JTAG Bridge IP – Lattice Propel Builder 2026.1

IP Version: v1.1.0

User Guide

FPGA-IPUG-02324-1.0

June 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
AXI4	Advanced eXtensible Interface 4
EBR	Embedded Block RAM
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FTDI	Future Technology Devices International
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
I/O	Input/Output
IP	Intellectual Property
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LSB	Least Significant Bit
LUT	Lookup Table
PAR	Place and Route
RAM	Random Access Memory
RISC-V	Reduced Instruction Set Computer – V (Five)
SoC	System on Chip
TAP	Test Access Port
TCL	Tool Command Language
USB	Universal Serial Bus

1. Introduction

The JTAG Bridge IP provides an efficient solution for debugging on-board issues by allowing you to access memory and peripheral registers directly using this IP, without involving the processor. It converts Joint Test Action Group (JTAG) Test Access Port (TAP) signals into Advanced eXtensible Interface 4 (AXI4) transactions, enabling you to perform read and write operations with byte, halfword, or word lengths.

The JTAG Bridge IP is implemented using Verilog HDL, and it can be configured and generated using the Lattice Propel™ Builder software. The IP supports Certus™-N2, Lattice Avant™, MachXO5™-NX, CrossLinkU™-NX, CrossLink™-NX, CertusPro™-NX, and Certus™-NX FPGA devices.

1.1. What's New in This IP Release

Added JTAGHUB support.

1.2. Quick Facts

Table 1.1 presents a summary of the JTAG Bridge IP.

Table 1.1. JTAG Bridge IP Quick Facts

IP Requirements	Supported Devices	Certus-N2, Lattice Avant, MachXO5-NX, CrossLinkU-NX, CrossLink-NX, CertusPro-NX, and Certus-NX
Resource Utilization	Supported User Interfaces	AXI4 Interface, JTAG Interface
	Resources	See Table A.1 and Table A.2.
Design Tool Support	Lattice Implementation	IP v1.1.0 – Lattice Propel Builder 2026.1
	Simulation	No simulation module available.

1.3. Features

The JTAG Bridge IP has the following features:

- Supports AXI4 Interface.
- Supports configurable address width.
- Supports data mask and data size.
- Supports word-only access, or a combination of word, halfword, and byte operations.

1.4. Conventions

1.4.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.4.2. Signal Names

Signal Names that end with:

- _n are active low.
- _i are input signals.
- _o are output signals.
- _io are bidirectional input/output signals.

1.5. Licensing and Ordering Information

The JTAG Bridge IP is provided at no additional cost with the Lattice Propel design environment. The IP can be fully evaluated in hardware without requiring an IP license string.

2. Functional Descriptions

2.1. Overview

The JTAG Bridge IP uses memory-mapped registers to convert JTAG TAP to AXI4 read or write operations. You can use Tool Command Language (TCL) scripts to transmit Address, Data, and Mask information. The IP returns the exact value for the given address or returns an error status if the system hangs. Figure 2.1 shows the JTAG Bridge IP diagram.

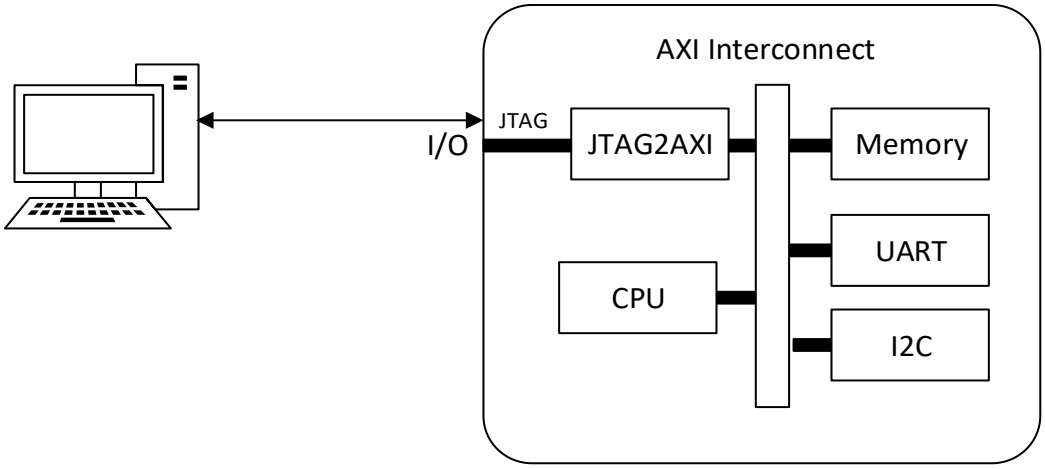

Figure 2.1. JTAG Bridge IP Diagram

2.2. Modules Description

The JTAG Bridge IP output interface is fixed to the AXI4 interface. The input interface is fixed to the formal JTAG interface for the initial version.

The IP can be set to either the JTAGHUB mode or Standard mode. JTAGHUB is a hardened block, and the Lattice Radiant™ software handles pin assignment automatically. For board users, you can use it directly with the USB Dual RS232-HS cable, as with the Programmer tool. The Standard mode requires you to export the JTAG interface pins to the top level and assign them to valid pins on the board. The TCK signal should be assigned to the pclk pin. You should use the HW-USBN-2B cable to connect the pins to a USB cable.

The AXI4 interface should be connected to the targeted memory or peripherals by AXI Interconnect. The default data width is 32 bits. Enable mask and size settings to access the memory space with 8-bit or 16-bit transactions.

The Lattice Propel Builder software provides user-friendly TCL scripts to use this IP. Refer to the JTAG Bridge IP Generation and JTAG Bridge Demo Template sections for details.

2.3. Signal Description

Table 2.1, Table 2.2, and Table 2.3 list the ports of the soft IP in different categories.

2.3.1. Clock and Reset

Table 2.1. Clock and Reset Ports

Name	Direction	Width	Description
clk_i	In	1	SoC clock domain.
rstn_i	In	1	SoC clock domain reset, active low.

2.3.2. AXI4 Interface

Table 2.2. AXI4 Manager Ports

Name	Direction	Width	Group	Description
dBusAxi_aw_ready	In	1	AXI4 Mandatory Write Address Channel	—
dBusAxi_aw_valid	Out	1		—
dBusAxi_aw_payload_addr	Out	32		—
dBusAxi_aw_payload_len	Out	8		—
dBusAxi_aw_payload_size	Out	3		—
dBusAxi_aw_payload_burst	Out	2		Not implemented.
dBusAxi_aw_payload_lock	Out	1		Not implemented.
dBusAxi_aw_payload_cache	Out	4		Not implemented.
dBusAxi_aw_payload_prot	Out	3		Not implemented.
dBusAxi_aw_payload_qos	Out	4		Not implemented.
dBusAxi_aw_payload_region	Out	4		Not implemented.
dBusAxi_aw_payload_id	Out	1		Not implemented.
dBusAxi_w_ready	In	1	AXI4 Mandatory Write Data Channel	—
dBusAxi_w_valid	Out	1		—
dBusAxi_w_payload_data	Out	32		—
dBusAxi_w_payload_last	Out	1		Not implemented.
dBusAxi_w_payload_strb	Out	4		—
dBusAxi_b_valid	In	1	AXI4 Mandatory Write Response Channel	—
dBusAxi_b_payload_resp	In	2		b'00: OKAY[1] b'10: SLVERR[1] b'11: DECERR[1]
dBusAxi_b_payload_id	In	1		Not implemented.
dBusAxi_b_ready	Out	1		—
dBusAxi_ar_valid	Out	1	AXI4 Mandatory Read Address Channel	—
dBusAxi_ar_ready	In	1		—
dBusAxi_ar_payload_cache	Out	4		Not implemented.
dBusAxi_ar_payload_prot	Out	3		Not implemented.
dBusAxi_ar_payload_qos	Out	4		Not implemented.
dBusAxi_ar_payload_region	Out	4		Not implemented.
dBusAxi_ar_payload_id	Out	1		Not implemented.
dBusAxi_ar_payload_addr	Out	32		—
dBusAxi_ar_payload_len	Out	8		—
dBusAxi_ar_payload_size	Out	3		—
dBusAxi_ar_payload_burst	Out	2		Fixed 2'b01.
dBusAxi_ar_payload_lock	Out	1		Not implemented.
dBusAxi_r_payload_id	In	1	AXI4 Mandatory Read Data Channel	Not implemented.
dBusAxi_r_payload_data	In	32		—
dBusAxi_r_payload_resp	In	2		b'00: OKAY[1] b'10: SLVERR[1] b'11: DECERR[1]
dBusAxi_r_payload_last	In	1		—

Name	Direction	Width	Group	Description
dBusAxi_r_valid	In	1		—
dBusAxi_r_ready	Out	1		—

Note:

1. Refer to the AMBA AXI Protocol Specification for more detailed descriptions of these responses.

2.3.3. JTAG Interface

The JTAG interface is only shown in the Standard mode.

Table 2.3. JTAG Ports

Name	Type	Width	Description
TCK	In	1	JTAG clock. Assign it to pclk.
TDI	In	1	—
TMS	In	1	—
TDO	Out	1	—

2.4. Attribute Summary

The configurable attributes of the JTAG Bridge IP are listed and described in Table 2.4 and Table 2.5.

The attributes can be configured through the Lattice Propel Builder software.

Table 2.4. Attributes Summary

Attribute	Values	Default	Dependency on Other Attributes
Mode	JTAGHUB, Standard	JTAGHUB	—
JTAG Channel	14, 15, 16	14	Mode == JTAGHUB
Bus Type	AXI4	AXI4	—
Address Width	7 to 32	32	—
AXI ID Width	1 to 15	4	—
AXI Port ID Number	0 to $2^{\text{AXI ID WIDTH}}-1$	0	—
Enable Mask and Size Settings	Enabled, Disabled	Enabled	—

Table 2.5. Attributes Description

Attribute	Description
Mode	Implementation type of JTAG Bridge IP
JTAG Channel	JTAGHUB channel number
Bus Type	Interface type
Address Width	Address width
AXI ID Width	AXI Channel ID width
AXI Port ID Number	AXI Channel ID number
Enable Mask and Size Settings	Enable the customized size and mask function.

3. JTAG Bridge IP Generation

This section provides information on how to generate the JTAG Bridge IP module using the Lattice Propel Builder software.

To generate the JTAG Bridge IP module:

1. In the Lattice Propel Builder software, create a new design. Select the JTAG Bridge package.
2. Enter the component name. Click **Next**, as shown in Figure 3.1.

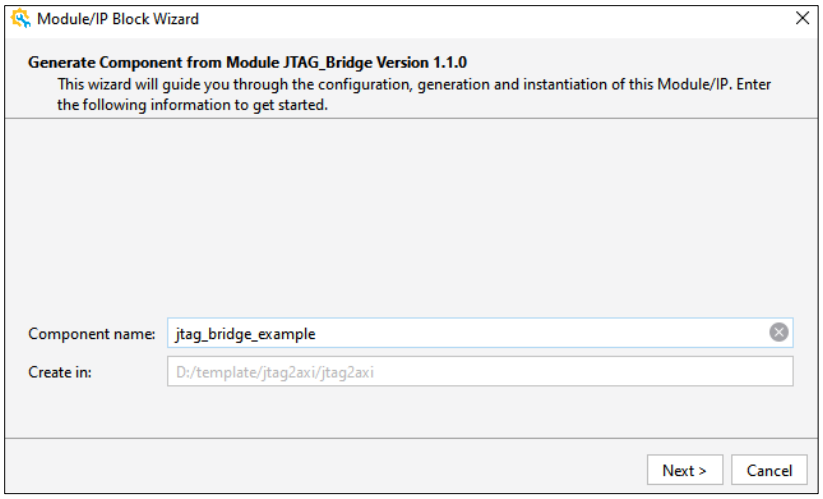

Figure 3.1. Entering Component Name

3. Configure the parameters as needed. Click **Generate** (Figure 3.2).

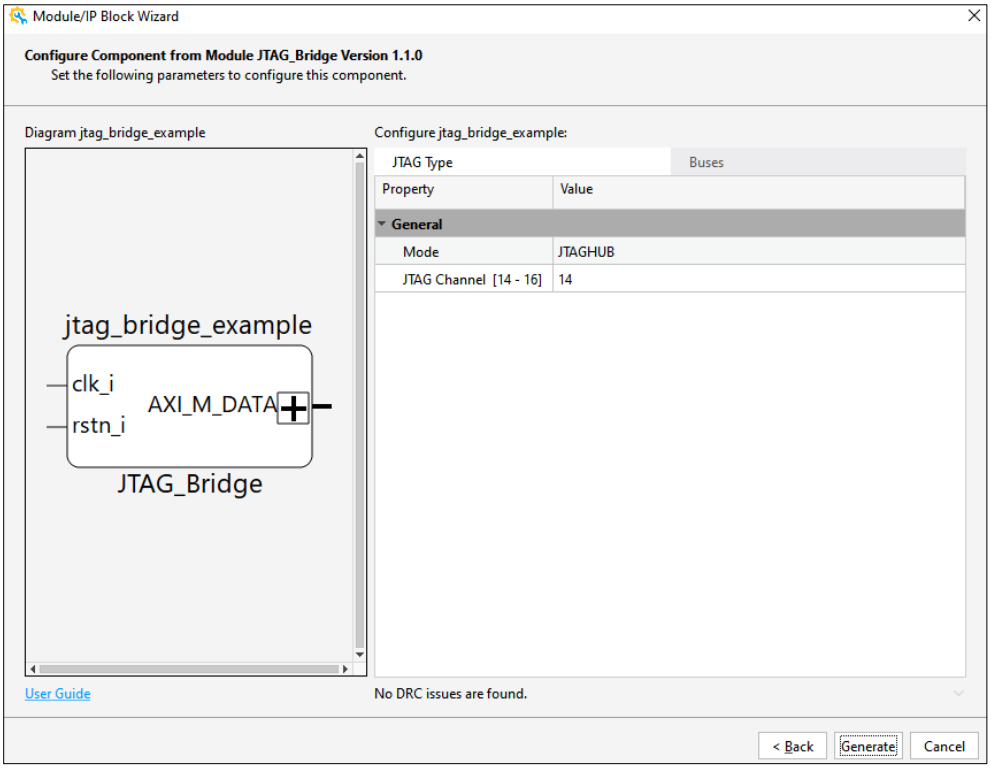

Figure 3.2. Configuring Parameters

4. Verify the information. Click **Finish** (Figure 3.3).

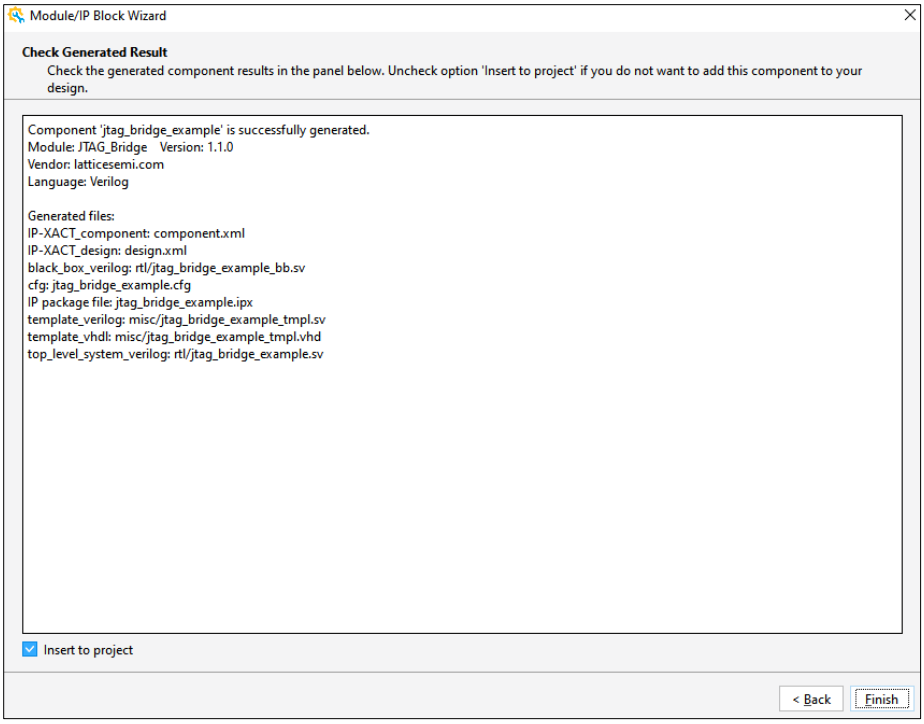

Figure 3.3. Verifying Results

5. Confirm or modify the module instance name. Click **OK** (Figure 3.4).

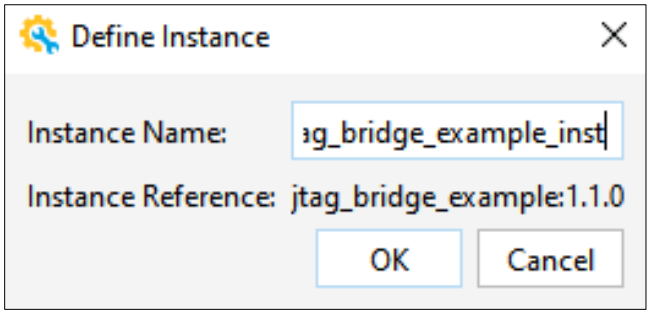

Figure 3.4. Specifying Instance Name

6. The JTAG Bridge IP instance is successfully generated, as shown in Figure 3.5.

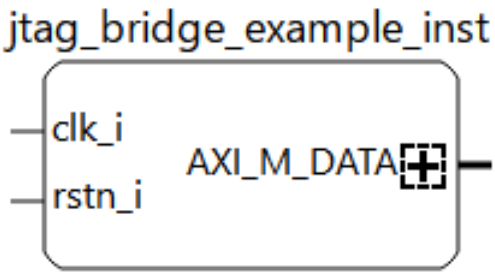

Figure 3.5. Generated Instance

4. JTAG Bridge Demo Template

In the 2026.1 release, the Lattice Propel Builder software provides a JTAG Bridge Demo template to demonstrate the functions of the JTAG Bridge IP. The template instantiates two JTAG Bridge IPs, and they work in separate modes. The demonstration includes how to enable the port, access a memory, and access peripheral registers.

4.1. Create a New SoC Design

1. Click **File** > **New SoC Design** to create a new design from a template (Figure 4.1).

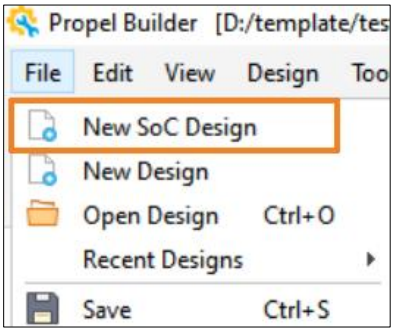

Figure 4.1. Create a New SoC Design

2. Name the SoC design and select the language and the file location (Figure 4.2).

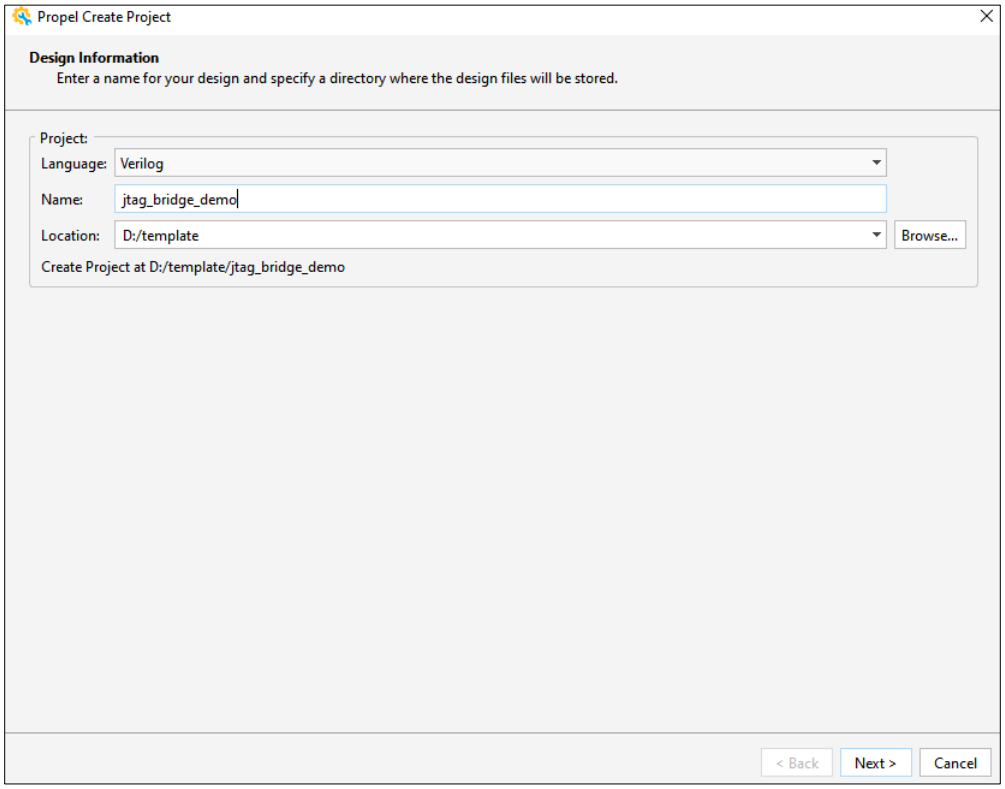

Figure 4.2. Name the Design and Select the File Path

3. Select **Scalable RISC-V SoC Project** (Figure 4.3).

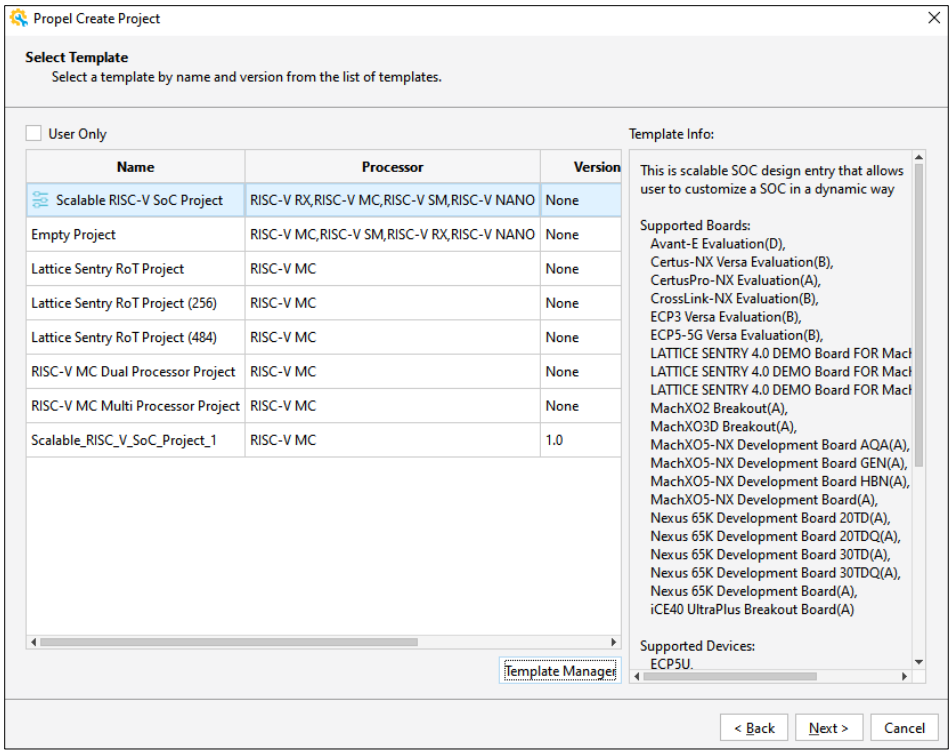

Figure 4.3. Select Scalable SoC Project

4. Select **JTAG Bridge Demo (Unified Interconnect)**, as shown in Figure 4.4.

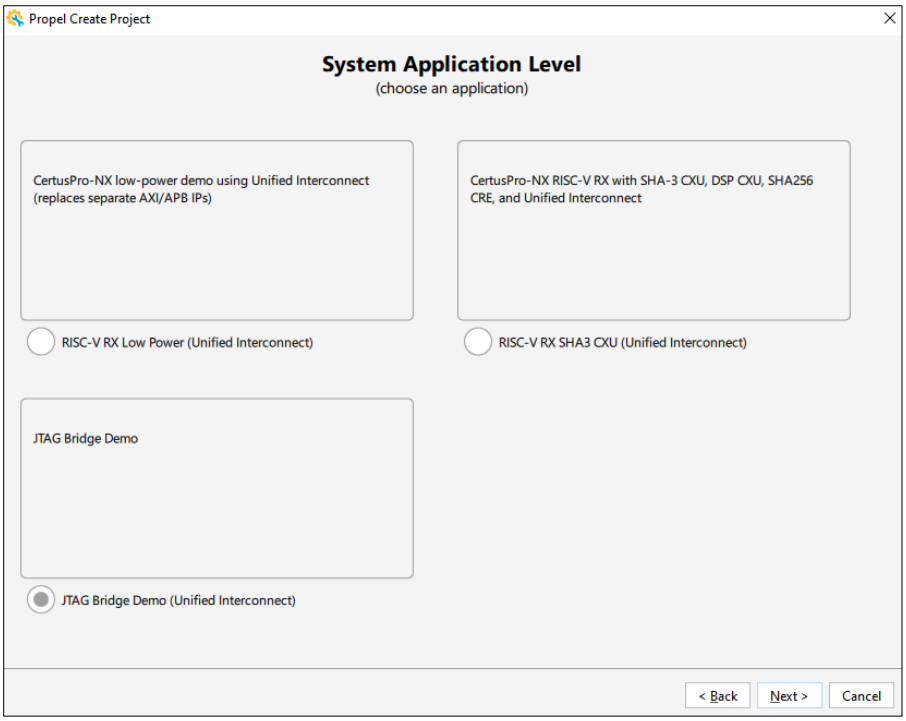

Figure 4.4. Select JTAG Bridge Demo (Unified Interconnect)

5. Select the target device or board. Device is for a plain SoC design while Board includes I/O constraints for the target board (Figure 4.5 and Figure 4.6).

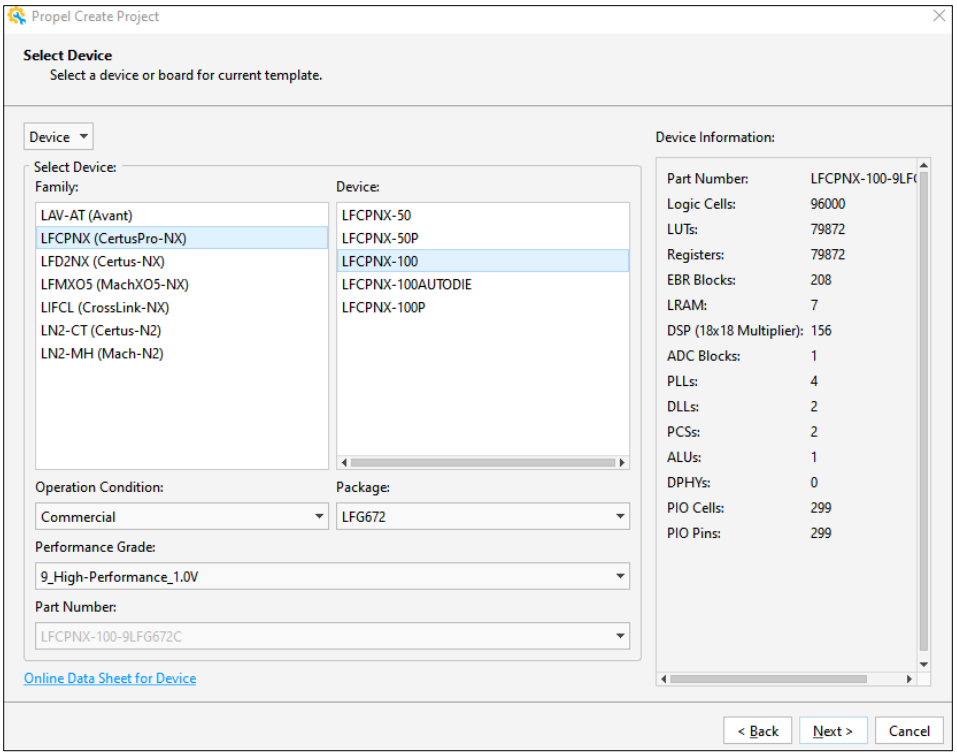

Figure 4.5. Select Device Settings

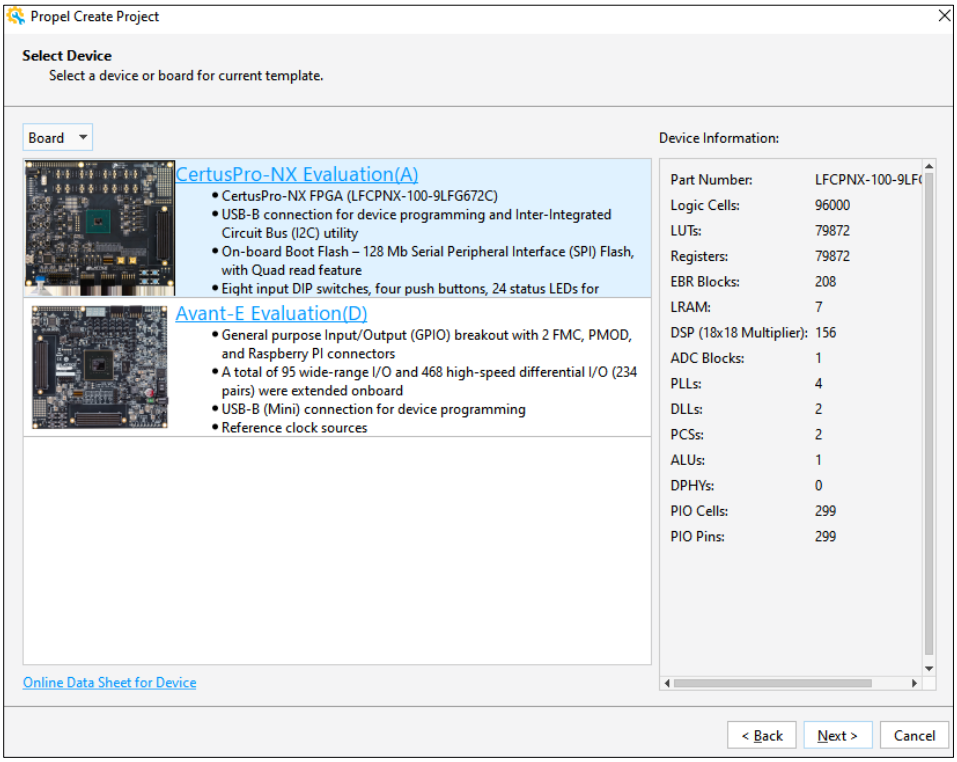

Figure 4.6. Select Board Settings

6. Install the required IPs from the server if they are not installed. Otherwise, click **Next** (Figure 4.7).

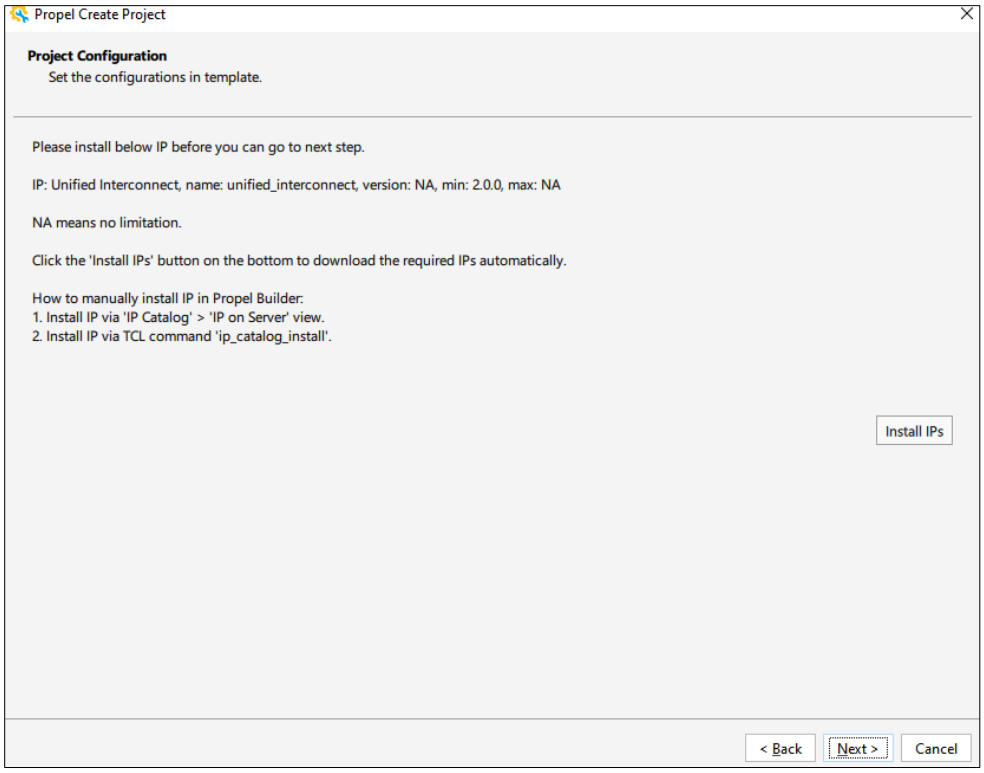

Figure 4.7. Install Required IPs if They Are Missing

7. Click **Finish**. The design is ready (Figure 4.8).

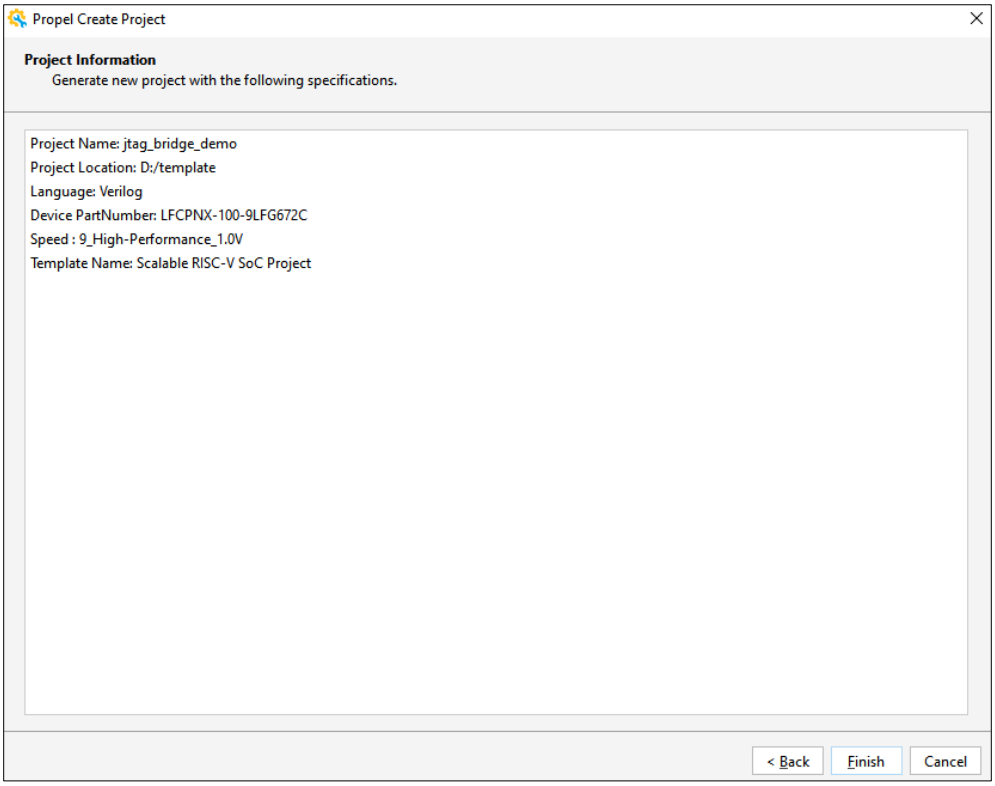

Figure 4.8. Finish the SoC Creation

8. The design includes two memory IPs with different interface types, one GPIO IP, and two JTAG Bridge IPs with different working modes (Figure 4.9).

   For the Standard mode:
   Use an external JTAG interface that must be connected and routed to device pins. This requires additional setup, such as pin assignment and external cable connection.

   For the JTAGHUB mode:
   Use an integrated JTAG path within the device. No external JTAG pin routing or additional setup is required, enabling faster bring-up.

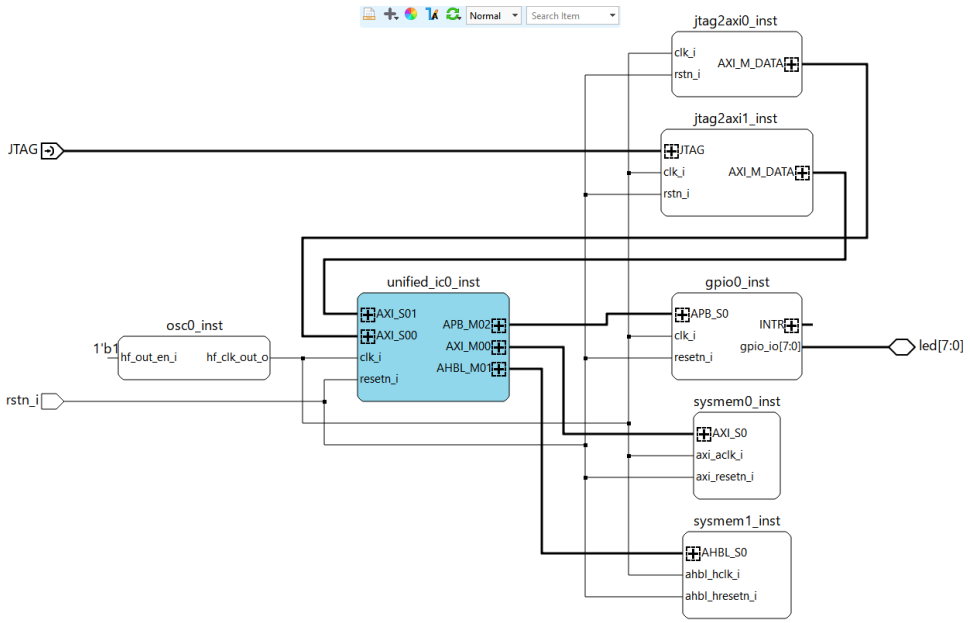

Figure 4.9. SoC Schematic

4.2. Generate a Bitstream

1. Click **Design > Run Radiant** or click the icon on the toolbar to launch the Lattice Radiant software (Figure 4.10).

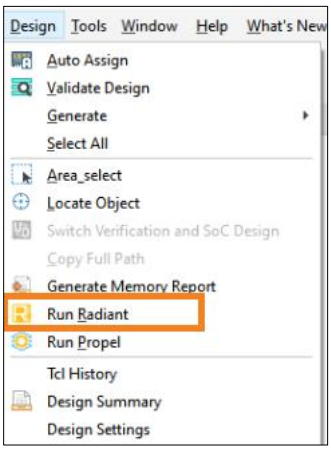

Figure 4.10. Launch the Lattice Radiant Software

2. If you select **Board** during SoC generation, the I/O constraints are automatically copied to the .pdc file (Figure 4.11). Otherwise, I/O constraints should be set based on the customer design.

For the Standard mode, the JTAG interface should be assigned in the .pdc file. Refer to Line 11 to Line 14 in Figure 4.11 for an example. For the JTAGHUB mode, no extra pin assignment is required. The Lattice Radiant software automatically assigns the JTAG signals to the dedicated pins.

Figure 4.11. Add I/O Constraints in the Lattice Radiant Software

3. If you use the Standard mode and the JTAG_TCK_port is not assigned to a pclk pin, add command-line options in the **Place & Route Design** strategies to pass the Lattice Radiant software flow (Figure 4.12). An example command line is shown below:

   ```
   -exp WARNING_ON_PCLKPLC1=1
   ```

   For details of pclk, refer to the Lattice Radiant software Help document.

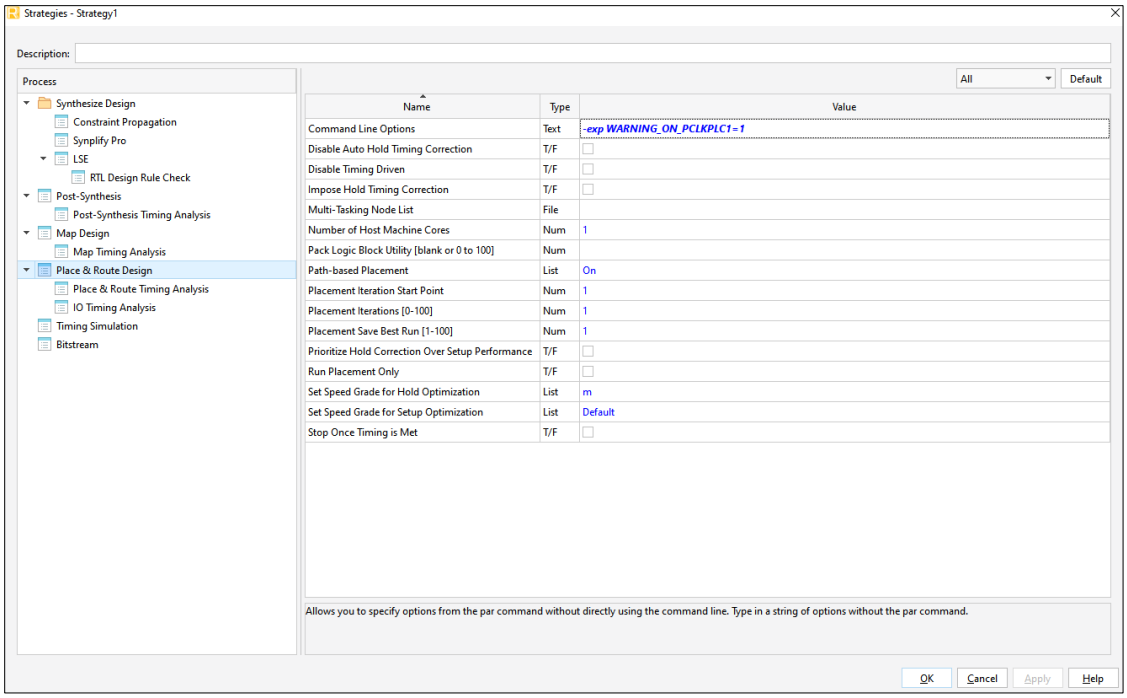

Figure 4.12. Add a Command Line in PAR Strategy

4. Run **Export Files** to generate a bitstream file.

4.3. Download the Bitstream to the User Board and Connect the Cables

Download the generated bitstream to the user board using Lattice Radiant Programmer.

To connect the JTAG Bridge IP in the JTAGHUB mode:

Connect a USB cable from the host computer to the FTDI USB port on the board.

To connect the JTAG Bridge IP in the Standard mode:

1. Connect a USB cable between the host computer and the HW-USBN-2B programmer cable (Figure 4.13).
2. Connect the JTAG signals, VCC, and GND from the HW-USBN-2B cable to the corresponding I/O pins on the target board.

Figure 4.13. HW-USBN-2B Cable

4.4. Start Lattice Propel Builder TCL Console

Install the latest Lattice Propel Builder 2026.1 software to use this feature. The TCL Console is by default at the bottom of the software GUI.

4.5. TCL Example

The following examples show how to use TCL to access a memory or drive a peripheral. All parameters are based on the template SoC design. Update the parameters to match your own design.

4.5.1. sbp_cable list

The sbp_cable list command identifies the type of USB cable and its port number. For example, Figure 4.14 shows Port 0 is the HW-USBN-2B cable and it is for the Standard mode. Port 1 and Port 2 are two channels of the normal USB cable. One of them is for the JTAGHUB mode, which is determined by the FTDI chip circuitry on the board.

```
% sbp_cable list
-------------------------------------------------------------------
 Port |    Name    |          Cable           | Channel | Location
-------------------------------------------------------------------
 0    | FTUSB-0    | Lattice HW-USBN-2B Ch A  | A       | 0
 1    | FTUSB-1    | Dual RS232-HS A          | A       | 1
 2    | FTUSB-2    | Dual RS232-HS B          | B       | 3
-------------------------------------------------------------------
```

Figure 4.14. sbp_cable list Example

4.5.2. sbp_cable open -port <num>

To enable the channel for the Standard mode, you should use this TCL command without the platform and channel number inputs, as shown in Figure 4.15.

```
% sbp_cable open -port 0
Open USB port 0 (Standard mode) successfully.
```

Figure 4.15. sbp_cable open Example for the Standard Mode

4.5.3. sbp_cable open -port <num> -platform nexus|avant -channel <num>

To enable the channel for the JTAGHUB mode, you should add the platform information of the chip and the channel number in the IP configuration, as shown in Figure 4.16.

```
% sbp_cable open -port 1 -platform nexus -channel 14
Open USB port 1 (JTAGHUB mode) successfully.
```

Figure 4.16. sbp_cable open Example for the JTAGHUB Mode

4.5.4. sbp_cable close

The sbp_cable close command disables the channel of the current connection (Figure 4.17).

```
% sbp_cable close
Done to close USB port.
```

Figure 4.17. sbp_cable close Example

4.5.5. sbp_read_8/16/32 -addr <address> [-len<len>]

The sbp_read_8/16/32 command reads a list of registers or memory locations of a specific length. The len value represents the number of bytes, halfwords, or words. The returned value is little-endian. Figure 4.18 shows how to read the GPIO direction register and the return value 0xFF indicates that the 8 bits, starting from LSB, are configured as outputs.

```
% sbp_read_32 -addr 0x00020010
0x000000FF
```

Figure 4.18. sbp_read_32 Example

4.5.6. sbp_write_8/16/32 -addr <address> -data <data>

The sbp_write_8/16/32 command writes data to a list of registers or memory locations of a specific length. The input is little-endian. There are no returns for this command. Figure 4.19 shows how to change the LED status by writing to the GPIO output. You can see the LED status changes on the board.

```
% sbp_write_32 -addr 0x00020004 -data 0x000000ff
```

Figure 4.19. sbp_write_32 Example

4.5.7. sbp_read_memory -addr <address> [-len <length>] [-format table|row]

The sbp_read_memory command reads a user-specified number of bytes from a memory. The output can be a table or a row. The output is big-endian and the length should be less than 4 KB. Figure 4.20 shows how to read eight bytes from sysmem0.

```
% sbp_read_memory -addr 0x00000000 -len 8 -format raw
0x47ab392c44afe098
% sbp_read_memory -addr 0x00000000 -len 8 -format table
-------------------------------------------------------------
Address     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-------------------------------------------------------------
0x00000000  47 AB 39 2C 44 AF E0 98
%
```

Figure 4.20. sbp_read_memory Example

4.5.8. sbp_write_memory -addr <address> -data <data> [-readback 1|0]

The sbp_write_memory command writes user-specified byte data to a memory. The input is big-endian and the length should be less than 4 KB. Figure 4.21 shows how to write to sysmem1 with readback. Since the write is consecutive, it should be used carefully when writing to peripheral registers.

```
% sbp_write_memory -addr 0x00010000 -data 0x1122334455667788
-------------------------------------------------------------
Address     00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-------------------------------------------------------------
0x00010000  11 22 33 44 55 66 77 88
```

Figure 4.21. sbp_write_memory Example

Appendix A. Resource Utilization

Table A.1. Resource Utilization in CertusPro-NX-100 Device

Configuration	LUTs	Registers	sysMEM EBRs
Address Width = 32, AXI ID Width = 4, MASK = 0, Mode = Standard	203	197	0
Address Width = 32, AXI ID Width = 4, MASK = 1, Mode = Standard	203	197	0
Address Width = 32, AXI ID Width = 4, MASK = 1, Mode = JTAGHUB	91	277	0

Table A.2. Resource Utilization in LAV-AT-E70 Device

Configuration	LUTs	Registers	sysMEM EBRs
Address Width = 32, AXI ID Width = 4, MASK = 0, Mode = Standard	286	194	0
Address Width = 32, AXI ID Width = 4, MASK = 1, Mode = Standard	277	206	0
Address Width = 32, AXI ID Width = 4, MASK = 1, Mode = JTAGHUB	95	295	0

References

- Lattice Propel Builder 2026.1 User Guide (FPGA-UG-02254)
- Lattice Memory Mapped Interface and Lattice Interrupt Interface User Guide (FPGA-UG-02039)
- AMBA AXI Protocol Specification

For more information, refer to:

- Lattice Propel Design Environment web page
- Lattice Certus-N2 Family Devices web page
- Lattice Avant-E Family Devices web page
- Lattice Avant-G Family Devices web page
- Lattice Avant-X Family Devices web page
- MachXO5-NX Family Devices web page
- CrossLink-NX Family Devices web page
- CertusPro-NX Family Devices web page
- Certus-NX Family Devices web page
- Lattice Insights for Training Series 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 1.0, IP v1.1.0, June 2026

Section	Change Summary
All	Production release.

www.latticesemi.com