



MachXO4 CRE Mux IP

IP Version: v1.0.0

User Guide

FPGA-IPUG-02337-1.0

September 2026

Disclaimers

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

Inclusive Language

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

Contents

Contents	3
Abbreviations in This Document.....	5
1. Introduction	6
1.1. Overview of the IP	6
1.2. Quick Facts	6
1.3. IP Support Summary	6
1.4. Features	7
1.5. Licensing and Ordering Information	7
1.5.1. Ordering Part Number.....	7
1.6. Naming Conventions	7
1.6.1. Nomenclature.....	7
1.6.2. Signal Names	7
2. Functional Description.....	8
2.1. IP Architecture Overview	8
2.2. Clocking	9
2.2.1. Clocking Overview	9
2.3. Reset	10
2.3.1. Reset Overview	10
2.4. User Interfaces	10
3. Signal Description	11
3.1. Clock Interface	11
4. Register Description	12
5. Designing with the IP	13
5.1. Generating and Instantiating the IP	13
5.1.1. Generated Files and File Structure	14
5.2. Design Implementation	15
5.3. Timing Constraints	15
5.4. Specifying the Strategy.....	15
5.5. Running Functional Simulation	16
5.5.1. Simulation Results	18
Appendix A. Resource Utilization	20
References.....	21
Technical Support Assistance	22
Revision History.....	23

Figures

Figure 2.1. CRE Mux Block Diagram.....	8
Figure 2.2. CRE Mux Block Diagram.....	8
Figure 2.3. Clock Design for Using CRE Mux	9
Figure 2.4. CRE Clock Domain Diagram	9
Figure 6.1. Module/IP Block Wizard	13
Figure 6.2. IP Configuration	14
Figure 6.3. Check Generated Result	14
Figure 6.4. Timing Constraint File (.pdc) for the CRE Mux IP	15
Figure 6.5. Simulation Wizard.....	16
Figure 6.6. Add and Reorder Source	17
Figure 6.7. Simulation Waveform	17
Figure 6.8. ECDSA Verification Waveform 1	18
Figure 6.9. ECDSA Generation Waveform 2	18

Figure 6.10. Simulation log	19
Figure 6.11. ECDSA Generation Algorithm	19

Tables

Table 1.1. Summary of the CRE Mux IP	6
Table 1.2. CRE Mux IP Support Readiness	6
Table 1.3. Ordering Part Number	7
Table 2.1. User Interfaces and Supported Protocols	10
Table 3.1. CRE Mux Signal Description	11
Table 3.1 Clock Ports	11
Table 4.1. Control Register	12
Table 7.1. Generated File List	15
Table A.1. Resource Utilization	20

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
AES	Advanced Encryption Standard
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
API	Application Programming Interface
ASF	Asynchronous FIFO
CPU	Central Processing Unit
CRE	Cryptographic Engine
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
FIFO	First-In, First-Out
FPGA	Field Programmable Gate Array
HMAC	Hash-based Message Authentication Code
HSP	High-Speed Data Port
LUT	Look-Up Table
PFR	Platform Firmware Resiliency
PLL	Phase-locked Loop
RISC-V	Reduced Instruction Set Computer-V (five)
RoT	Root of Trust
RTL	Register Transfer Level
SHA	Secure Hash Algorithm

1. Introduction

This document describes how to select between the two available logical interfaces of the MachXO4 RoT™ Cryptographic Engine (CRE). The CRE has two logical interfaces for sending and receiving data: a WISHBONE register interface and a High-Speed Data Port (HSP) FIFO-style interface. These two logical interfaces share one physical data interface with the CRE (one shared 32-bit input port and one shared 32-bit output data port). The WISHBONE interface of the CRE is converted to an Advanced Peripheral Bus (APB) interface for RISC-V CPU access.

1.1. Overview of the IP

The design is implemented in Verilog HDL. It can be configured and generated using Lattice Propel™ Builder. It can be targeted to MachXO4 RoT FPGA devices and implemented using the Lattice Radiant™ software Place and Route tool integrated with the Synplify Pro® synthesis tool.

1.2. Quick Facts

Table 1.1. Summary of the CRE Mux IP

IP Requirements	Supported Devices	MachXO4 RoT (LFMXO4D)
	IP Changes ¹	For a list of changes to the IP, refer to the MachXO4 CRE Mux Module IP Release Notes (FPGA-RN-02120) .
Resource Utilization	Supported User Interface	APB
	Resources	Refer to Appendix A. Resource Utilization .
Design Tool Support	Lattice Implementation	IP Core v1.0.0 – Lattice Radiant software 2026.1.1 or later
	Synthesis	Synopsys® Synplify Pro for Lattice, Lattice Synthesis Engine
	Simulation	Lattice Synthesis Engine, Synopsys® Synplify Pro for Lattice.
Driver Support	API Reference	QuestaSim

Notes:

1. In some instances, the IP may be updated without changes to the user guide. This user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.
2. Lattice Implementation indicates the IP version release coinciding with the software version release. Check the software for IP version compatibility with earlier or later software versions.

1.3. IP Support Summary

Table 1.2. CRE Mux IP Support Readiness

Device Family	Device	IP	Configuration	Radiant Timing Model	Hardware Validated
MachXO4	LFMXO4D-040HC LFMXO4D-040ZC LFMXO4D-050HC LFMXO4D-050ZC LFMXO4D-080HC LFMXO4D-080HE LFMXO4D-080ZC LFMXO4D-110HC LFMXO4D-110HE LFMXO4D-110ZC	CRE Mux	Default	Preliminary	No

The example design for the CRE Mux IP allows simulation and deployment to development boards for testing.

1.4. Features

Key features of the CRE Mux IP include:

- Soft wrapper around the CRE to provide separate APB and high-speed port interfaces to user logic.
- Support for AMBA 3 APB Protocol v1.0.

1.5. Licensing and Ordering Information

An IP specific license string is required to enable full use of the CRE Mux IP in a complete, top-level design.

The IP can be fully evaluated through functional simulation and implementation (synthesis, map, place and route) without an IP license string. This IP supports Lattice's IP hardware evaluation capabilities. You can create versions of the IP to operate in hardware for a limited time (approximately four hours) without requiring an IP license string. A license string is required to enable timing simulation and to generate a bitstream file that does not include the hardware evaluation timeout limitation.

For more information about pricing and availability of the CRE Mux IP, contact your [local Lattice Sales Office](#).

1.5.1. Ordering Part Number

Table 1.3. Ordering Part Number

Device Family	Part Number	
	Single Seat Annual	Single Seat Perpetual
MachXO4	—	—

1.6. Naming Conventions

1.6.1. Nomenclature

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

1.6.2. 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. Functional Description

2.1. IP Architecture Overview

The CRE Mux is a thin wrapper around the CRE, which converts the WISHBONE interface of the CRE to an APB interface. It also provides separate interface ports for the APB and the HSP plus an internal Mux to select between the two. The Mux is controlled by a control register, which is mapped into unused CRE address space. This register is always available through the APB interface, regardless of whether Mux is set to APB or HSP. When the Mux is set to APB, the external read FIFO status is forced to empty (ASFEMPTYO=1) and FIFO read requests (ASFRDI) are ignored to prevent external logic from initiating FIFO reads, which could cause data conflicts with APB.

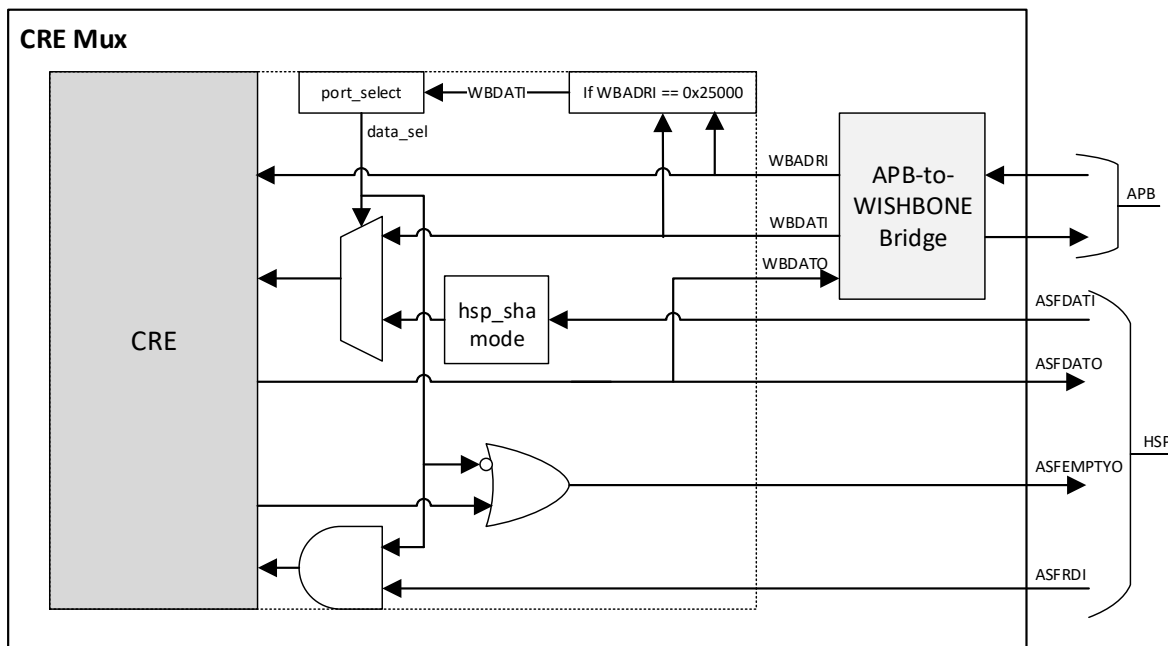


Figure 2.1. CRE Mux Block Diagram

Figure 2.2 shows the block diagram for a typical application.

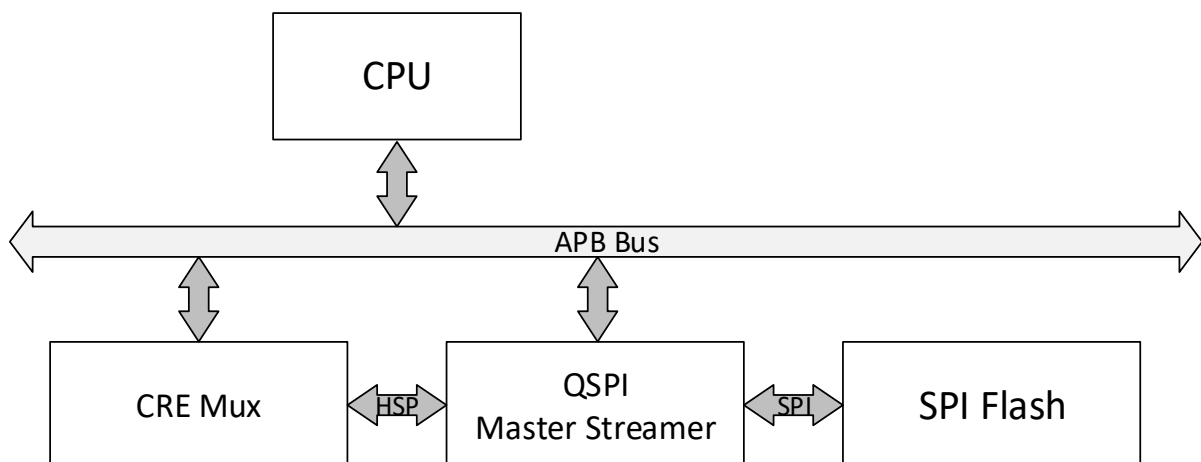


Figure 2.2. CRE Mux Typical Application

2.2. Clocking

The OSCJ primitive must be instantiated when instantiating the CRE Mux. Connect the OSCJ *OSCESB* output to the CREB primitive clock input *OSCCLK*, as shown in Figure 2.3. The *OSCESB* output is fixed at 66.5 MHz. Other functions, inputs, and outputs of the OSCJ primitive such as *OSC* and *NOM_FREQ*, are independent of *OSCESB* function and thus can still be used for arbitrary user-purposes.

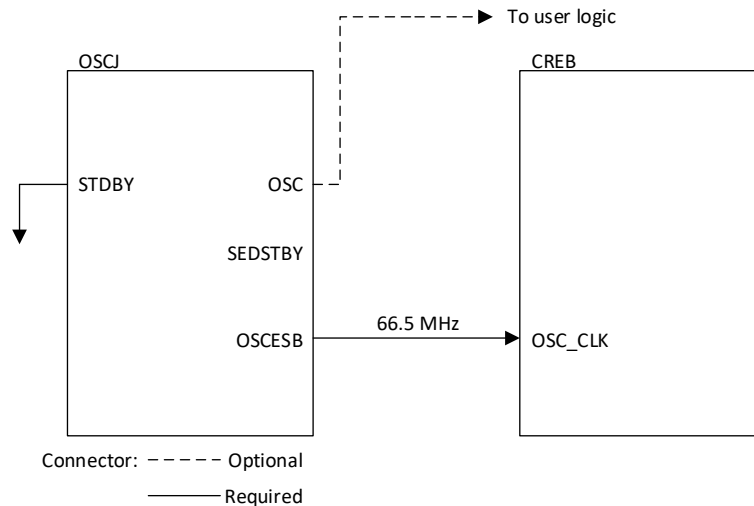


Figure 2.3. Clock Design for Using CRE Mux

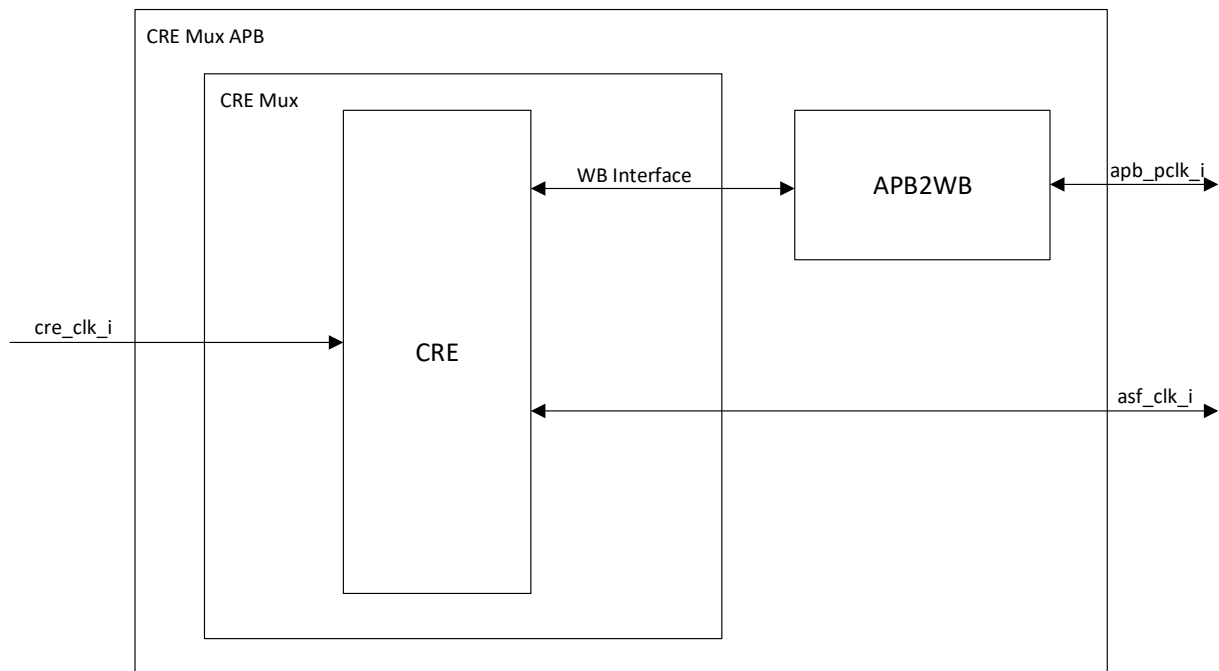


Figure 2.4. CRE Clock Domain Diagram

2.2.1. Clocking Overview

The MachXO4 RoT CRE Mux IP has multiple clock domains, which are as follows.

- *apb_pclk_i* – System clock.
- *cre_clk_i* – CRE clock.
- *asf_clk_i* – Asynchronous FIFO clock.

2.3. Reset

The CRE Mux core logic is held in reset automatically by the FPGA configuration logic during configuration and asserted during power-down wake-up operations.

2.3.1. Reset Overview

The MachXO4 RoT CRE Mux IP uses a single reset domain.

- `reset_n_i` – System asynchronous reset (active low).

2.4. User Interfaces

Table 2.1. User Interfaces and Supported Protocols

User Interface	Supported Protocols	Description
Control	APB	Used for transacting in the CRE core.
Control	HSP	Used for transacting in the CRE core.

3. Signal Description

The CRE Mux includes all the interface ports of the CREB primitive, with the addition of asf_dat_i[31:0] and asf_dat_o[31:0].

Table 3.1. CRE Mux Signal Description

Port	Width	Direction	Description
System			
reset_n_i	1	In	System Asynchronous Reset
apb_pclk_i	1	In	APB clock
asf_clk_i	1	In	HSP clock
cre_clk_i	1	In	CRE clock; Oscillator input.
APB Interface			
apb_psel_i	1	In	Select signal, indicates that the target device is selected and a data transfer is required.
apb_paddr_i	32	In	Address signal
apb_pwdata_i	32	In	Write data signal
apb_pwrite_i	1	In	Direction signal, Write = 1, Read = 0
apb_penable_i	1	In	Enable signal, Indicates the second and subsequent cycles of an APB transfer
apb_pready_o	1	Out	Ready signal, Indicates transfer completion. Slave uses this signal to extend an APB transfer
apb_prdata_o	32	Out	Read data signal
HSP Interface			
asf_dat_i	32	In	HSP Write Data: 32-bit wide input data to write into the CRE FIFO
asf_dat_o	32	Out	HSP Read Data: 32-bit wide output data read from the CRE FIFO
asf_full_o	1	Out	FIFO full status
asf_empty_o	1	Out	FIFO empty status
asf_wr_i	1	In	Write enable
asf_rd_i	1	In	Read enable

3.1. Clock Interface

Table 3.2 Clock Ports

Port	Type	Description
apb_pclk_i	Input	This signal is the APB Interface Domain Input Clock
asf_clk_i	Input	This signal is the HSP Interface Domain Input Clock
cre_clk_i	Input	This signal is the CRE Domain Input Clock

4. Register Description

CRE registers are mapped to offsets 0x00000-0x24FFF, and the CRE Mux port_select register is mapped to offset 0x25000.

Table 4.1. Control Register

Offset	Name	Access	Default Value	Description
0x00000–0x24FFF	CRE registers	—	—	Refer to the <i>MachXO4 RoT CRE Module IP User Guide (FPGA-IPUG-02336)</i> for more information. To obtain a copy of this document, create a new Technical Support Request through www.latticesemi.com/techsupport and indicate the following: - Case Type: Documentation - Case Category: App Note/Tech Note - Product Family: MachXO4
0x25000	port_select	RW	0	data_sel[0]: 0 – APB bus 1 – HSP hsp_sha_mode[1]: 0 – Data on ASFDAT1 is for non-SHA operations. 1 – Data on ASFDAT1 is for SHA operations.

5. Designing with the IP

This section provides information on how to generate the IP core using the Lattice Radiant software and how to run simulation and synthesis. For more details on the Lattice Radiant software, refer to the Lattice Radiant software user guide.

Note: The screenshots provided are for reference only. Details may vary depending on the version of the IP or software being used. If there have been no significant changes to the user interface, a screenshot may reflect an earlier version of the IP.

5.1. Generating and Instantiating the IP

You can use the Lattice Radiant software to generate IP modules and integrate them into the device architecture. The steps below describe how to generate the CRE Mux IP in the Lattice Radiant software.

To generate the CRE Mux IP:

1. Create a new Lattice Radiant software project or open an existing project.
2. Click the **IP Catalog** button to view the **IP Catalog** pane.
3. On the IP on Local tab, double-click CRE Mux under the IP/.../Processors_Controllers_and_Peripherals category. The Module/IP Block Wizard opens.

Note: If the IP is not available on the **IP on Local** tab, download the IP from the **IP on Server** tab.

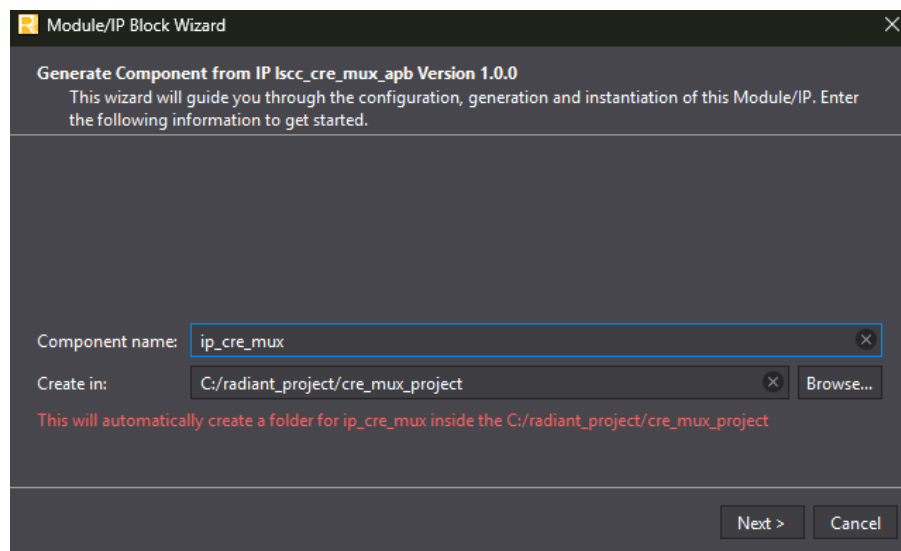


Figure 5.1. Module/IP Block Wizard

4. Enter values in the **Component name** and **Create in** fields, then click **Next**.
5. Customize the selected <IP NAME> IP using drop-down lists and check boxes. Figure 5.2 shows an example configuration of the <IP NAME> IP.

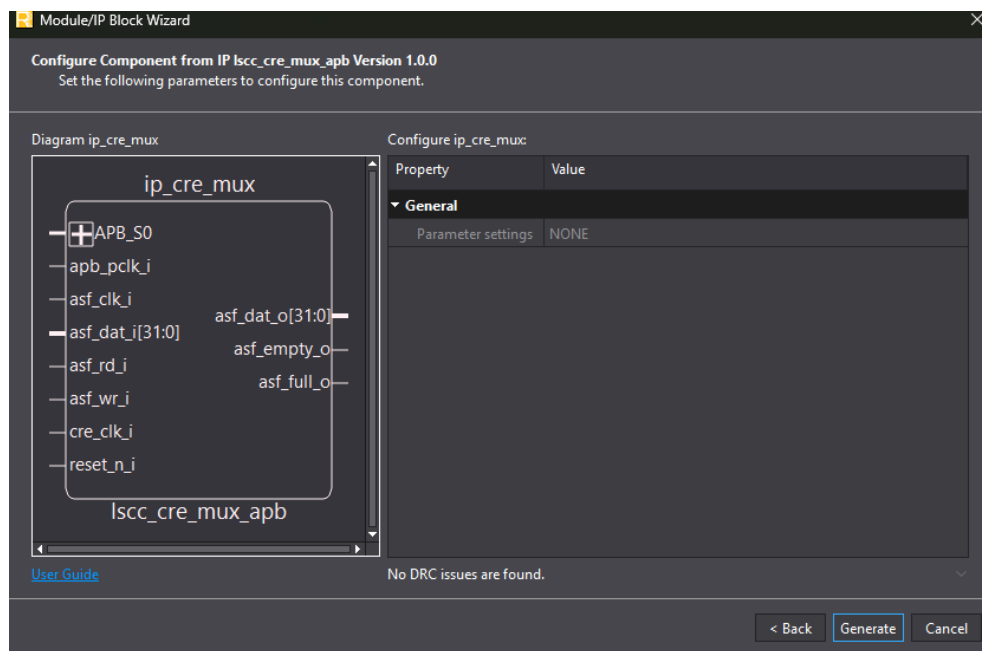


Figure 5.2. IP Configuration

- Click **Generate**. The **Check Generated Result** window opens. This window shows design block messages and results.

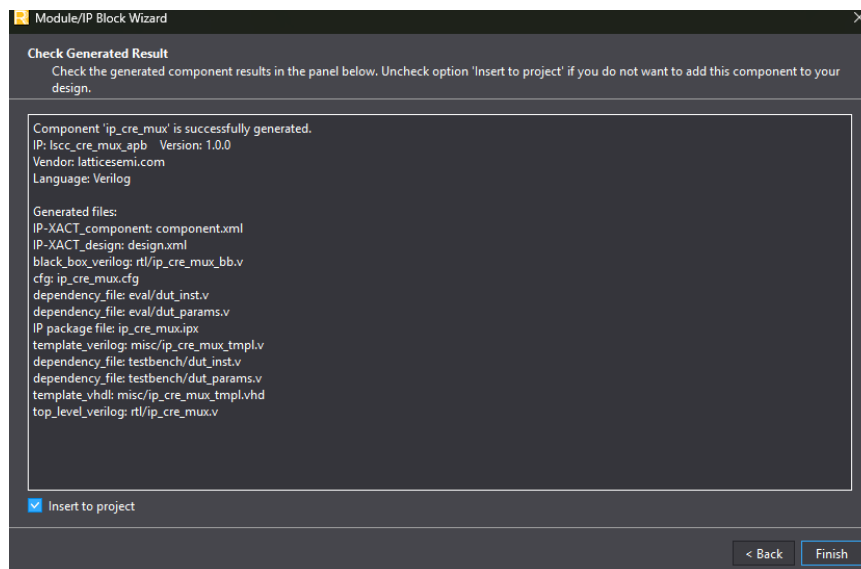


Figure 5.3. Check Generated Result

- Click **Finish**. All generated files are placed in the directory specified by the **Component name** and **Create in** fields shown in Figure 5.1.

5.1.1. Generated Files and File Structure

The generated CRE Mux module package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the module is also provided. You may also use this example as the starting template for your top-level design. The generated files are listed in Table 5.1.

Table 5.1. Generated 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 IP configuration.
component.xml	Contains the ipxact: component information of the IP.
design.xml	Documents the configuration parameters of the IP in the 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 misc/<Component name>_tmpl.vhd	Provide instance templates for the module.

5.2. Design Implementation

Completing your design includes additional steps to specify analog properties, pin assignments, and timing and physical constraints. You can add and edit the constraints using the Device Constraint Editor or by manually creating a .pdc file.

Post-Synthesis constraint files (.pdc) contain both timing and non-timing *constraint.pdc* source files for storing logical timing/physical constraints. Constraints that are added using the Device Constraint Editor are saved to the active .pdc file. The active post-synthesis design constraint file is then used as input for post-synthesis processes.

Refer to the relevant sections in the Lattice Radiant Software User Guide for more information on how to create or edit constraints and how to use the Device Constraint Editor.

5.3. Timing Constraints

The timing constraints are based on the clock frequency used. The timing constraints for the IP are defined in relevant constraint files. The example below shows the IP timing constraints generated for the CRE Mux IP.

```
create_clock -name {apb_pclk_i} -period 25 {get_ports apb_pclk_i}
create_clock -name {asf_clk_i} -period 10 {get_ports asf_clk_i}

set_clock_groups -asynchronous \
    -group {apb_pclk_i} \
    -group {asf_clk_i}
```

Figure 5.4. Timing Constraint File (.pdc) for the CRE Mux IP

For timing closure, add the following timing constraints in the .pdc file. Generate PLL with output clocks as shown below, and define proper clock constraints as follows:

- Create apb_pclk_i = 40 MHz and asf_clk_i = 100 MHz.
- Define input clk of PLL with clk name & period with “create_clock -name ...”.
- Define the asynchronous relationship between apb_pclk_i and asf_clk_i.

For more information on timing constraints, refer to the [Lattice Radiant Timing Constraints Methodology \(FPGA-AN-02059\)](#).

5.4. Specifying the Strategy

The Radiant software provides two predefined strategies: Area and Timing. The software also enables you to create customized strategies. For details on how to create a new strategy, refer to the *Strategies* section of the Lattice Radiant software user guide.

5.5. Running Functional Simulation

You can run functional simulation after the IP is generated.

To run functional simulation:

1. Click the  button located on the **Toolbar** to initiate the **Simulation Wizard** shown in [Figure 5.5](#).

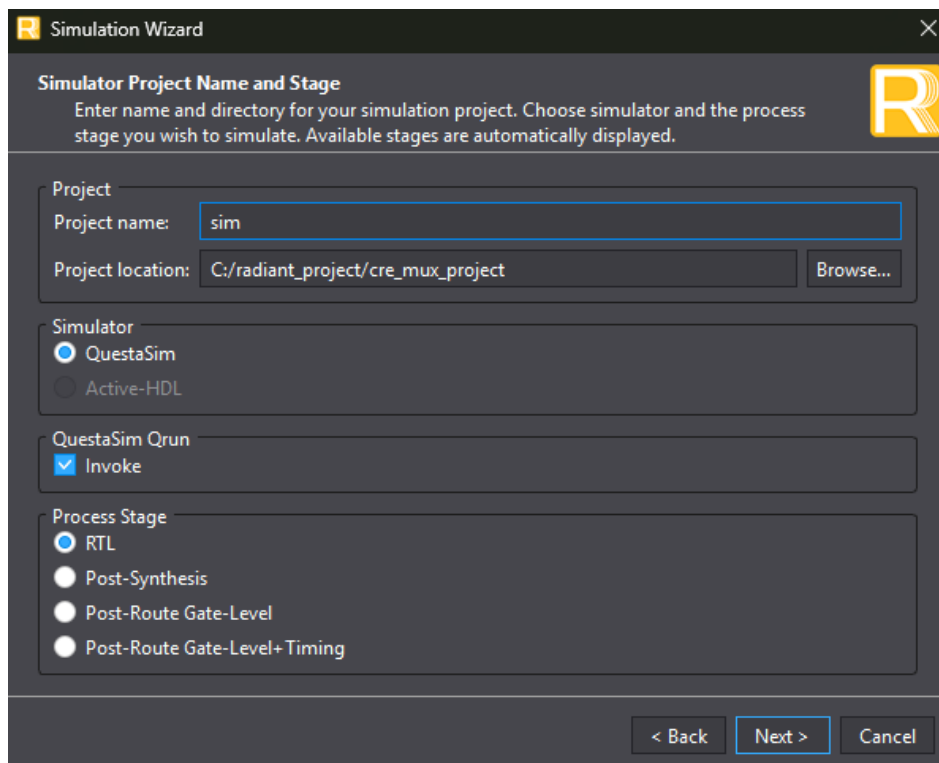


Figure 5.5. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in [Figure 5.6](#).

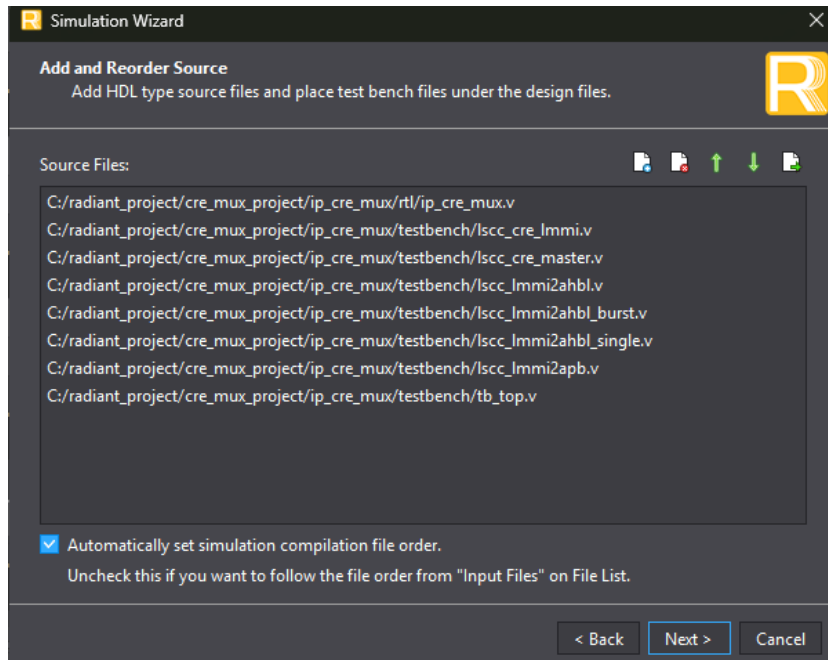


Figure 5.6. Add and Reorder Source

3. Click **Next**. The **Summary** window is shown.
4. Click **Finish** to run the simulation.

The waveform in Figure 5.7 shows an example simulation result.

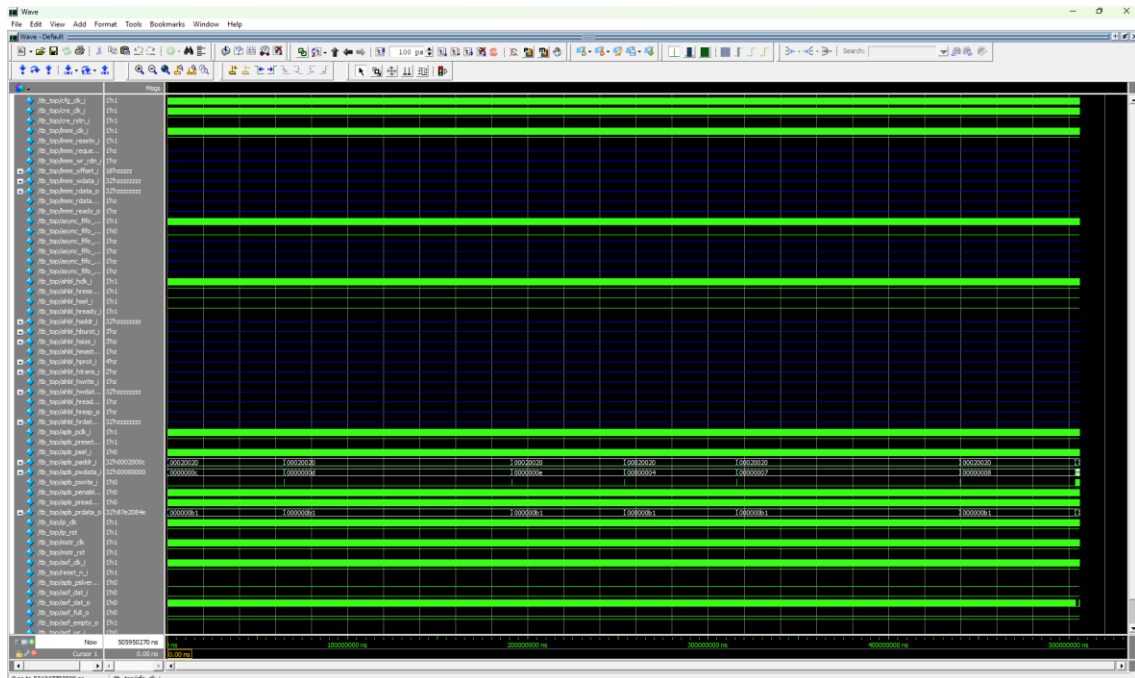


Figure 5.7. Simulation Waveform

[illegible]

Figure 5.11 shows the sample flow diagram of ECDSA Generation. For other CRE APIs, refer to the CRE specifications.



Appendix A. Resource Utilization

Table A.1 shows a sample resource utilization of the CRE Mux IP core on the LFMXO4D-110ZC-2BBG256C device.

Table A.1. Resource Utilization

IP Configuration	Slices	LUTs	Registers	EBR
Default	90/9400	201/9400	90/10015	0/48

References

- [Lattice Radiant Timing Constraints Methodology \(FPGA-AN-02059\)](#)
- [MachXO4 CRE Mux Module IP Release Notes \(FPGA-RN-02120\)](#)
- [MachXO4](#) web page
- [Lattice Radiant Software](#) web page
- [Lattice Propel Design Environment](#) web page
- [Lattice Solutions IP Cores](#) web page
- [Lattice Solutions Reference Designs](#) 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

Note: In some instances, the IP may be updated without changes to the user guide. The user guide may reflect an earlier IP version but remains fully compatible with the later IP version. Refer to the IP Release Notes for the latest updates.

Revision 1.0, October 2026

Section	Change Summary
All	Initial release.



www.latticesemi.com