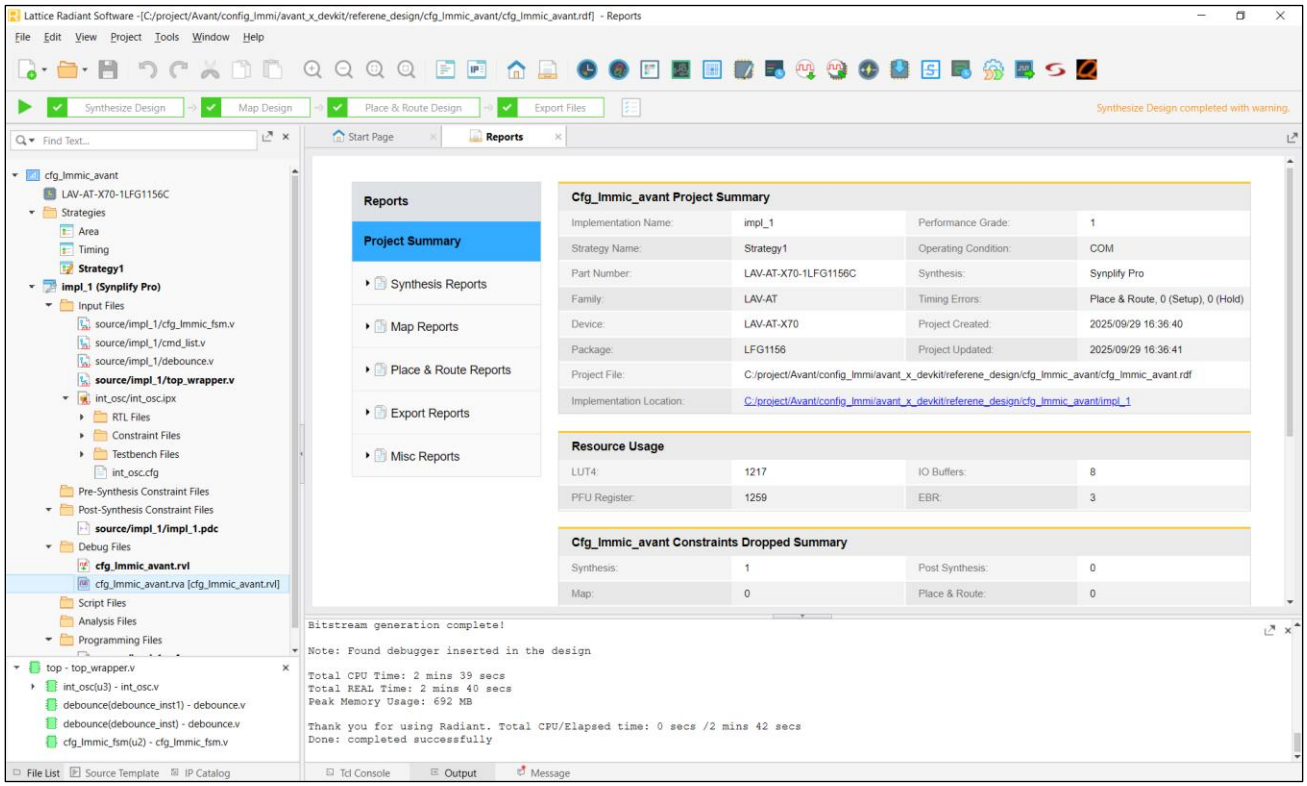



**Figure 6.3. Generated Bitstream Log**

# 7. Implementing the Reference Design on Board

## 7.1. Hardware Requirements

The following are the hardware requirements:

- Avant-X Versa Board
- USB-A to mini-B programming cable

## 7.2. Configuring the FPGA through JTAG Port

To configure the FPGA through a Joint Test Action Group (JTAG) port, follow the steps below:

1. Launch the Radiant Programmer software and select **Edit** > **Device Properties**.
2. Configure the device properties as shown in Figure 7.1, then click **OK**.

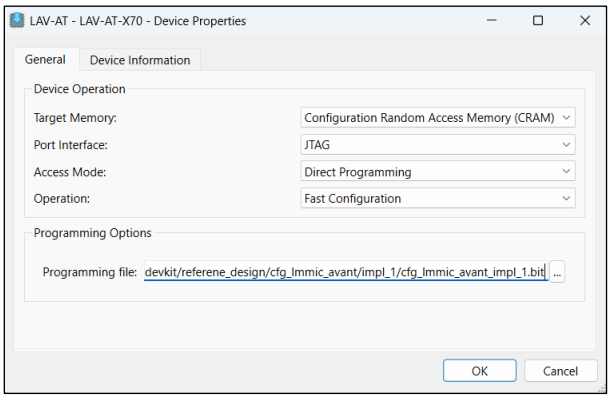



**Figure 7.1. Device Properties Window**

3. In the **Cable Settings** section of the **Cable Setup** panel, set **Port** to *FTUSB-1*.
4. Click the **Program Device** button to start programming the board.


**Figure 7.2. Radiant Programmer Window**

## 7.3. Running the Design

### 7.3.1. Performing and Viewing LMMI Operations

The LEDs D45 and D46 start blinking continuously once the device is configured successfully.

To perform and view the LMMI operations using the Reveal analyzer, follow the steps below:

1. Open the cfg_lmmic_avant.rdf project located in the project directory in the Lattice Radiant software.
2. Run the Reveal analyzer by double-clicking the cfg_lmmic_avant.rva file as shown in Figure 7.3.

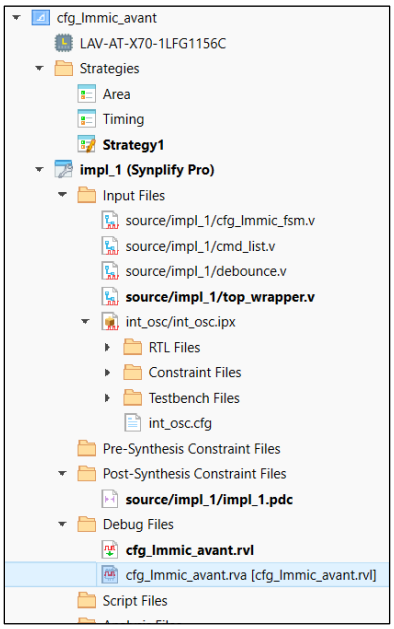



**Figure 7.3. cfg_lmmic_avant.rva File**

3. On the **Reveal Analyzer/Controller** tab, click the button.

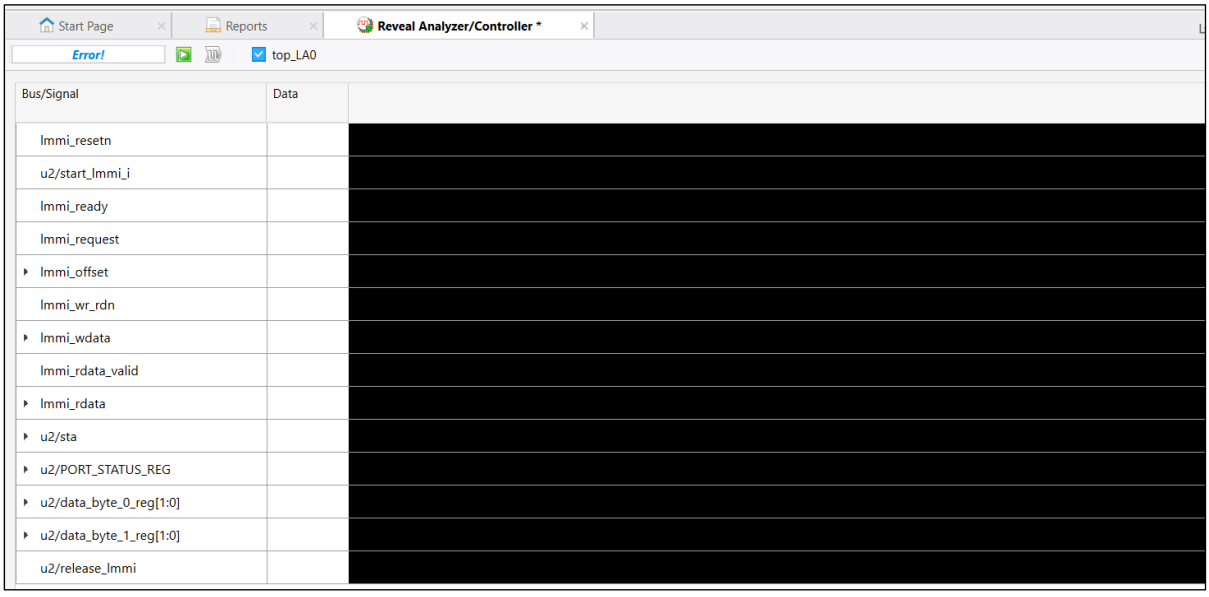



**Figure 7.3 Reveal Analyzer – Waiting for Trigger**

4. Press the SW13 momentary-contact push button. LMMI operation waveforms appear as shown in Figure 7.4. Use the zoom controls to examine specific LMMI write or read operations.

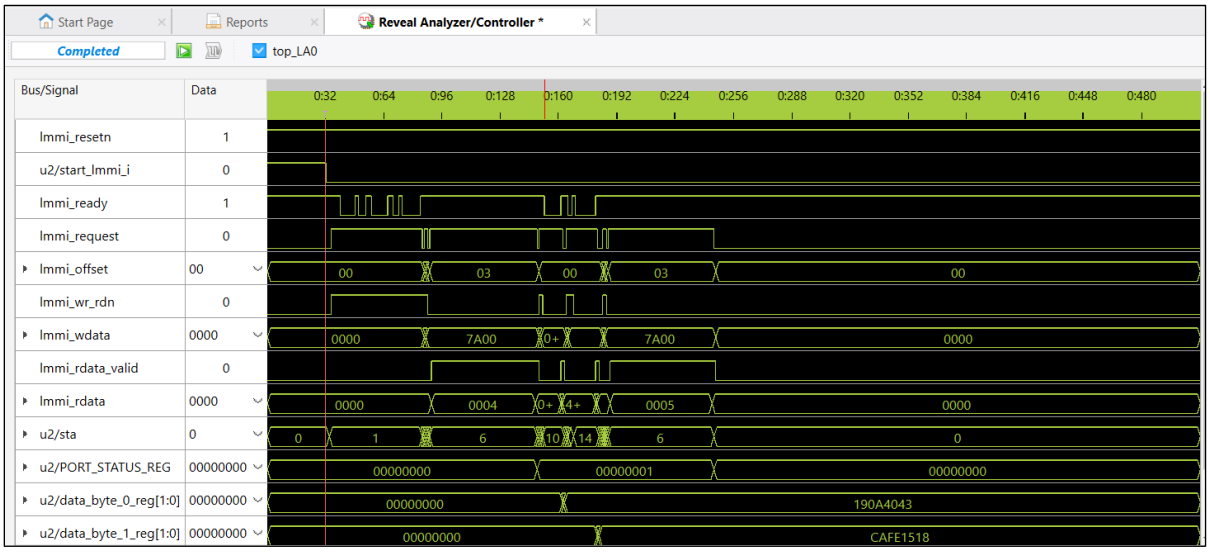



**Figure 7.4. LMMI Operation Waveforms in Reveal Analyzer**

## 7.3.2. LMMI Operations

### 7.3.2.1. LMMI_WRITE_NOOP

The LMMI_WRITE_NOOP operation executes 32 bytes of NOOP command.

- LMMI Offset = 8'h00
- LMMI_CFG_DATA[15:0] = 16'h0000

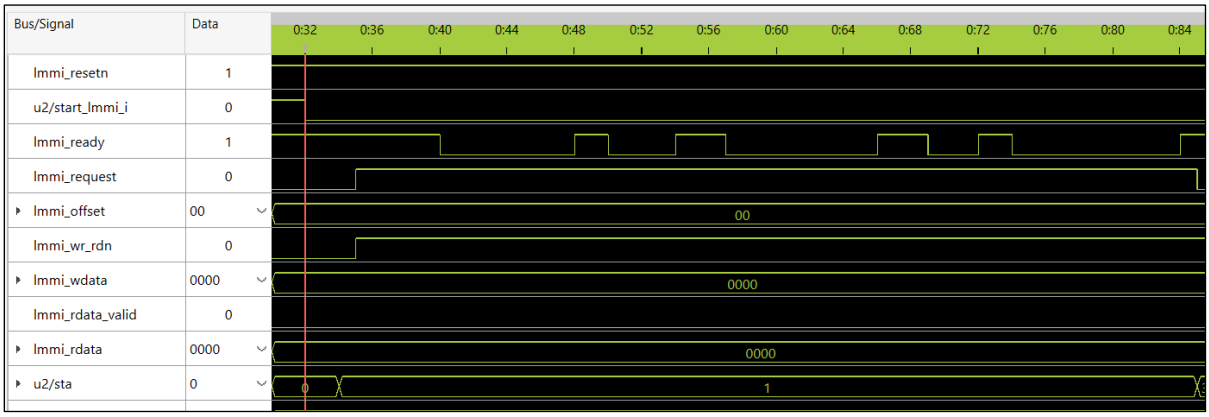



**Figure 7.5. LMMI_WRITE_NOOP**

### 7.3.2.2. PORT_REQUEST_FORCE

The PORT_REQUEST_FORCE operation executes a forced port request to obtain exclusive access to the configuration engine.

- LMMI Offset = 8'h01 and LMMI_CFG_PORT_REQUEST[15:0] = 16'h0002
- LMMI Offset = 8'h02 and LMMI_CFG_PORT_REQUEST[31:16] = 16'h7A00

**Note:** Write LMMI_CFG_PORT_REQUEST[15:0] first followed by LMMI_CFG_PORT_REQUEST[31:16]. Failure to follow this sequence might cause the LMMI port to become inactive, thus requiring power cycling of the device to recover the LMMI port.

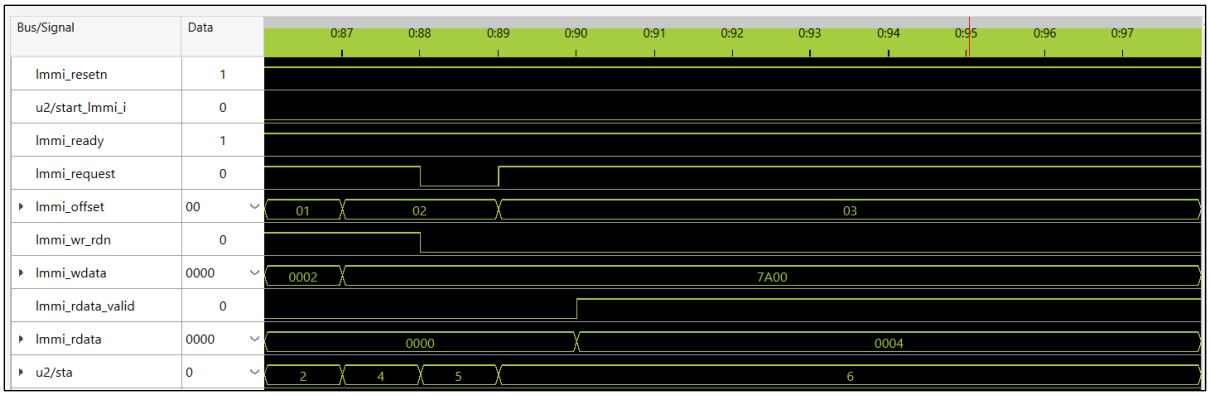



**Figure 7.6. PORT_REQUEST_FORCE**

### 7.3.2.3. READ_PORT_STATUS for Exclusive Access

The READ_PORT_STATUS operation reads the port status (LMMI_CFG_PORT_STATUS[15:0]) for the CONFIG_LMMI interface.

- LMMI Offset = 8'h03

Port status data is available once lmmi_rdata_valid is asserted high. lmmi_rdata = 16'h0004, as shown in Figure 7.7, indicates that the device is busy processing the PORT_REQUEST command . lmmi_rdata transitioning from 16'h0004 to 16'h0001, as shown in Figure 7.8, indicates that the LMMI port is active and has exclusive control.

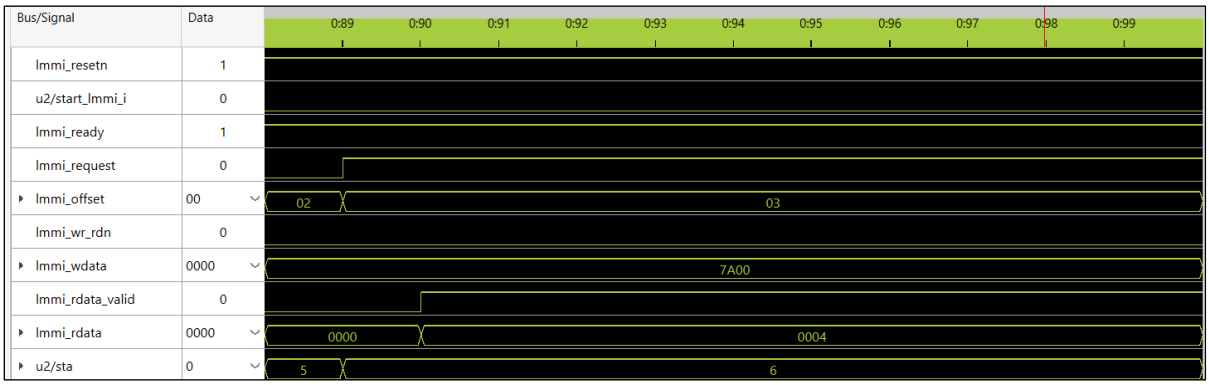



**Figure 7.7. Device is Busy Processing the PORT_REQUEST Command**

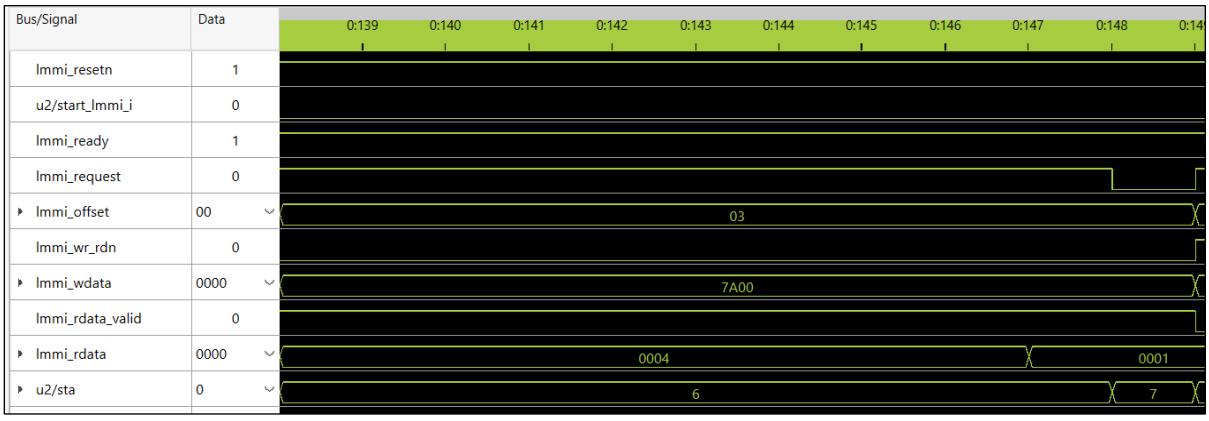



**Figure 7.8. LMMI Port is Active and Has Exclusive Control**

#### 7.3.2.4. LMMI_READ_0

The LMMI_READ_0 operation performs an LMMI read operation using the READ_IDCODE_PUB command to read the 32-bit public IDCODE of the device. It sends the 32-bit command in two LMMI clock cycles.

- First cycle: LMMI Offset = 8'h00 and LMMI_CFG_DATA[15:0] = 16'h0101
- Second cycle: LMMI Offset = 8'h00 and LMMI_CFG_DATA[15:0] = 16'h0000

Once lmmi_rdata_valid is asserted high, you can read the 32-bit public IDCODE in two LMMI clock cycles. The IDCODE value is 32'h190A_4043, which is the IDCODE for the LAV-AT-X70 AS2 device.

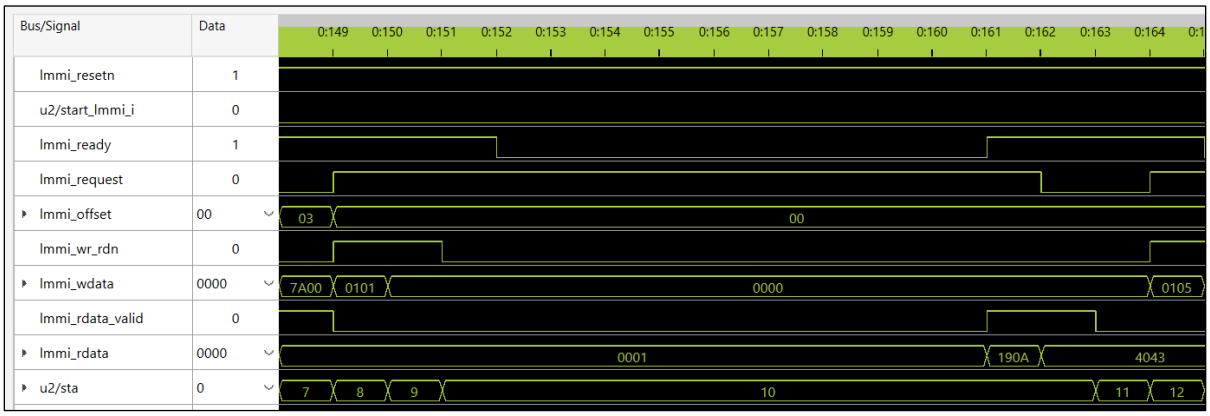



**Figure 7.9. Read Public IDCODE**

#### 7.3.2.5. LMMI_READ_1

The LMMI_READ_1 operation performs an LMMI read operation using the READ_USERCODE command to read the 32-bit user code set through the Global tab of the Device Constraint Editor in the Lattice Radiant software. It sends the 32-bit command in two LMMI clock cycles.

- First cycle: LMMI Offset = 8'h00 and LMMI_CFG_DATA[15:0] = 16'h0105
- Second cycle: LMMI Offset = 8'h00 and LMMI_CFG_DATA[15:0] = 16'h0000

Once lmmi_rdata_valid is asserted high, you can read the 32-bit user code in two LMMI clock cycles. The IDCODE value is 32'hCAFE_1518, which matches the user code value set in the Radiant software.

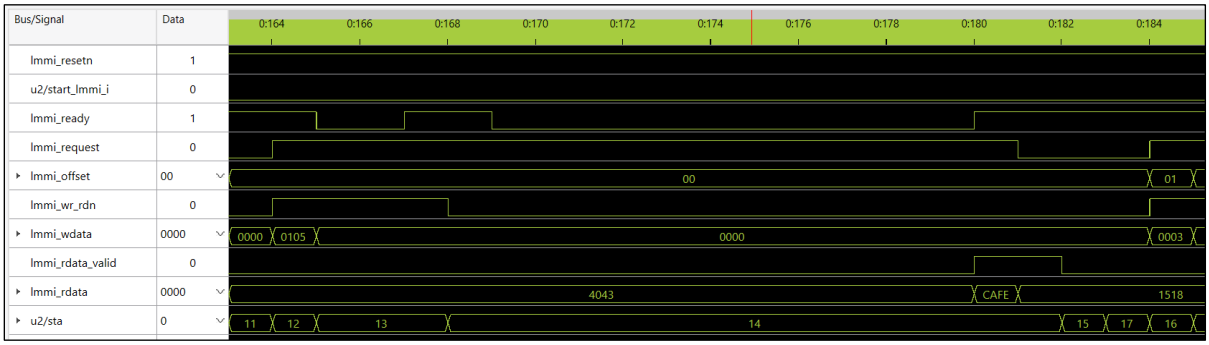



**Figure 7.10. Read User Code**

#### 7.3.2.6. PORT_REQUEST_RELEASE

The PORT_REQUEST_RELEASE operation releases exclusive access to the configuration engine to ensure the LMMI operation flow gracefully exits the active state.

- LMMI Offset = 8'h01 and LMMI_CFG_PORT_REQUEST[15:0] = 16'h0003
- LMMI Offset = 8'h02 and LMMI_CFG_PORT_REQUEST[31:16] = 16'h7A00

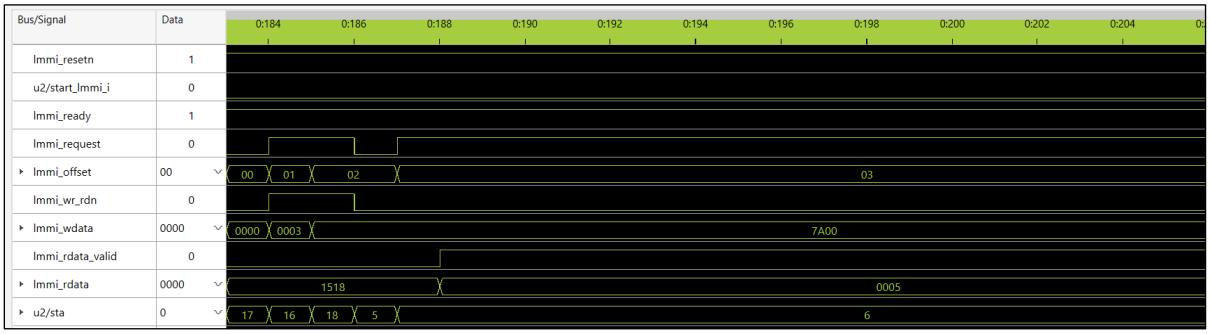



**Figure 7.11. PORT REQUEST Command with Release Request**

#### 7.3.2.7. READ_PORT_STATUS for Released Exclusive Access

The READ_PORT_STATUS operation reads the port status (LMMI_CFG_PORT_STATUS[15:0]) for the CONFIG_LMMI interface.

- LMMI Offset = 8'h03

Port status data is available once lmmi_rdata_valid is asserted high. lmmi_rdata = 16'h0005 indicates the device is busy processing the PORT_REQUEST command and the LMMI port is still active with exclusive control. lmmi_rdata transitioning from 16'h0005 to 16'h0000, as shown in Figure 7.12, indicates that the LMMI port is active but in the non-exclusive state. This means that the LMMI port has successfully released exclusive access.

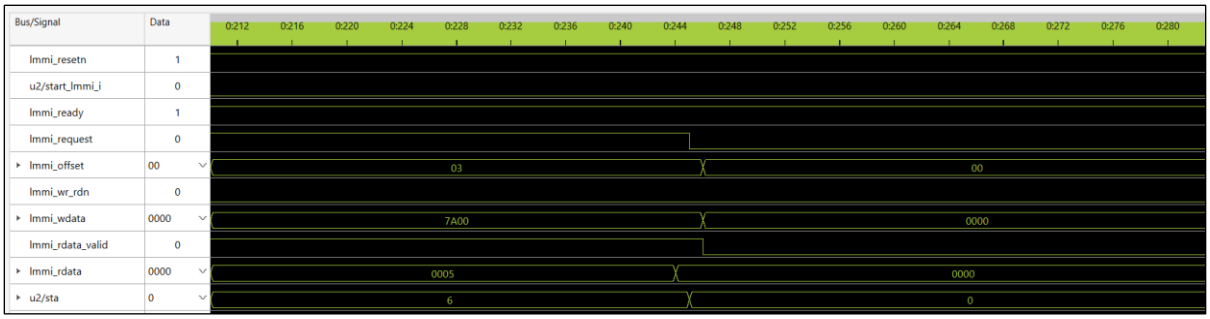



**Figure 7.12. LMMI Port Release to Active but Non-exclusive State**

# References

- Avant sysCONFIG User Guide (FPGA-TN-02299)
- Lattice Avant OSC Module User Guide (FPGA-IPUG-02184)
- Avant-X web page
- Avant-X Versa Board web page
- Lattice Avant Platform web page
- Lattice Radiant Software web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

# Technical Support Assistance

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

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

# Revision History

**Revision 0.80, January 2026**

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


www.latticesemi.com