LVDS Tunneling Protocol and Interface (LTPI) User Guide

Reference Design

FPGA-RD-02247-1.6

April 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.

Abbreviations	Definition
AC	Alternating Current
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
CDC	Clock Domain Crossing
CRC	Cycle Redundancy Check
CSR	Control and Status Register
DC-SCI	Datacenter-ready Secure Control Interface
DC-SCM	Datacenter-ready Secure Control Module
DDR	Double Data Rate
EFB	Embedded Function Block
FPGA	Field Programmable Gate Array
GDDR	Graphics Double Data Rate
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HDL	Hardware Description Language
HPM	Host Processor Module
I/O	Input/Output
I2C	Inter-Integrated Circuit
IP	Intellectual Property
IPK	Intellectual Property Kit
LED	Light-Emitting Diode
LLGPIO	Low Latency General Purpose Input/Output
LTPI	LVDS Tunneling Protocol and Interface
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MCSI	Multi-Channel Serial Interface
MCTP	Management Component Transport Protocol
N/A	Not Applicable
NLGPIO	Normal Latency General Purpose Input/Output
OCP	Open Compute Project
OEM	Original Equipment Manufacturer
PFU	Programmable Functional Unit
PLL	Phase-Locked Loop
RTL	Register Transfer Level
Rx	Receiver
SCM	Secure Control Module
SDR	Single Data Rate
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

1. Introduction

The DC-SCM 2.0 LTPI Reference Design offers multiple solution templates utilizing the current LVDS Tunneling Protocol Interface (LTPI) IP. This IP complies with Datacenter-ready Secure Control Module (DC-SCM) 2.2 LTPI 1.2 specifications and features a standardized Datacenter-ready Secure Control Interface (DC-SCI). It aggregates multiple data channels, including I2C, GPIO, and UART to enhance flexibility in a customer's system and board design.

This reference design is OCP Ready™.

1.1. Quick Facts

You can download the reference design files in the LVDS Tunneling Protocol and Interface Reference Design web page.

Table 1.1. Summary of the Reference Design

General	Supported Devices	MachXO3™, MachXO3D™, and MachXO5-NX™
	Source code format	Verilog
Simulation	Functional simulation	Performed
	Timing simulation	Not performed
	Test bench	Available
	Test bench format	Verilog
Software Requirements	Software tool and version	Lattice Diamond™ version 3.14 for the MachXO3, MachXO3D devices Lattice Radiant™ version 2025.2 for the MachXO5-NX devices Lattice Propel™ version 2025.2 Synplify Pro® Questa Lattice OEM Edition-64 2024.2
	IP version (if applicable)	DC-SCM LTPI 3.0.0 (via Propel Builder IP Catalog) I2C-to-APB Bridge Reference Design 1.3.0 (via Propel Builder IP Catalog) APB Streamer 1.0.0 (via IPK folder)
Hardware Requirements	Board	N/A
	Cable	N/A

1.2. Features

Key features of the LTPI reference design include:

- Support for DC-SCM 2.2 LTPI version 1.2 (Oct 2025)
- Link initialization, discovery, and negotiation
- Support for multi-channel serial interface
- Support for LVDS
- Aggregation/disaggregation of up to five channels in total
- Support for GPIO, I2C, UART, OEM, and data channel aggregation
- Configurable I2C interface as single-node (controller or target only) or multi-node (supports both external controller and target)
- Each multi-node I2C bus uses 1 payload byte, equivalent to two single-node I2C buses
- Support for up to 1,200 Mbps LVDS data rate (up to 800 Mbps on the MachXO3 and MachXO3D devices; up to 1,200 Mbps on the MachXO5-NX devices)
- Support for AMBA 3 APB Protocol version 1.0 for register access of the soft IP and data channel:
  - PREADY signal indicates completion of an APB transfer
  - PSLVERR signal indicates failure of a transfer, supported only in data channel-related access

1.3. Naming Conventions

1.3.1. Nomenclature

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

1.3.2. Signal Names

Signal names that have:

- *_n* are active low (asserted when value is logic 0)
- *_i* are input signals
- *_o* are output signals
- *_0* are Node 0 (default node) signals
- *_1 are Node 1 signals (for Dual-Node implementation only)*

2. Directory Structure and Files

Figure 2.1 shows the directory structure.

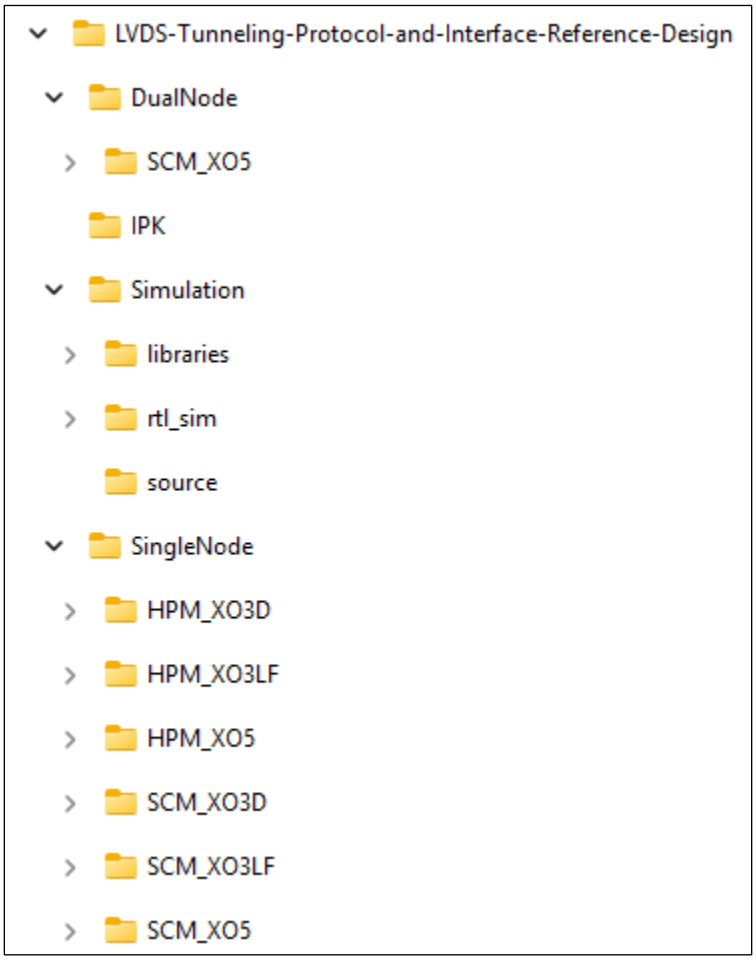

Figure 2.1. Directory Structure

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

Table 2.1. File List

Subfolder	Description
IPK	This subfolder contains the IPK used in the project. • latticesemi.com_ltpi_pll_cfg_inside_1.0.0.ipk • latticesemi.com_apb_streamer_1.0.0.ipk
DualNode	This subfolder contains the pre-built dual-node project using MachXO5-NX. • SCM_XO5 This project has been tested through simulation.
Simulation	This subfolder contains simulation files used for functional simulation. • Libraries: pre-built simulation libraries for the Lattice FPGA devices • Source: contains the test bench RTL and other submodules needed for simulation Additionally, this subfolder *.do files used to run preset functional simulations. These files are named in this format: *scm_<device_used_for_SCM>_hpm_<device_used_for_HPM>.do*

Subfolder	Description
	• dual_scm_xo5_hpm_xo3d.do • dual_scm_xo5_hpm_xo3lf.do • dual_scm_xo5_hpm_xo5.do • scm_xo3d_hpm_xo3d.do • scm_xo3d_hpm_xo3lf.do • scm_xo3d_hpm_xo5.do • scm_xo3lf_hpm_xo3d.do • scm_xo5_hpm_xo3lf.do • scm_xo3lf_hpm_xo5.do • scm_xo5_hpm_xo3d.do • scm_xo5_hpm_xo3lf.do • scm_xo5_hpm_xo5.do • wave.do – this is invoked by the other non-dual-node *.do files to add preset signals on the Wave view of the simulator • wave_dual.do – this is invoked by the other dual-node *.do files to add preset signals on the Wave view of the simulator
SingleNode	This subfolder contains pre-built single-node projects, subcategorized by the function and the Lattice FPGA used. • HPM_XO3D • HPM_XO3LF • HPM_XO5 • SCM_XO3D • SCM_XO3LF • SCM_XO5 Each project has been tested through simulation and on hardware.

3. Functional Description

The reference design provides you with a ready-to-use LTPI template for both SCM and HPM. The design manages all initialization requirements for SCM and HPM to establish a working link, following the LTPI state flow internally, as shown in Figure 3.1. It also provides access to the LTPI IP's Control and Status Registers through the I2C interface.

Figure 3.2 and Figure 3.3 illustrate the implementation of the DC-SCM LTPI IP within the reference design. Some modules differ between HPM and SCM implementations based on their intended usage. Additionally, because the design supports multiple target devices, some modules vary for each device. These differences will be discussed in the subsection of each block.

The reset from the Reset Timer is also used as the primary reset net unless specified otherwise.

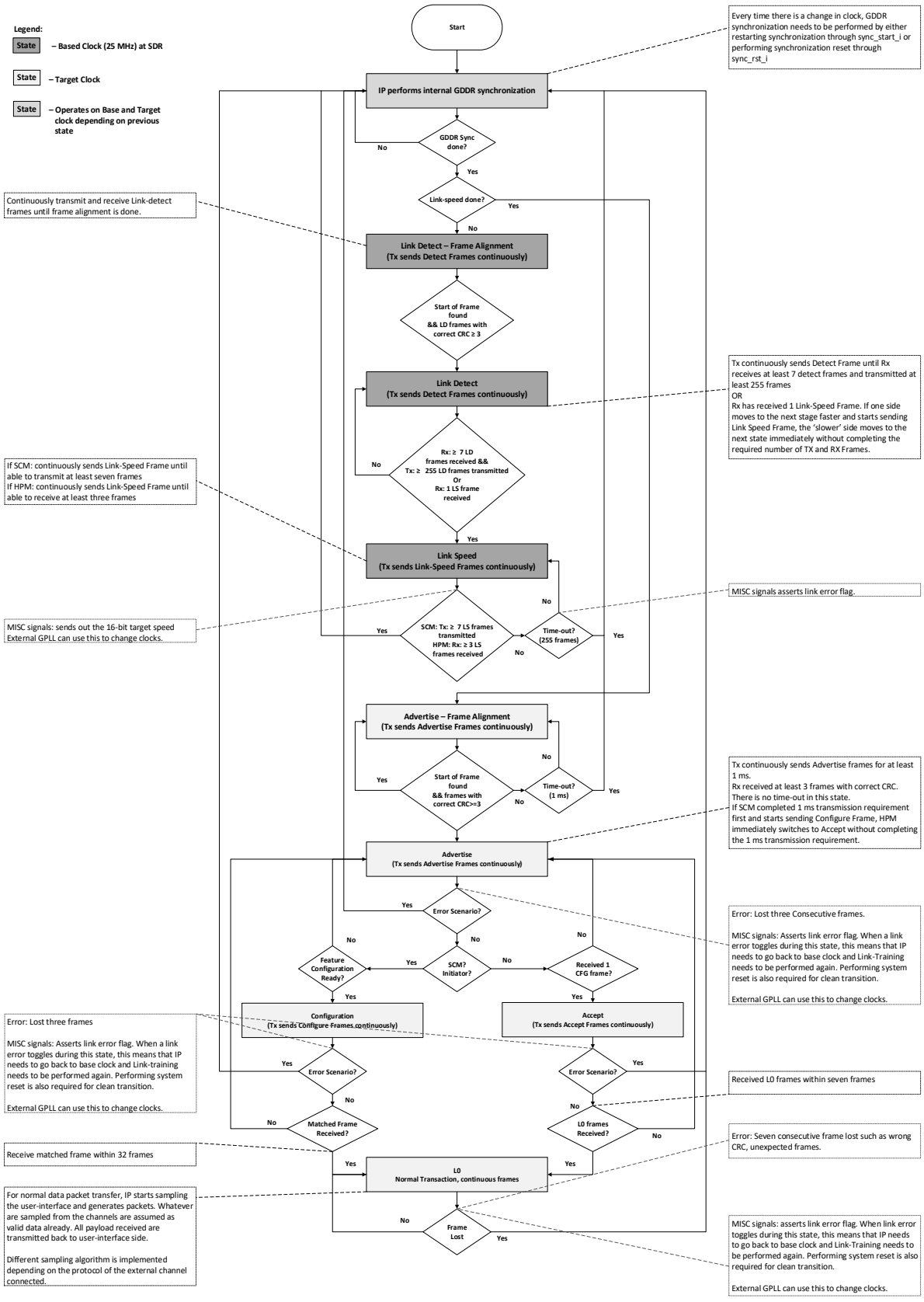

Figure 3.1. LTPI State Flow

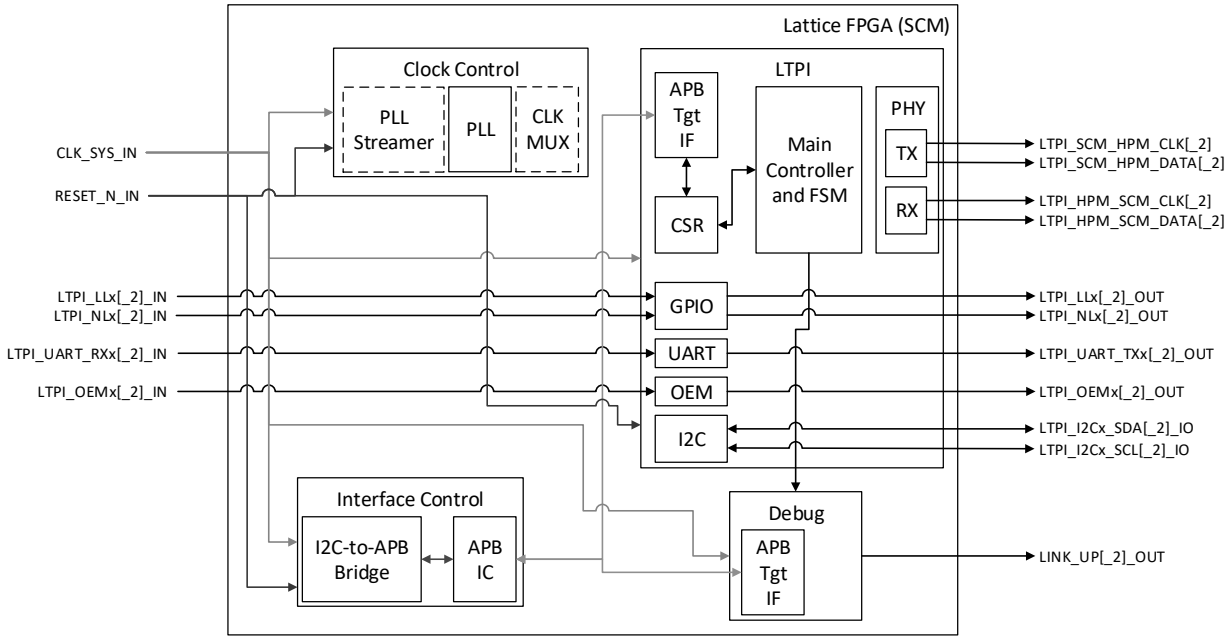

Figure 3.2. SCM Block Diagram

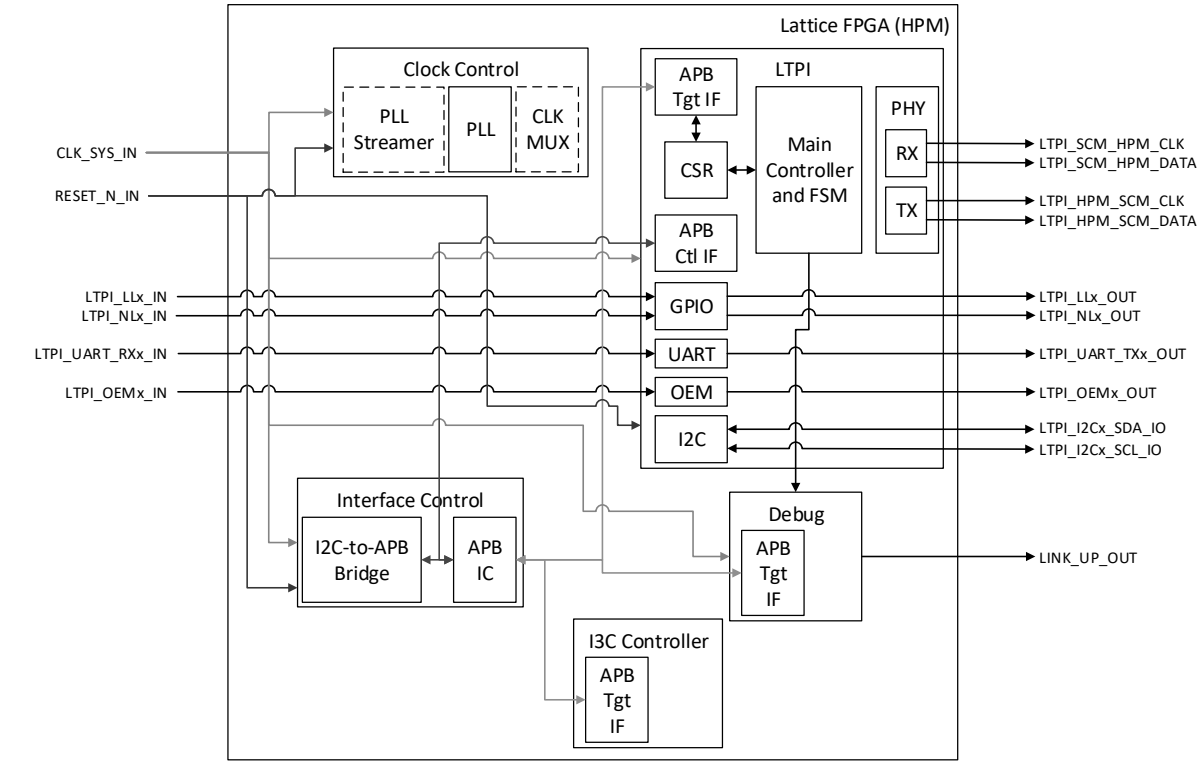

Figure 3.3. HPM Block Diagram

3.1. Design Components

This section outlines the module components of the top-level module in the reference design. Note that the bus width and other parameters are based on how the reference design is structured.

3.1.1. LTPI

This module utilizes the DC-SCM LTPI IP from Lattice Propel. Depending on the configuration, it can be either an SCM or HPM instance.

Module Name: scm0_<devicename> (for SCM instance) or hpm0_<devicename> (for HPM instance).

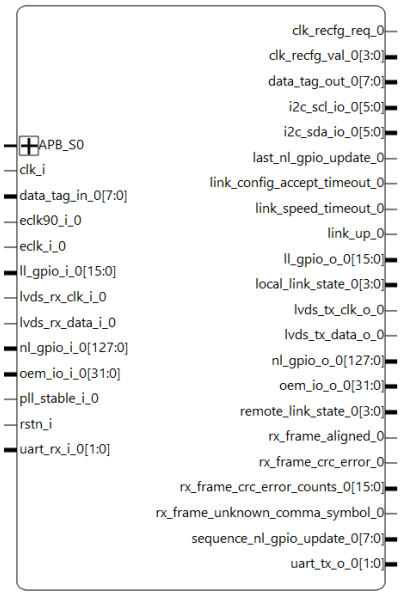

Figure 3.4. LTPI (SCM) Block Diagram

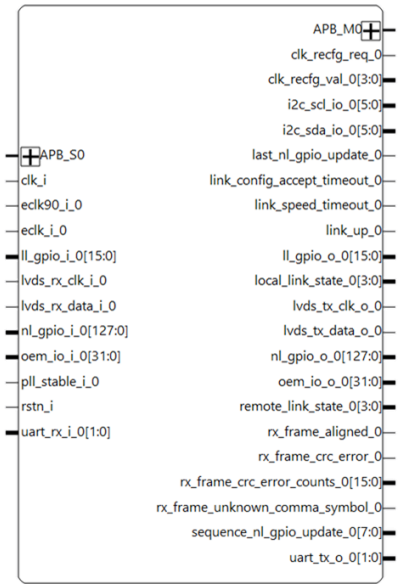

Figure 3.5. LTPI (HPM) Block Diagram

Table 3.1. LTPI Clock Source Signals

Port Name	Input/Output	Width	Default Value	Description
System Signals				
clk_i	Input	1	N/A	System Clock
rstn_i	Input	1	N/A	Active low reset
eclk_i_{node}	Input	1	N/A	ECLK for PHY
eclk90_i_{node}	Input	1	N/A	ECLK 90^{O} for PHY
LTPI Signals				
lvds_rx_clk_i_{node}	Input	1	N/A	LVDS Receive Clock
lvds_rx_data_i_{node}	Input	1	N/A	LVDS Receive Data
lvds_tx_clk_o_{node}	Output	1	N/A	LVDS Transmit Clock
lvds_tx_data_o_{node}	Output	1	N/A	LVDS Transmit Data
Low Latency GPIO Channel				
ll_gpio_i_{node}	Input	16	N/A	Low Latency General Purpose Input
ll_gpio_o_{node}	Output	16	N/A	Low Latency General Purpose Output
Normal Latency GPIO Channel				
nl_gpio_i_{node}	Input	128	N/A	Normal Latency General Purpose Input
nl_gpio_o_{node}	Output	128	N/A	Normal Latency General Purpose Output
OEM Channel				
oem_io_i_{node}	Input	32	N/A	OEM Input
oem_io_o_{node}	Output	32	N/A	OEM Output
UART Channel				
uart_tx_o_{node}	Output	2	N/A	UART Tx
uart_rx_i_{node}	Input	2	N/A	UART Rx
I2C Channel				
i2c_scl_io_{node}	Inout	6	Tri-state	I2C Serial Clock
i2c_sda_io_{node}	Inout	6	Tri-state	I2C Serial Data
Data Channel				
data_tag_in_{node}	Input	8	8'h0	Data channel input tag (SCM only). Unused
data_tag_out_{node}	Output	8	N/A	Data channel output tag (SCM only). Unused
APB_M0	APB IF	N/A	N/A	Data Channel APB Requester Interface (HPM only)
APB Interface				
APB_S0	APB IF	N/A	N/A	• CSR APB Completer Interface (for both SCM and HPM) • Data Channel APB Completer Interface (SCM only)
Debug Mode Signals				
local_link_state_{node}	Output	4	4'h0	Local LTPI link state: 4'h0: Frame Alignment/Link Detect 4'h1: Link Speed 4'h2: Advertise 4'h3: Configuration (SCM)/Accept (HPM) 4'h4: Active

Port Name	Input/Output	Width	Default Value	Description
remote_link_state_{node}	Output	4	4'h0	Remote LTPI link state: 4'h0: Frame Alignment/Link Detect 4'h1: Link Speed 4'h2: Advertise 4'h3: Configuration (SCM)/Accept (HPM) 4'h4: Active
sequence_nl_gpio_update_{node}	Output	8	8'h0	Normal Latency GPIO Frame Counter
rx_frame_crc_error_counts_{node}	Output	16	16'h0	Consecutive CRC error count
rx_frame_aligned_{node}	Output	1	0	Frame aligned flag
rx_frame_crc_error_{node}	Output	1	0	CRC error flag
rx_frame_unknown_comma_symbol_{node}	Output	1	0	Received unknown comma symbol (Frame Byte 0) flag
last_nl_gpio_update_{node}	Output	1	0	End of Normal Latency GPIO round-robin cycle flag
link_config_accept_timeout_{node}	Output	1	0	Configuration/Accept State timeout flag
link_speed_timeout_{node}	Output	1	0	Link Speed timeout flag
Miscellaneous Signals				
link_up_{node}	Output	1	N/A	Link established flag
pll_stable_i_{node}	Input	1	N/A	PLL stable input
clk_recfg_val_{node}	Output	4	N/A	Clock re-configuration value
clk_recfg_req_{node}	Output	1	N/A	Clock re-configuration request flag

3.1.2. PLL Streamer Logic (MachXO5-NX only)

This module manages the process of changing the LTPI Clock Source frequency from Base Speed to Target Speed and vice versa.

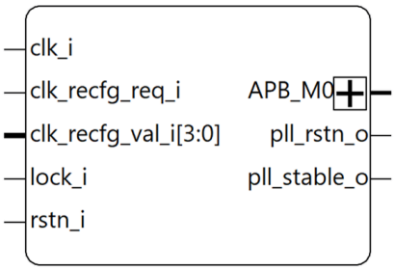

Figure 3.6. PLL Streamer Logic Block Diagram

Table 3.2. PLL Streamer Logic Signals

Port Name	Input/Output	Width	Default Value	Description
clk_i	Input	1	N/A	System clock
rstn_i	Input	1	N/A	Active-low reset
clk_recfg_req_i	Input	1	N/A	Clock reconfiguration request based on clk_recfg_val_i
clk_recfg_val_i	Input	4	N/A	Indicates the requested clock frequency of the LTPI instance 4'd0 – 25 MHz (default) 4'd1 – 50 MHz 4'd2 – 75 MHz 4'd3 – 100 MHz 4'd4 – 150 MHz 4'd5 – 200 MHz 4'd6 – 250 MHz 4'd7 – 300 MHz 4'd8 – 400 MHz 4'd9 – 600 MHz Other values – reserved
lock_i	Input	1	N/A	LTPI Clock Source lock
APB_M0	APB IF	N/A	N/A	APB Requester Interface for PLL CSR access
pll_rstn_o	Output	1	0	LTPI Clock Source reset trigger
pll_stable_o	Output	1	0	LTPI Clock Source stable flag

3.1.3. LTPI Clock Source

This module serves as a wrapper for a PLL instance, providing the necessary clocks for the LTPI. For MachXO3 and MachXO3D projects, additional logic is added to the PLL instance. For MachXO5-NX, it is an instance of the PLL IP.

Module name: dynamic_pll_wrapper (MachXO3/MachXO3D), dpll_xo5 (MachXO5-NX)

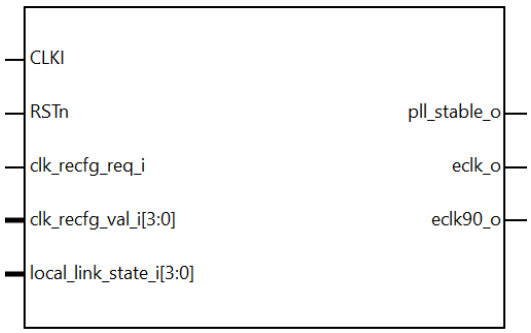

Figure 3.7. LTPI Clock Source (MachXO3/MachXO3D) Block Diagram

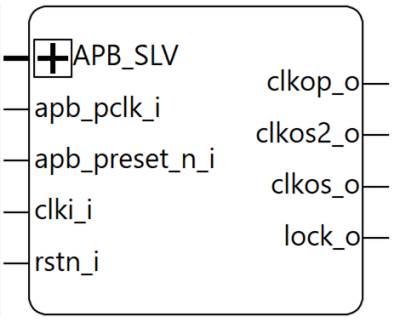

Figure 3.8. LTPI Clock Source (MachXO5-NX) Block Diagram

Table 3.3. LTPI Clock Source Signals

Port Name	Input/Output	Width	Default Value	Description
MachXO3/MachXO3D				
CLKI	Input	1	N/A	System clock
RSTn	Input	1	N/A	Active-low reset
clk_recfg_req_i	Input	1	N/A	Clock re-configuration request from LTPI IP
clk_recfg_val_i	Input	4	N/A	Clock re-configuration value from LTPI IP
local_link_state_i	Input	4	N/A	Local link state of LTPI IP
eclk_o	Output	1	N/A	LTPI ECLK source
eclk90_o	Output	1	N/A	LTPI ECLK 90 source
pll_stable_o	Output	1	1’b0	LTPI Clock Source stable flag
MachXO5-NX				
clki_i	Input	1	N/A	System clock
rstn_i	Input	1	N/A	Active-low reset. Connects to pll_rstn_o of PLL Streamer Logic.
apb_pclk_i	Input	1	N/A	APB clock. Tied to clki_i
apb_preset_n_i	Input	1	N/A	APB active low-reset. Tied to rstn_i
APB_SLV	APB IF	N/A	N/A	APB Completer Interface for CSR access
clkop_o	Output	1	N/A	LTPI ECLK source
clkos_o	Output	1	N/A	LTPI ECLK 90 source
clkos2_o	Output	1	N/A	Reserved
lock_o	Output	1	1’b0	PLL lock

3.1.4. I2C-to-APB Bridge

This module utilizes the I2C-to-APB Bridge Reference Design from Lattice Propel. It converts external I2C commands into APB commands.

Module name: i2c_to_apb0

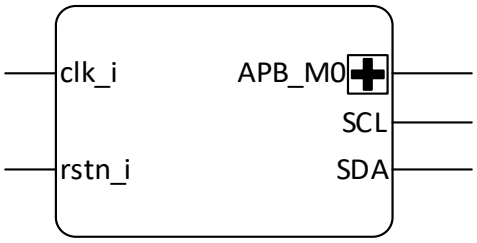

Figure 3.9. I2C-to-APB Bridge Block Diagram

Table 3.4. I2C-to-APB Bridge Signals

Port Name	Input/Output	Width	Default Value	Description
clk_i	Input	1	N/A	System Clock
rstn_i	Input	1	N/A	Active-low reset
APB_M0	APB IF	N/A	N/A	APB Requester Interface
SCL	Inout	1	Tri-state	I2C Serial Clock
SDA	Inout	1	Tri-state	I2C Serial Data

3.1.5. LTPI Debug Module

This module is used to capture the debug mode signals of the LTPI IP to a FIFO module. A capture is made when any of the signals change. For Dual Node implementation, there are two instances of this module.

Module name: ltpi_debug

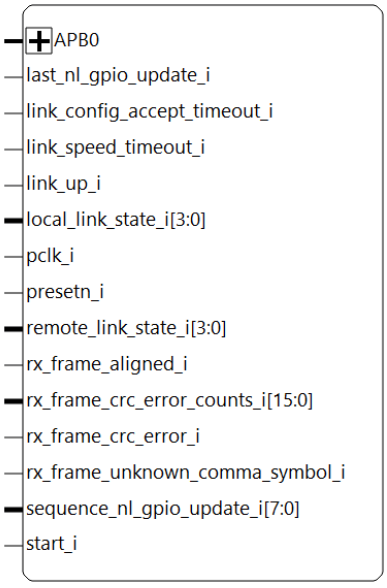

Figure 3.10. LTPI Debug Module Block Diagram

Table 3.5. LTPI Debug Module Signals

Port Name	Input/Output	Width	Default Value	Description
pclk_i	Input	1	N/A	System clock
presetn_i	Input	1	N/A	Active-low reset
APB0	APB IF	N/A	N/A	APB Completer Interface for CSR access
link_up_i	Input	1	N/A	Link established flag
local_link_state_i	Input	4	N/A	Local LTPI link state: 4'h0: Frame Alignment/Link Detect 4'h1: Link Speed 4'h2: Advertise 4'h3: Configuration (SCM)/Accept (HPM) 4'h4: Active
remote_link_state_i	Input	4	N/A	Remote LTPI link state: 4'h0: Frame Alignment/Link Detect 4'h1: Link Speed 4'h2: Advertise 4'h3: Configuration (SCM)/Accept (HPM) 4'h4: Active
sequence_nl_gpio_update_i	Input	8	N/A	Normal Latency GPIO Frame Counter
rx_frame_crc_error_counts_i	Input	16	N/A	Consecutive CRC error count
rx_frame_aligned_i	Input	1	N/A	Frame aligned flag
rx_frame_crc_error_i	Input	1	N/A	CRC error flag
rx_frame_unknown_comma_symbol_i	Input	1	N/A	Received unknown comma symbol (Frame Byte 0) flag
last_nl_gpio_update_i	Input	1	N/A	End of Normal Latency GPIO round-robin cycle flag
link_config_accept_timeout_i	Input	1	N/A	Configuration/Accept State timeout flag
link_speed_timeout_i	Input	1	N/A	Link Speed timeout flag

3.1.6. I3C Controller (MachXO3D and MachXO5-NX only, HPM only)

This module is an instance of the I3C Controller IP. Access to this module is only through the LTPI data channel.

Module name: i3c_ctl0_{device_name}

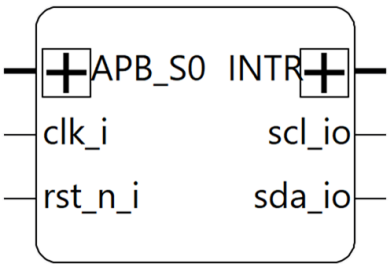

Figure 3.11. I3C Controller Block Diagram

Table 3.6. I3C Controller Signals

Port Name	Input/Output	Width	Default Value	Description
clk_i	Input	1	N/A	System clock
rst_n_i	Input	1	N/A	Active-low reset
APB_S0	APB IF	N/A	N/A	APB Completer Interface for CSR access
scl_io	Inout	1	Tri-state	I3C Serial Clock
sda_io	Inout	1	Tri-state	I3C Serial Data
INTR	Input	N/A	N/A	Interrupt Request source. Unused

3.2. Clocking Scheme

The reference design is currently limited to a 50 MHz system clock. This 50 MHz source is essential for generating the derived clocks required by the DC-SCM LTPI IP (MachXO5-NX only) and is currently the maximum allowed frequency of the I3C Controller. This configuration enables a single set of computed parameters to produce LTPI-compliant operating frequencies as defined by the specification – up to 200 MHz DDR or 400 MHz SDR.

3.3. Reset Scheme

The reference design uses an active-low reset source for its system reset of all modules except for the LTPI Clock Source module for MachXO5-NX (the reset for this module comes from the PLL Streamer Logic).

3.4. DC-SCM LTPI IP Control and Status Registers

Refer to the DC-SCM LTPI IP User Guide (FPGA-IPUG-02200) for the DC-SCM LTPI IP register description information.

3.5. GPIO Latency

The latency for the LL GPIO is at best case 700 ns (at 800 Mbps for MachXO3 and MachXO3D) and 425 ns (at 1200 Mbps for MachXO5-NX). This is measured from the time the transmitting module system clock samples the data on the LL Input Interface to the time the LL Output Interface of the receiver module reflects the data.

The latency for the NL GPIO depends on the ratio of the number of NL GPIO and the number of NL GPIO per frame (rounded up). This pertains to how many frames we need to send before the next frame contains an update to the same NL GPIO bits. Below is a sample computation:

Number of NL GPIO : 130

Number of NL GPIO per Frame: 16

Target Device: MachXO3

Total Delay (Best Case) = 700 ns × ceiling(130/16) = 6300 ns

4. Signal Description

This section describes the top module signals of the reference design.

Table 4.1. LTPI Clock Source Signals

Port Name	Input/Output	Width	Default Value	Description
System				
CLK_SYS_IN	Input	1	N/A	50 MHz System Clock Input
RESETN_IN	Input	1	N/A	Active low reset
LINK_UP*{_2}*_OUT	Output	1	0	Link Established Flag. Asserts when LTPI link reaches Active State
I2C-to-APB Bridge				
SCL_APB_IO	Inout	1	Tri-state	I2C Serial Clock
SDA_APB_IO	Inout	1	Tri-state	I2C Serial Data
I3C Controller (HPM only except MachXO3)				
SCL_I3C_IO	Inout	1	Tri-state	I3C Serial Clock
SDA_I3C_IO	Inout	1	Tri-state	I3C Serial Data
LTPI (Interface)				
LTPI_SCM_HPM_CLK*{_2}*	Input (HPM) Output (SCM)	1	N/A	LTPI Clock from SCM to HPM
LTPI_SCM_HPM_DATA*{_2}*	Input (HPM) Output (SCM)	1	N/A	LTPI Data from SCM to HPM
LTPI_HPM_SCM_CLK*{_2}*	Input (SCM) Output (HPM)	1	N/A	LTPI Clock from HPM to SCM
LTPI_HPM_SCM_DATA*{_2}*	Input (SCM) Output (HPM)	1	N/A	LTPI Data from HPM to SCM
LTPI (Channel)				
LTPI_LL*{_2}*_IN	Input	16	N/A	Low Latency General Purpose Input
LTPI_LL*{_2}*_OUT	Output	16	16'h0	Low Latency General Purpose Output
LTPI_NL*{_2}*_IN	Input	128	N/A	Normal Latency General Purpose Input
LTPI_NL*{_2}*_OUT	Output	128	128'h0	Normal Latency General Purpose Output
LTPI_OEM*{_2}*_IN	Input	32	N/A	OEM Input
LTPI_OEM*{_2}*_OUT	Output	32	32'h0	OEM Output
LTPI_UART_RX*{_2}*_IN	Input	2	N/A	UART Rx
LTPI_UART_TX*{_2}*_OUT	Output	2	N/A	UART Tx
LTPI_I2C_SCL*{_2}*_IO	Inout	6	Tri-state	I2C Serial Clock
LTPI_I2C_SDA*{_2}*_IO	Inout	6	Tri-state	I2C Serial Data

5. Register Description

This section defines the register mapping of the SCM and HPM projects. Specific descriptions for the I3C Controller registers are covered in FPGA-IPUG-02228.

Note that the register descriptions in this section are specific to the IP versions described in the Quick Facts section. Future updates of these IP may affect the available registers.

Table 5.1. SCM Register Map

DWORD Address Offset Range	Module	Accessed Via	Description
0x00000800 – 0x00000BFF	LTPI Debug	I2C-to-APB	LTPI Debug FIFO Register
0x40000000 – 0x400001FF	LTPI	I2C-to-APB	CPLD/FPGA Information: The offset is reserved for OEM-defined CPLD or FPGA information, such as CPLD or FPGA image version numbers, device identifiers, and OEM-specific information.
0x40000200 – 0x400003FF	LTPI	I2C-to-APB	LTPI Control and Status: LTPI Status and Link Training Control Registers
0x40000400 – 0x7FFFFFFF	LTPI	I2C-to-APB	LTPI Memory Mapped Channel: Address Space reserved for memory-mapped channel
0x80000000 – 0x800001FF	LTPI (Node 2)	I2C-to-APB	CPLD/FPGA Information: The offset is reserved for OEM-defined CPLD or FPGA information, such as CPLD or FPGA image version numbers, device identifiers, and OEM-specific information.
0x80000200 – 0x800003FF	LTPI (Node 2)	I2C-to-APB	LTPI Control and Status: LTPI Status and Link Training Control Registers
0x80000400 – 0xBFFFFFFF	LTPI (Node 2)	I2C-to-APB	LTPI Memory Mapped Channel: Address Space reserved for memory-mapped channel

Table 5.2. HPM Register Map

DWORD Address Offset Range	Module	Accessed Via	Description
0x00000400 – 0x000005FF	LTPI	I2C-to-APB, Data Channel	CPLD/FPGA Information: The offset is reserved for OEM-defined CPLD or FPGA information, such as CPLD or FPGA image version numbers, device identifiers, and OEM-specific information.
0x00000600 – 0x400007FF	LTPI	I2C-to-APB, Data Channel	LTPI Control and Status: LTPI Status and Link Training Control Registers
0x00000800 – 0x00000BFF	LTPI Debug	I2C-to-APB, Data Channel	LTPI Debug FIFO Register
0x00001000 – 0x000013FF	I3C Controller	Data Channel	LTPI Memory Mapped Channel: Address Space reserved for memory-mapped channel

Table 5.3. LTPI Debug Module Register Summary

DWORD Address Offset	Register Name	Access	Description
0x0	State FIFO	RO	This register stores the most recent unread Debug Mode values. Each read operation retrieves the current entry and subsequently advances the FIFO to load the next pending value.

Table 5.4. LTPI Register Summary

<table>
<tr><th>DWORD Address Offset</th><th>Register Name</th><th>Bit Field</th><th>Access</th><th>Description</th></tr>
<tr><td rowspan="15">0x00</td><td rowspan="15">LTPI Link Status</td><td>31:20</td><td>–</td><td>Reserved</td></tr>
<tr><td>19:16</td><td>RO</td><td>Local LTPI Link State<table><tr><th>State</th><th>Encoding</th></tr><tr><td>Link Detect</td><td>0x0</td></tr><tr><td>Link Speed</td><td>0x1</td></tr><tr><td>Advertise</td><td>0x2</td></tr><tr><td>Configuration</td><td>0x3</td></tr><tr><td>Operational</td><td>0x4</td></tr><tr><td>Reserved</td><td>0x5 – 0xF</td></tr></table></td></tr>
<tr><td>15:12</td><td>RO</td><td>Remote LTPI Link State<table><tr><th>State</th><th>Encoding</th></tr><tr><td>Link Detect</td><td>0x0</td></tr><tr><td>Link Speed</td><td>0x1</td></tr><tr><td>Advertise</td><td>0x2</td></tr><tr><td>Configuration</td><td>0x3</td></tr><tr><td>Operational</td><td>0x4</td></tr><tr><td>Reserved</td><td>0x5 – 0xF</td></tr></table></td></tr>
<tr><td>11:8</td><td>RO</td><td>LTPI Link Speed<table><tr><th>State</th><th>Encoding</th></tr><tr><td>Base Freq x1 (25 MHz)</td><td>0x0</td></tr><tr><td>Base Freq x2 (50 MHz)</td><td>0x1</td></tr><tr><td>Base Freq x3 (75 MHz)</td><td>0x2</td></tr><tr><td>Base Freq x4 (100 MHz)</td><td>0x3</td></tr><tr><td>Base Freq x6 (150 MHz)</td><td>0x4</td></tr><tr><td>Base Freq x8 (200 MHz)</td><td>0x5</td></tr><tr><td>Base Freq x10 (250 MHz)</td><td>0x6</td></tr><tr><td>Base Freq x12 (300 MHz)</td><td>0x7</td></tr><tr><td>Base Freq x16 (400 MHz)</td><td>0x8</td></tr><tr><td>Base Freq x24 (600 MHz)</td><td>0x9</td></tr><tr><td>Base Freq x32 (800 MHz)</td><td>0xA</td></tr><tr><td>Base Freq x40 (1 GHz)</td><td>0xB</td></tr><tr><td>Reserved</td><td>0xC – 0xF</td></tr></table></td></tr>
<tr><td>7</td><td>RO</td><td>DDR Mode:
0 – SDR Mode
1 – DDR Mode</td></tr>
<tr><td>6</td><td>–</td><td>Reserved</td></tr>
<tr><td>5</td><td>RWC</td><td>Link Configure/Accept Timeout Error</td></tr>
<tr><td>4</td><td>RWC</td><td>Link Speed Timeout Error</td></tr>
<tr><td>3</td><td>RWC</td><td>Unknown Comma Symbol Error</td></tr>
<tr><td>2</td><td>RWC</td><td>Frame CRC Error</td></tr>
<tr><td>1</td><td>RWC</td><td>LTPI Link Lost Error</td></tr>
</table>

DWORD Address Offset	Register Name	Bit Field	Access	Description
		0	RWC	LTPI Link Aligned: 0 – Link Not Aligned 1 – Link Aligned
0x04	LTPI Detect Capabilities Local	31:24	—	Reserved
		23:8	RW	Link Speed Capabilities
		7:4	RO	Local LTPI Major Version
		3:0	RO	Local LTPI Minor Version
0x04	LTPI Detect Capabilities Local	31:24	—	Reserved
		23:8	RW	Link Speed Capabilities
		7:4	RO	Local LTPI Major Version
		3:0	RO	Local LTPI Minor Version

Table 5.5. I3C Controller IP Register Summary

DWORD Address Offset	Register Name	Description
0x004	System Clock Divider	System clock divider for SCL clock generation
0x008	Controller Configuration 0	Controller configuration
0x00C	Open Drain Timer	Number of SCL clock cycles to generate open-drain clock pulse width
0x010	I2C Clock Divider	System clock divider for SCL clock generation in I2C mode
0x014	Pull Up Resistor Disable	SDA and SCL pull-up resistor configuration
0x018	HDR-DDR Configuration 0	Controller configuration for HDR-DDR mode
0x020	Soft Reset	Soft reset
0x044	Transmit Start	Transmit start signal to initiate processing of Tx FIFO contents
0x054	Secondary Controller Status Info	Secondary controller status information
0x058	Set Device Role	Manually set the device role
0x05C 0x060 0x064	Secondary Controller Timeout	Controller role handoff timeout value
0x070	Dynamic Address ACKed Counter	Number of dynamic addresses that acknowledged during the Enter DAA command
0x074	IBI Read Count	Number of bytes to read from the target during IBI
0x078	IBI Response	Response to IBI or Hot-Join request
0x07C	IBI Address	Address of the I3C target or secondary controller that requested the IBI
0x080	Interrupt Status 0	Interrupt status 0
0x084	Interrupt Set 0	Interrupt set 0
0x088	Interrupt Enable 0	Interrupt enable 0
0x090	Interrupt Status 1	Interrupt status 1
0x094	Interrupt Set 1	Interrupt set 1
0x098	Interrupt Enable 1	Interrupt enable 1
0x0A0	Bus Condition Debug	Bus condition information for debug
0x0A4	Last NAK Address Debug	Last NAK target address
0x0A8	Last ACK Address Debug	Last ACK target address
0x0B0	Interrupt Status 2	Interrupt status 2
0x0B4	Interrupt Set 2	Interrupt set 2
0x0B8	Interrupt Enable 2	Interrupt enable 2
0x0C0	Tx FIFO	Transmit FIFO

DWORD Address Offset	Register Name	Description
0x100	Rx FIFO	Receive FIFO
0x104 – 0x3FC	Reserved	Reserved

6. Timing Constraints

For timing constraint guidelines, refer to the Timing Analysis for High-Speed GDDR Interfaces in the following documents:

- Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057) for MachXO3 devices.
- Implementing High-Speed Interfaces with MachXO3D Usage Guide (FPGA-TN-02065) for MachXO3D devices.
- MachXO5-NX High-Speed I/O Interface (FPGA-TN-02286) for MachXO5-NX devices.

6.1. Clock Definitions

The following clock frequencies are set either manually (preset by the reference design) or automatically (preset by the software). All are referenced to input or internal clocks of the DC-SCM LTPI IP.

- CLK_SYS_IN: Clock supplied to the LTPI Clock Source PLL as well as the system clock of all other modules in the reference design except for the LTPI IP and I2C-to-APB Bridge module. This clock must be set to 25 MHz and is constrained by the reference design to 25 MHz.
- eclk_i: Clock supplied as the ECLK of the LTPI IP. This clock is constrained based on the LTPI Clock Source PLL setting (software-defined). This clock varies from 25 MHz (during Base Speed states) up to 200 MHz for DDR or 400 MHz for SDR.
- eclk90_i: Clock supplied as the ECLK90 of the LTPI IP and is 90 degrees out of phase with eclk_i. This clock is constrained based on the LTPI Clock Source PLL setting (software-defined). This clock varies from 25 MHz (during Base Speed states) up to 200 MHz for DDR or 400 MHz for SDR.
- LTPI_SCM_HPM_CLK (for HPM)/LTPI_HPM_SCM_CLK (for SCM): Expected LVDS RX Clock frequency. This clock is constrained by the reference design to 200 MHz.
- tx_sclk_o: Generated SCLK by the Tx PHY module. This clock is:
  - 1/4 of the ECLK and constrained by the reference design to 50 MHz for MachXO3 and MachXO3D devices.
  - 1/5 of the ECLK and constrained by the reference design to 40 MHz for MachXO5-NX devices.
- rx_sclk_o: Generated SCLK by the Rx PHY module. This clock is:
  - 1/4 of the ECLK and constrained by the reference design to 50 MHz for MachXO3 and MachXO3D devices.
  - 1/5 of the ECLK and constrained by the reference design to 40 MHz for MachXO5-NX devices.

6.2. Clocks, Resets, and False Paths

All reset paths are asynchronous and can be set as false paths. Data transfers and clock domain crossings are already handled in the reference design. All clocks defined in the Clock Definitions section can also be set as false paths.

7. Modifying the Reference Design

The reference design is pre-compiled and ready to use upon download. If the intent is to run the default design on hardware for evaluation purposes, proceed to the Hardware Implementation section.

This section discusses the software design flow if changes to the reference design need to be made.

IMPORTANT: Ensure that the linked project for Lattice Diamond or Lattice Radiant is not open when making any modifications in the Lattice Propel project, and vice versa.

7.1. Lattice Propel

This section discusses changes that the user can make to the reference design using Lattice Propel.

For more information on Lattice Propel usage, refer to the Lattice Propel 2024.2 Builder User Guide (FPGA-UG-02219).

7.1.1. Opening the Lattice Propel Project

1. Unzip the reference design ZIP file.
2. Open Lattice Propel 2024.2. Note that the reference design is tested using this version and does not guarantee backward compatibility with previous Lattice Propel versions.
3. Click the Open Design button or go to **File** > **Open Design**.

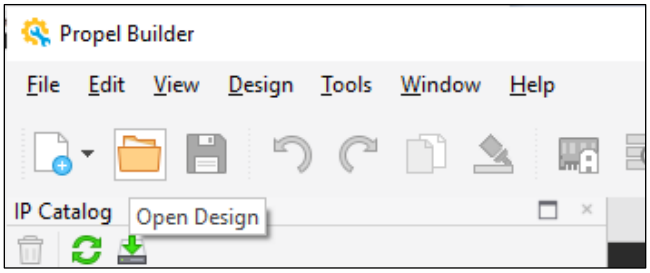

Figure 7.1. Open Design Button

4. Select the project file (*.sbx) found in *<Reference_Design_Folder>/Projects/<Design_Folder>/<Design_Folder>*. In this section, the design used as an example is **MachXO3D_SCM**.

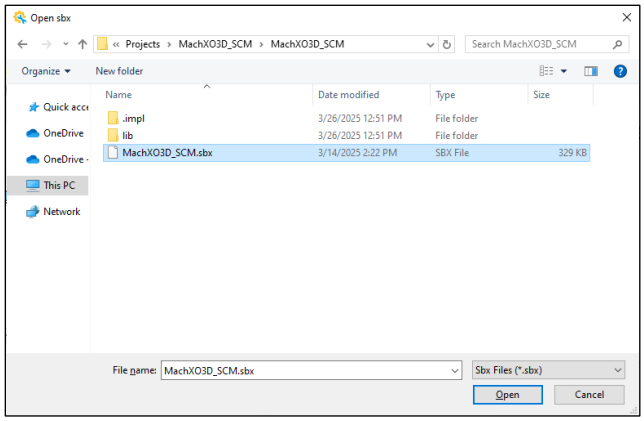

Figure 7.2. Selecting the Lattice Propel Project

7.1.2. Add/Remove Top-Level Module Ports

All module ports that are connected to an external port (input , output or inout) are part of the top-level module port list.

- To remove a top-level port, do any of the following:
  - Right-click the external port and select Disconnect.
  - Click the external port and press Delete.
- To add a top-level port to an unconnected module port, right-click.
- To add a top-level port to connected module port(s) or net(s), do the following:
  - Right-click on any blank space in the Schematic Window and select Create Port.
  - Configure the external port.
  - Connect to the module port or net.

Click **Save** after making changes.

7.1.3. Changing the DC-SCM LTPI IP Settings

This section provides an overview of what users can change in the IP GUI. All figures in this section show the default IP settings, not the SCM or HPM project settings. Each subsection describes the settings made for each tab (if applicable) and other changes that can be made to customize the reference design.

For more information on the IP function, refer to the DC-SCM LTPI IP User Guide (FPGA-IPUG-02200).

After making changes to the IP:

1. Click **Generate**.
2. Review the generated result.
3. Check or Uncheck the **Insert to project** option, depending on your needs.
4. Click **Finish**.
5. Create top-level ports, if necessary.
6. Click **Save**.

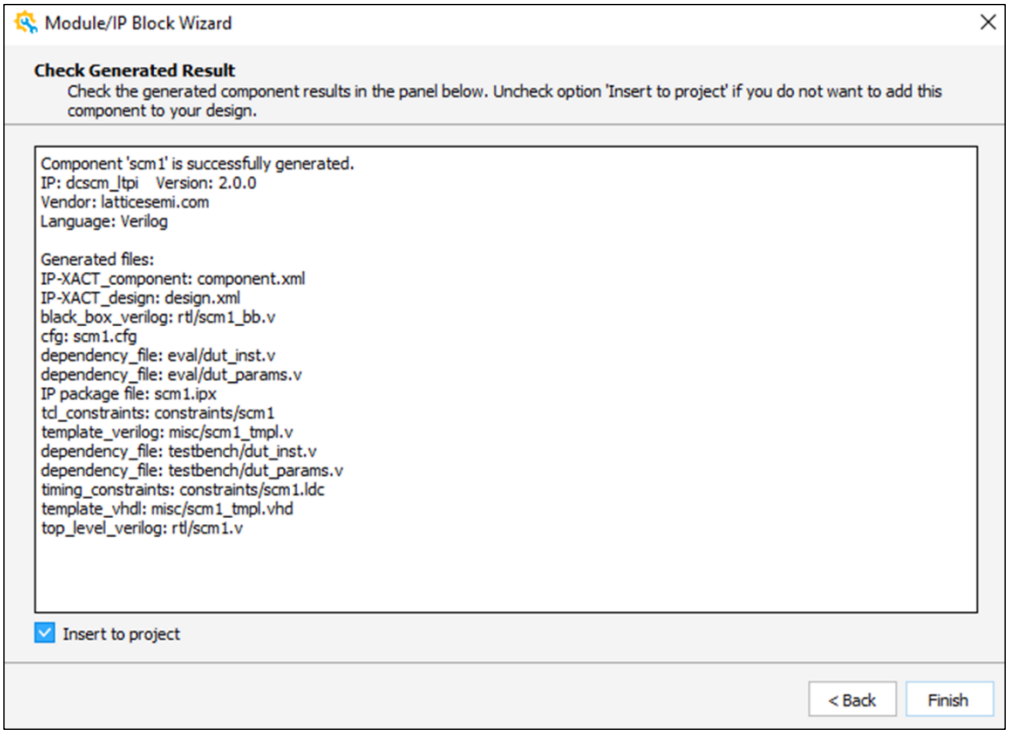

Figure 7.3. Check Generated Result Window

7.1.3.1. General Tab

This section provides descriptions of the parameters in the General tab.

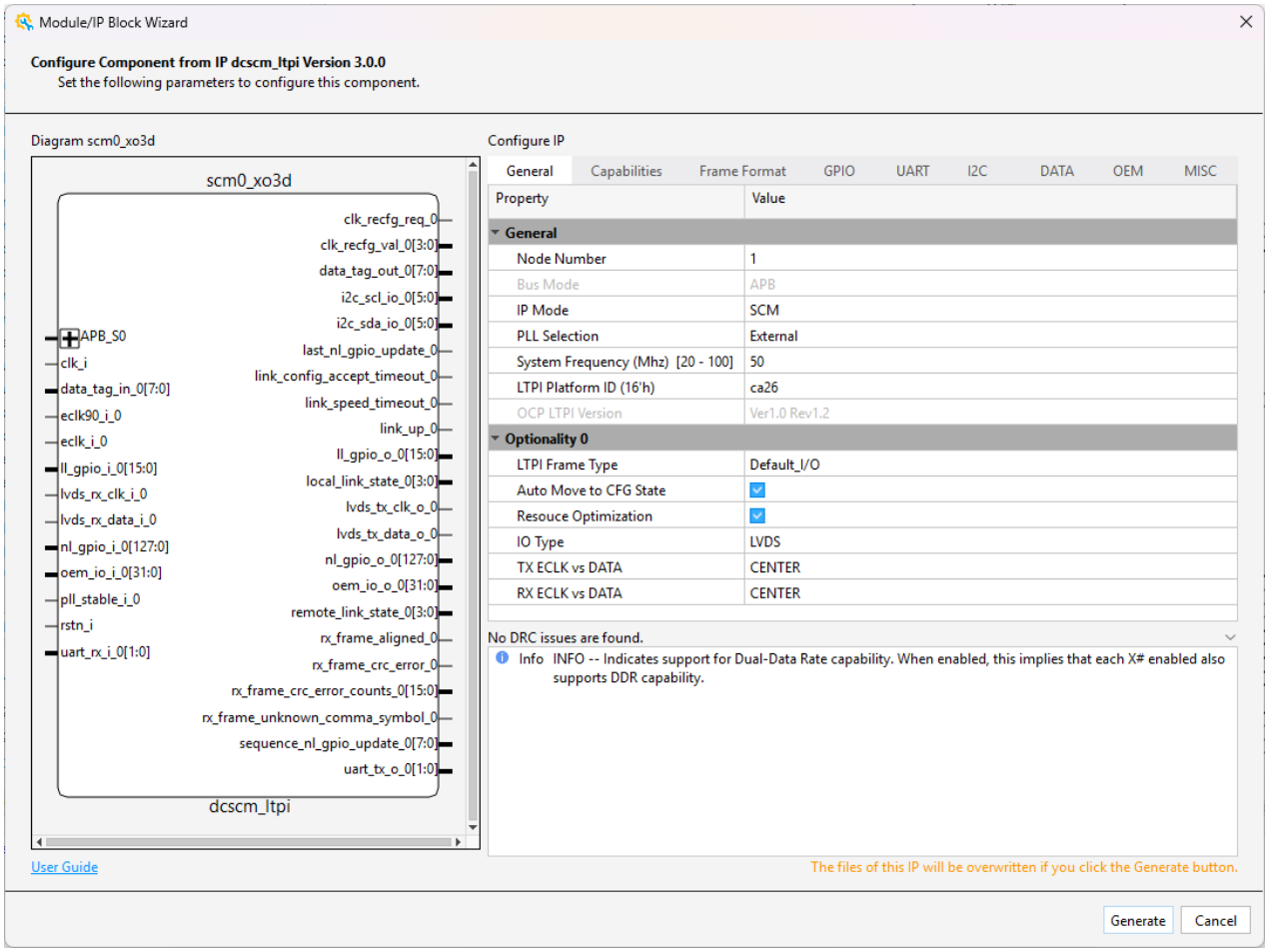

Figure 7.4. General Tab

Table 7.1. General Tab Parameter Descriptions

Parameter	Description
General	
Node Number	Number of node instances. Applicable for SCM only.
Bus Mode	Interface Mode for CSR and Data Channel. APB only.
IP Mode	SCM or HPM
PLL Selection	Determines if the PLL to be used for the ECLK generation would be internal within the IP instance or an separate external instance
System Frequency (MHz) [20 - 100]	Expected input clock on clk_i
LTPI Platform ID (16'h)	Instance unique Platform ID as defined by the LTPI specifications
OCP LTPI Version	Supported LTPI version of the IP. Ver 1.0 Rev 1.2 only.
Optionality (node)	
LTPI Frame Type	Can be the default I/O frame as defined by the LTPI specifications or OEM-specific definition
Auto Move to CFG State	Specifies that when IP Mode == SCM, the IP automatically goes to configuration state after completing the required frame transmission for Advertise State. When unchecked, you must perform manual feature request related programming. Refer to the requested features details in the DC-SCM LTPI IP User Guide (FPGA-IPUG-02200) for more information.
Resource Optimization	SPEED_CAP[1] = 1 when enabled. SPEED_CAP[1] = 0 when disabled. This setting is disabled in the reference design.
IO Type	LTPI port setting. LVDS only
TX ECLK vs DATA	DDR setting for transmission alignment between clock and data
RX ECLK vs DATA	DDR setting for receive alignment between clock and data

7.1.3.2. Capabilities Tab

This section provides descriptions of the parameters in the Capabilities tab.

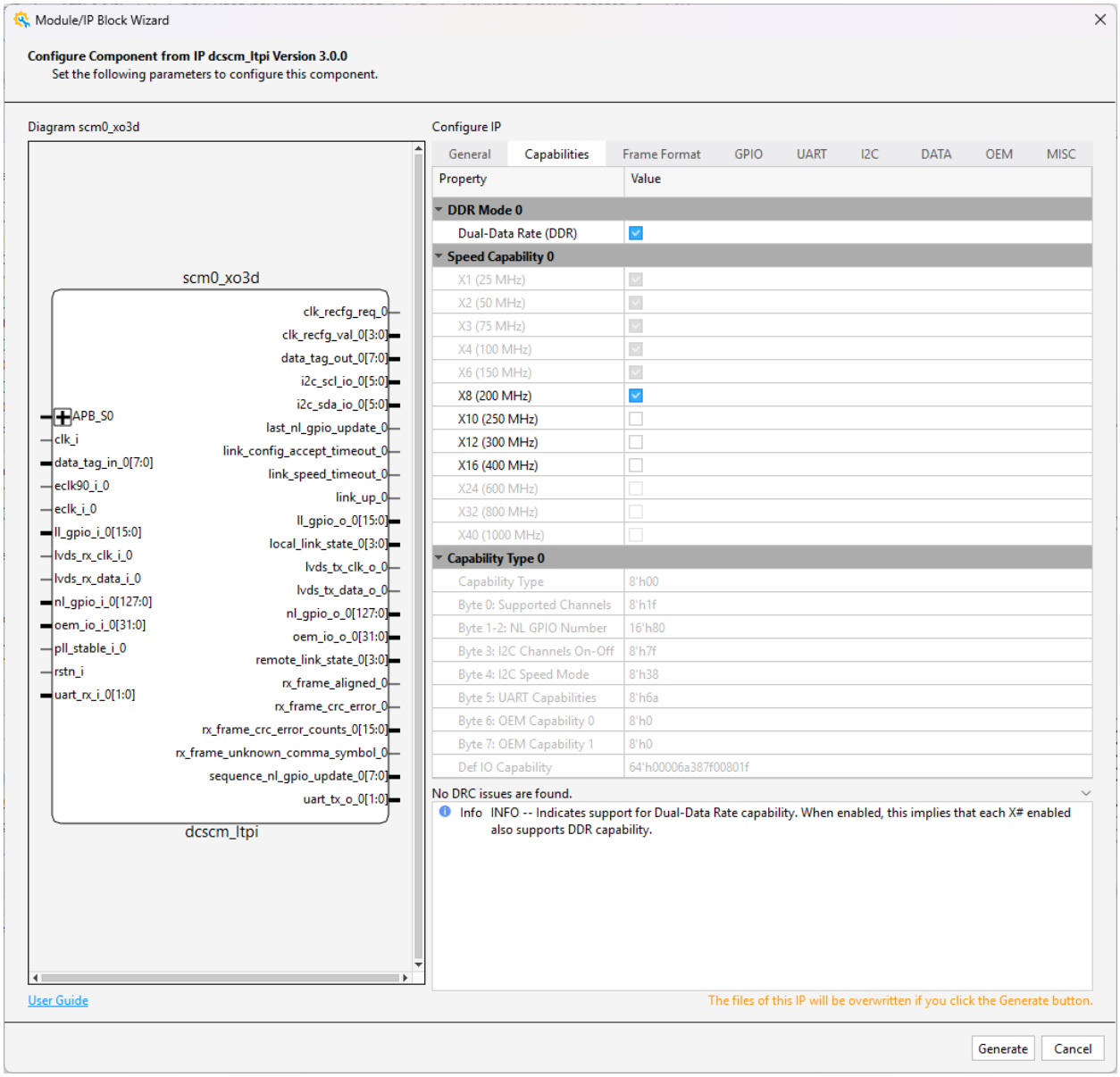

Figure 7.5. Capabilities Tab

Table 7.2. Capabilities Tab Parameter Descriptions

Parameter	Description
DDR Mode (node)	
Dual-Data Rate (DDR)	SPEED_CAP[15] = 1 when enabled. SPEED_CAP[15] = 0 when disabled.
Speed Capability (node)	
X1 (25 MHz)	This setting is always enabled. Mapped to SPEED_CAP[0].
X2 (50 MHz)	Mapped to SPEED_CAP[1].
X3 (75 MHz)	Mapped to SPEED_CAP[2]. Enabling this option also enables the previous option.
X4 (100 MHz)	Mapped to SPEED_CAP[3]. Enabling this option also enables the previous options.
X6 (150 MHz)	Mapped to SPEED_CAP[4]. Enabling this option also enables the previous options.
X8 (200 MHz)	Mapped to SPEED_CAP[5]. Enabling this option also enables the previous options.
X10 (250 MHz)	Mapped to SPEED_CAP[5]. Enabling this option also enables the previous options.
X12 (300 MHz)	Mapped to SPEED_CAP[5]. Enabling this option also enables the previous options.
X16 (400 MHz)	Mapped to SPEED_CAP[5]. Enabling this option also enables the previous options.
X24 (600 MHz)	Not available. Disabled by default.
X32 (800 MHz)	Not available. Disabled by default.
X40 (1000 MHz)	Not available. Disabled by default.
Capability Type (node)	
Capability Type	This displays the LTPI Frame setting (0x00 is default I/O).
Byte 0: Supported Channels	These are part of the default LTPI capabilities encoding as defined in the LTPI specifications.
Byte 1-2: NL GPIO Number	
Byte 3: Channels On-Off	
Byte 4: I2C Speed Mode	
Byte 5: UART Capabilities	
Byte 6: OEM Capability 0	
Byte 7: OEM Capability 1	
Def IO Capability	Displays the hex format of the capability frame.

7.1.3.3. Frame Format Tab

This section provides descriptions of the parameters in the Frame Format tab. By default, the parameters are not editable. They can only be modified by setting **Data Frame Type** to **OEM**. The default setting adheres to the default I/O frame format as defined in the LTPI specification.

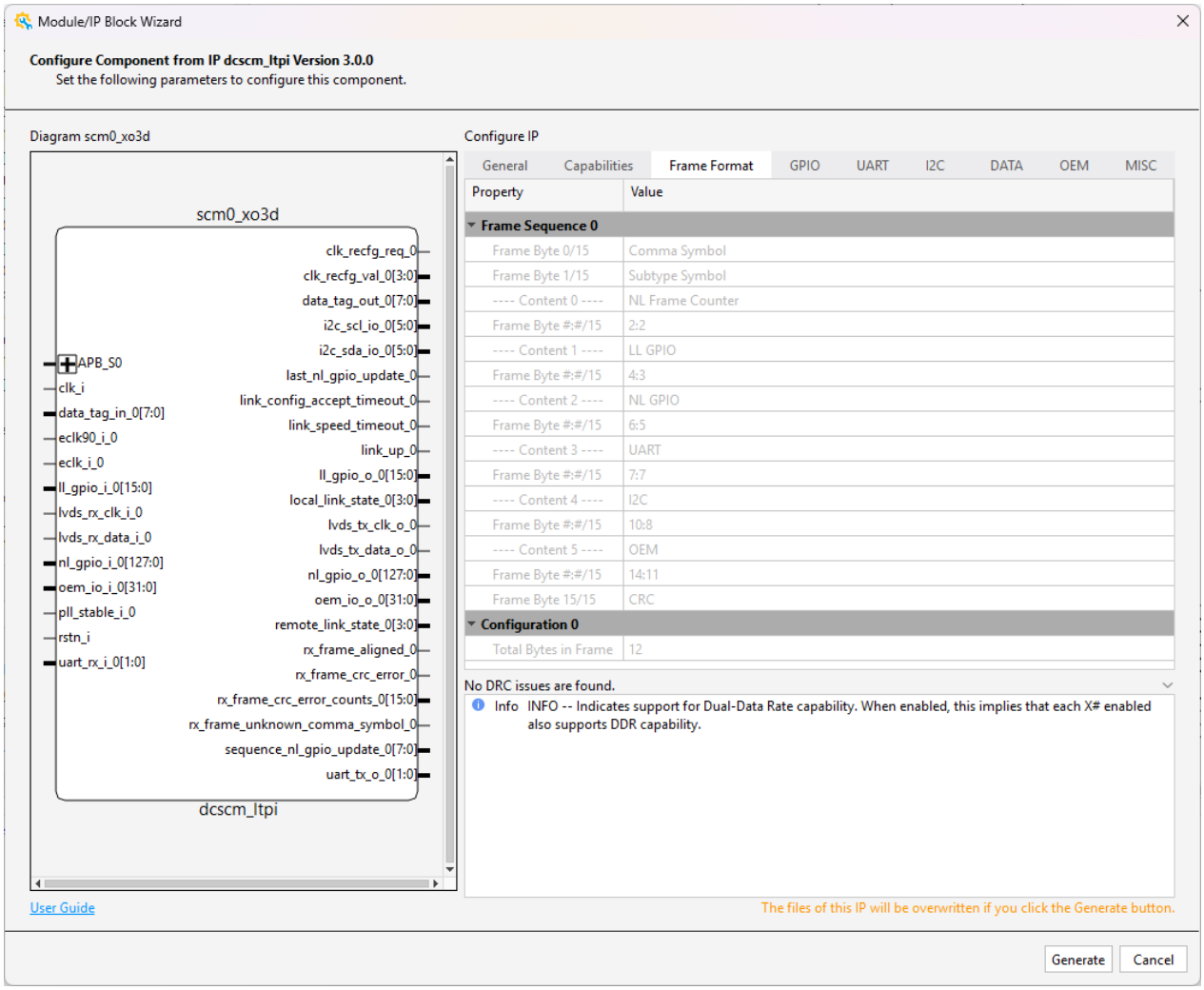

Figure 7.6. Frame Format Tab

7.1.3.4. GPIO Tab

This section provides descriptions of the parameters in the GPIO tab.

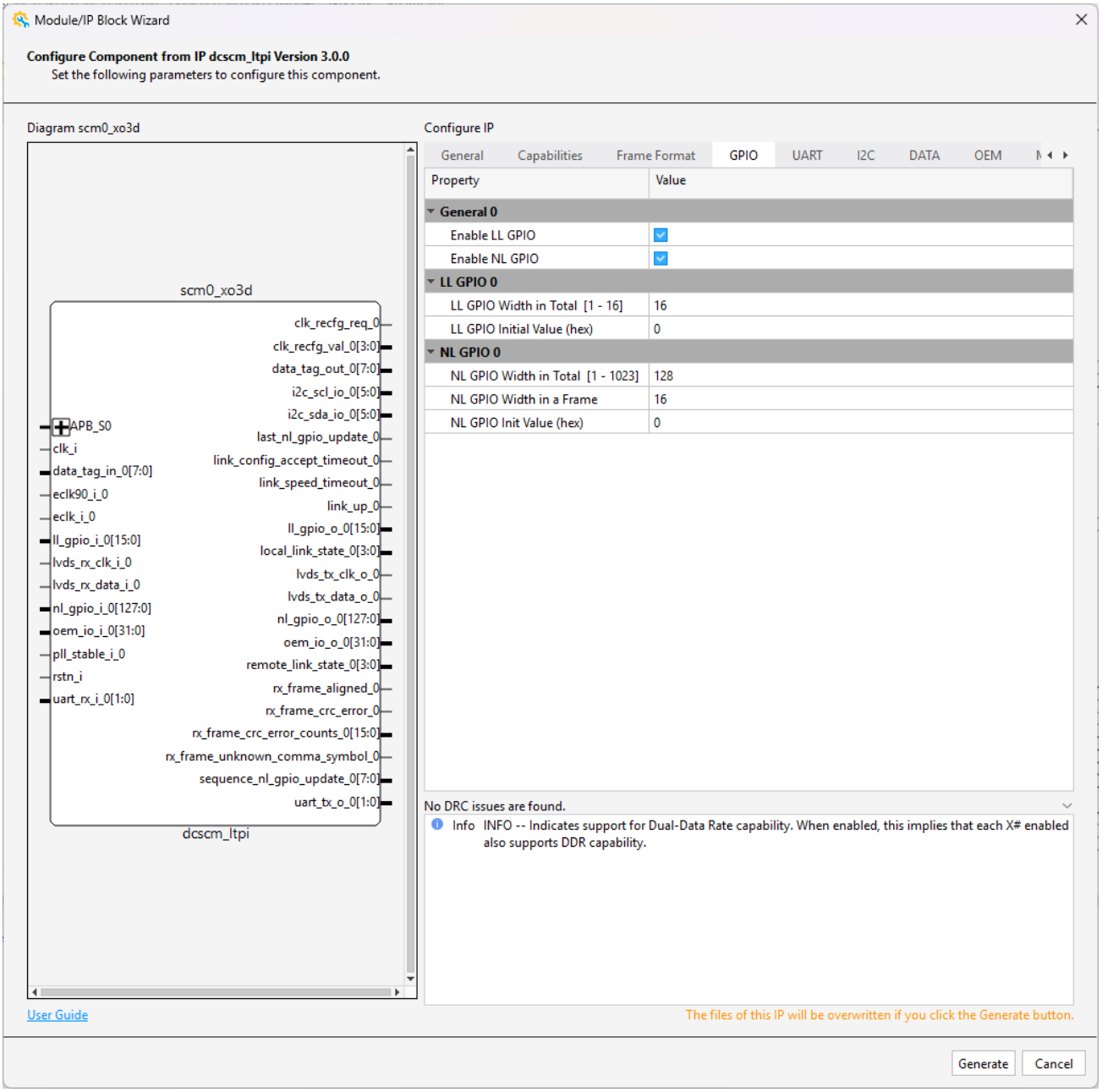

Figure 7.7. Channels Tab

Table 7.3. GPIO Tab Parameter Descriptions

Parameter	Description
General (node)	
Enable LL GPIO	Enables tunneling of Low Latency GPIO. When enabled, options for LL GPIO (node) are available.
Enable NL GPIO	Enables tunneling of Normal Latency GPIO. When enabled, options for NL GPIO (node) are available.
LL GPIO (node)	
LL GPIO Width in Total [1 - 16]	Determines the size of the Low Latency GPIO.
LL GPIO Initial Value (hex)	Determines the initial value for normal latency outputs during initial boot.
NL GPIO (node)	
NL GPIO Width in Total [1 - 1023]	Determines the size of the Normal Latency GPIO.
NL GPIO Width in a Frame	Determines the number of Normal Latency GPIO can be sent in a frame. This is 16 by default.
NL GPIO Init Value (hex)	Determines the initial value for normal latency outputs during initial boot.

7.1.3.5. UART Tab

This section provides descriptions of the parameters in the UART tab.

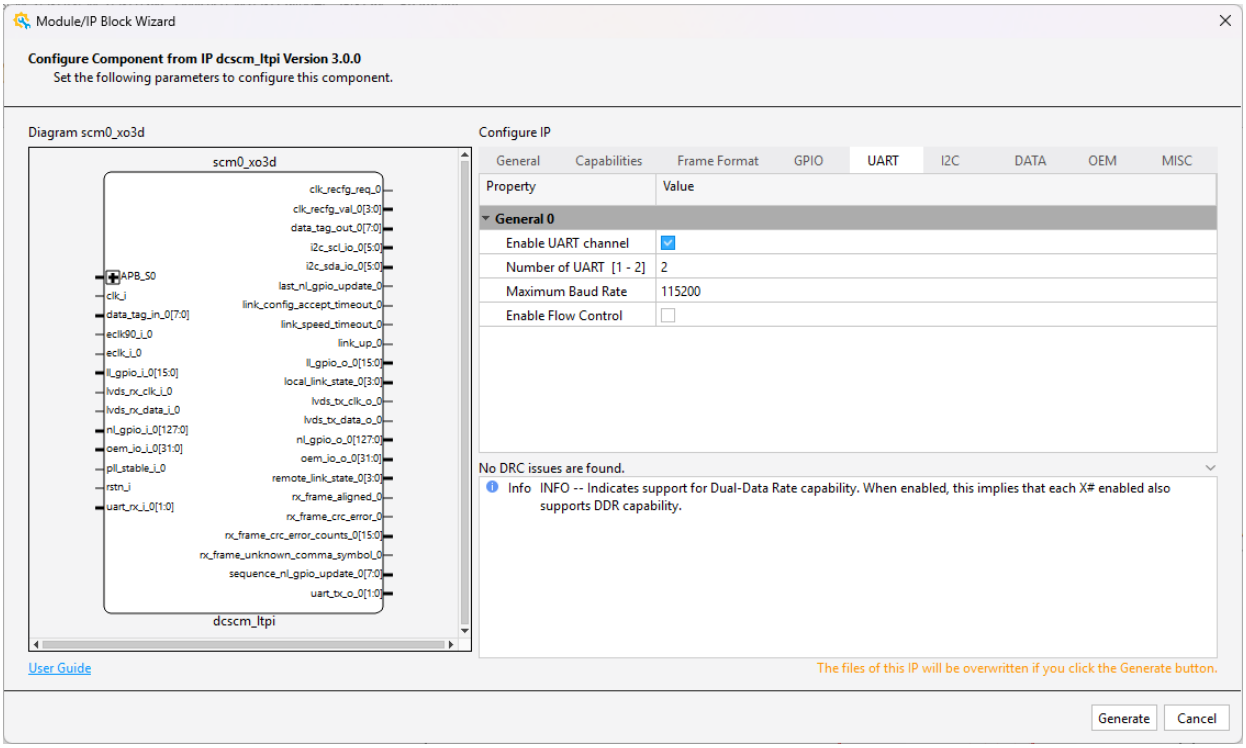

Figure 7.8. UART Tab

Table 7.4. UART Tab Parameter Descriptions

Parameter	Description
General (node)	
Enable UART Channel	Enables tunneling of UART channel. When enabled, additional options will be available
Number of UART [1 - 2]	Number of UART buses to be tunneled
Maximum Baud Rate	Determines the maximum baud rate that can be supported
Enable Flow Control	When enabled, UART flow control signals will be available as ports

7.1.3.6. I2C Tab

This section provides descriptions of the parameters in the I2C tab.

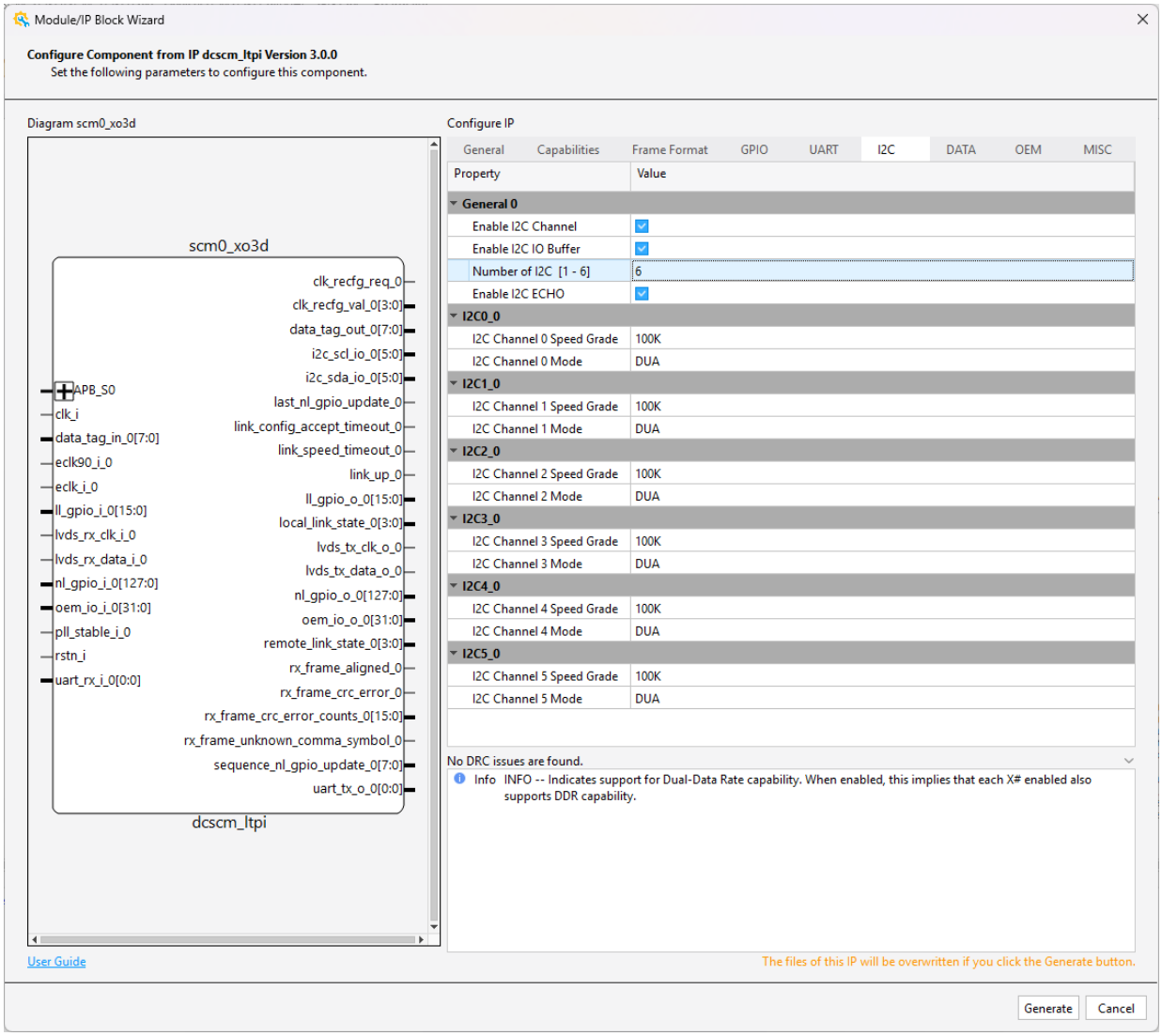

Figure 7.9. I2C Tab

Table 7.5. I2C Tab Parameter Descriptions

Parameter	Description
General (node)	
Enable I2C Channel	Enables tunneling of I2C Channel. When enabled, additional options are available.
Enable I2C IO Buffer	Enables tunneling of Normal Latency GPIO. When enabled, options for NL GPIO (node) are available.
Number of I2C [1 - 6]	Number of I2C buses to be tunneled. This determines the number of I2Cx_(node) options that are to be enabled.
Enable I2C ECHO	Disabling this option removes I2C ECHO from the I2C tunneling flow, improving the I2C duty cycle.
I2Cx_(node)	
I2C Channel x Speed Grade	Determines the I2C mode supported (Standard – 100 kHz or Fast – 400 kHz).
I2C Channel x Mode	Determines the device type that is to be tunneled (Controller, Target or both).

7.1.3.7. DATA Tab

This section provides descriptions of the parameters in the DATA tab.

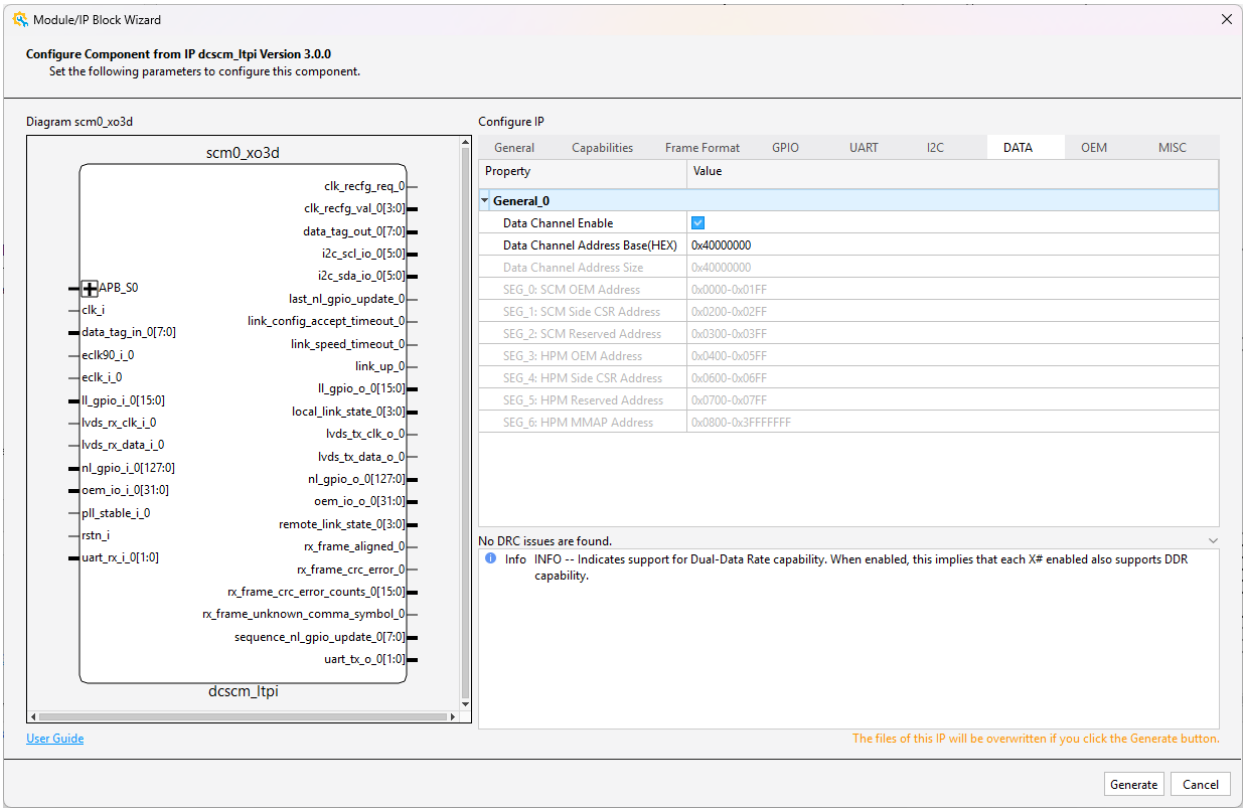

Figure 7.10. DATA Tab

Table 7.6. DATA Tab Parameter Descriptions

Parameter	Description
General_(node)	
CSR Port Outside	When enabled, the APB access to the CSR is external. HPM only.
Data Channel Enable	Enables tunneling of the Data Channel.
Data Channel Address Size	Address range of the data channel.
SEG_0: SCM OEM Address	This displays the address offset for each section of the data channel.
SEG_1: SCM Side CSR Address	
SEG_2: SCM Reserved Address	
SEG_3: HPM OEM Address	
SEG_4: HPM Side CSR Address	
SEG_5: HPM Reserved Address	
SEG_6: HPM MMAP Address	

7.1.3.8. OEM Tab

This section provides descriptions of the parameters in the OEM tab.

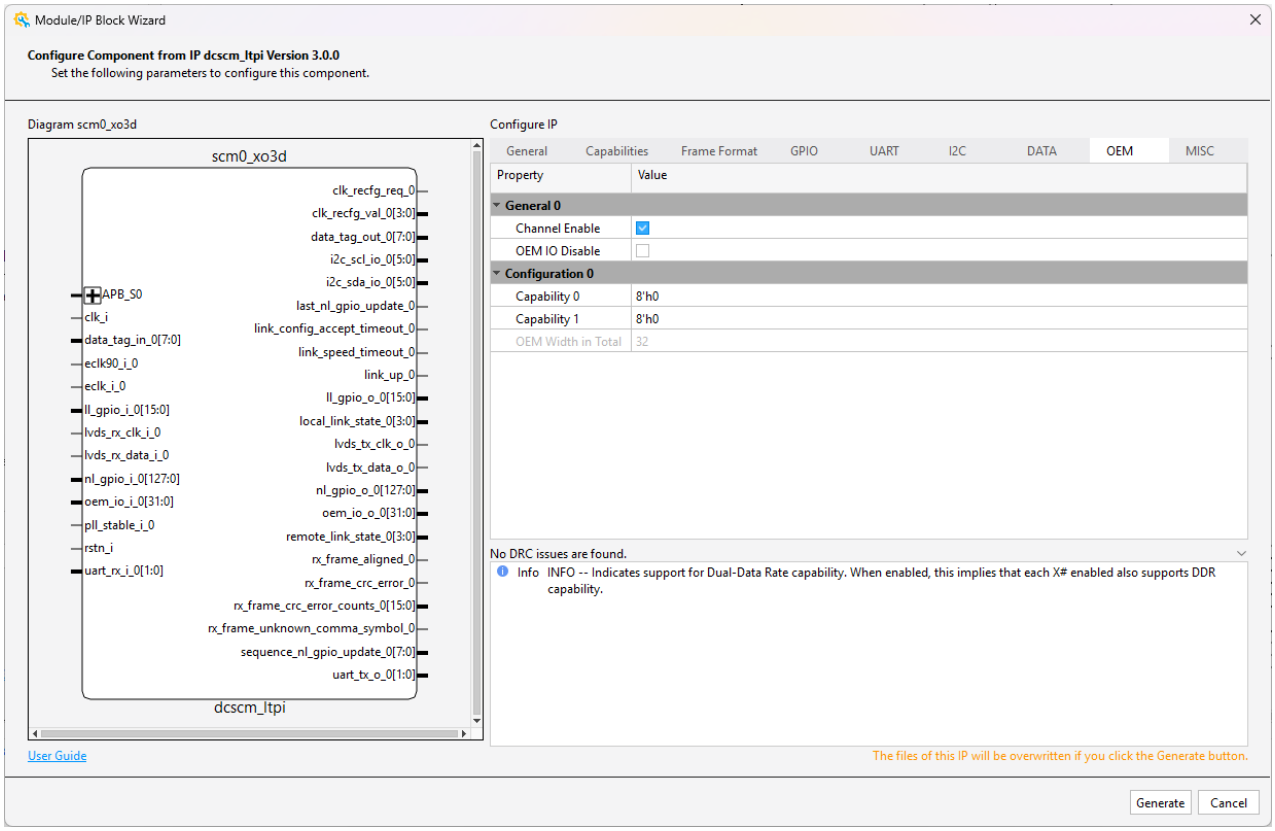

Figure 7.11. OEM Tab

Table 7.7. OEM Tab Parameter Descriptions

Parameter	Description
General (node)	
Channel Enable	When enabled, allows tunneling of OEM channel.
OEM IO Disable	When selected, user can use the OEM as additional GPIO, UART or I2C buses. This would generate additional options similar to the GPIO, UART and I2C options.
Configuration (node)	
Capability 0	OEM-defined capabilities
Capability 1	
OEM Width in Total	Number of OEM bits in the frame

7.1.3.9. MISC Tab

This section provides descriptions of the parameters in the MISC tab.

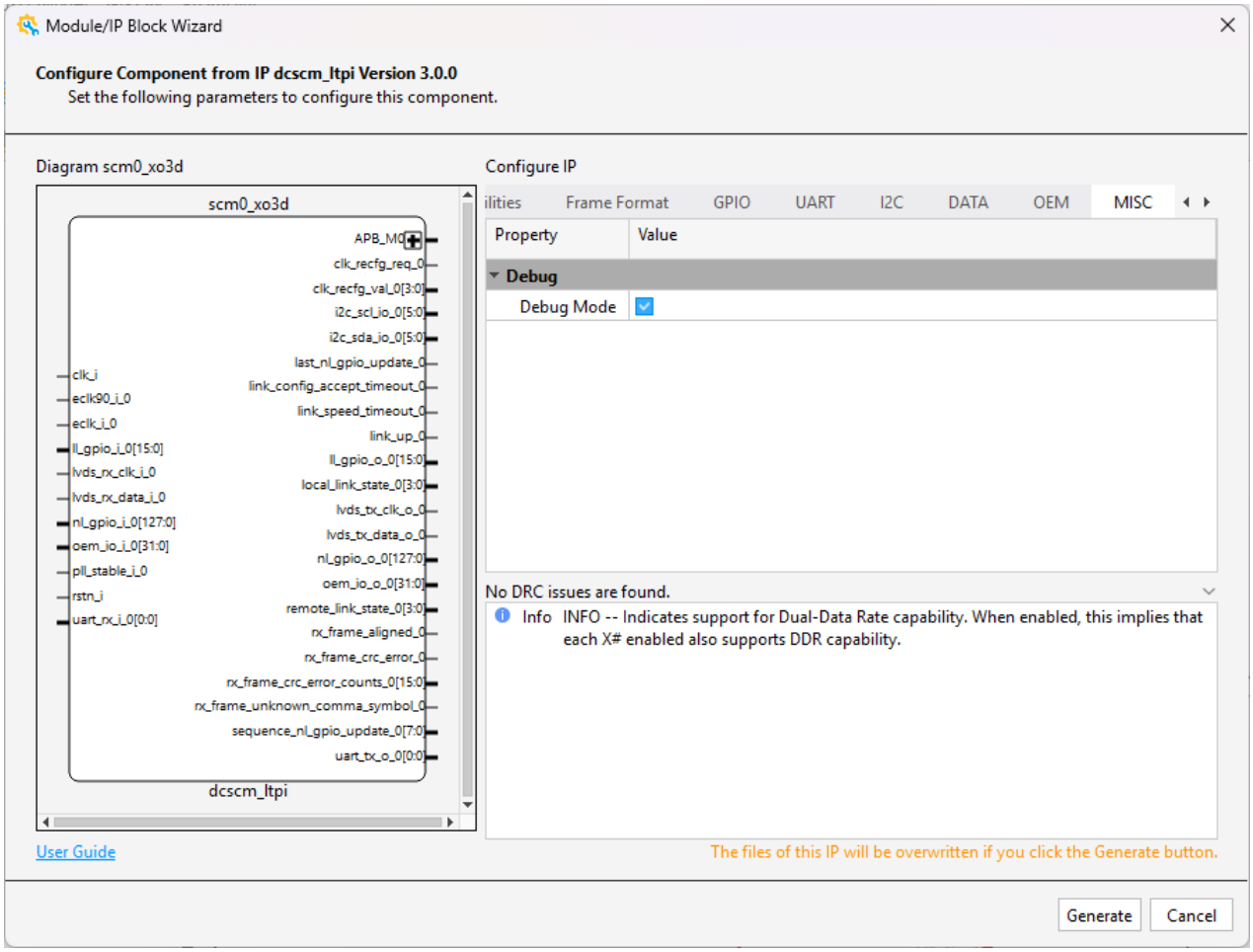

Figure 7.12. MISC Tab

Table 7.8. MISC Tab Parameter Descriptions

Parameter	Description
Debug	
Debug Mode	Enables various debug flags as ports. This is required to be enabled on the reference design

7.1.4. Adding a Custom RTL Module

This step can be used to add any of the following options:

- A placeholder RTL Module (only declarations of inputs and outputs).
- A wrapper module for RTL modules that is to be added through Lattice Diamond or Lattice Radiant.
- An existing RTL module.

1. Double-click or drag the **rtl** glue logic from the IP Catalog to the Schematic window.

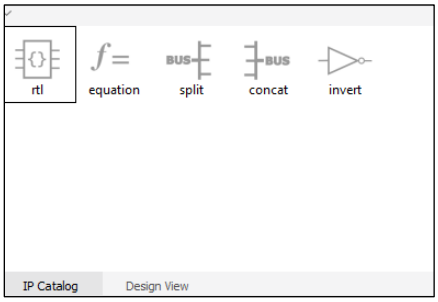

Figure 7.13. RTL Glue Logic

2. You can either place the RTL code in the **Rtl** box of the Glue Logic window and then click **OK.**

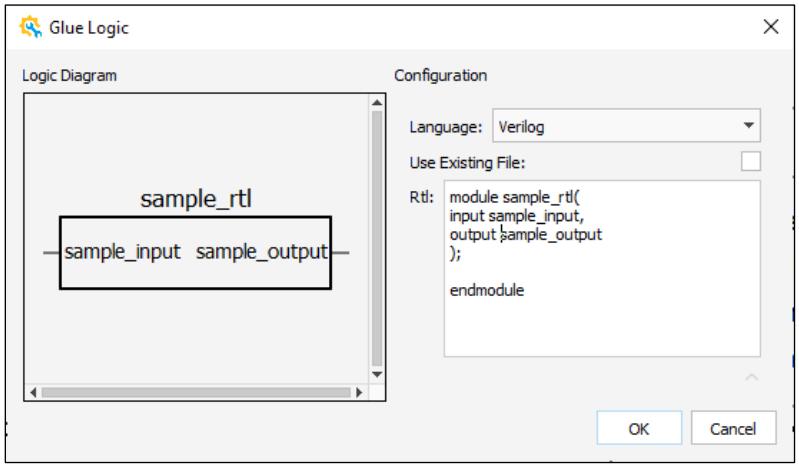

Figure 7.14. Using the RTL Box

3. Or, check the **Use Existing File** checkbox, link the file in **Path** field, and then click **OK**.

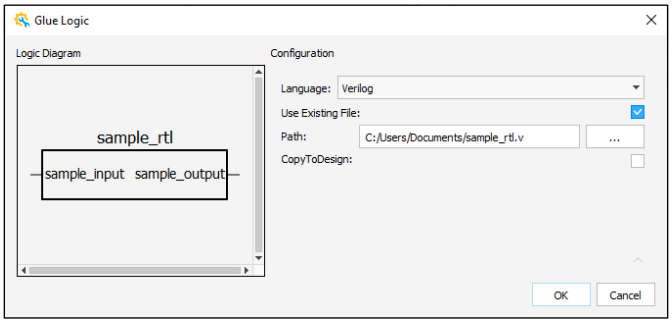

Figure 7.15. Using an Existing RTL File

4. You can either connect the module to existing modules, or connect it to a top-level module port. See the Add/Remove Top-Level Module Ports section for more details.

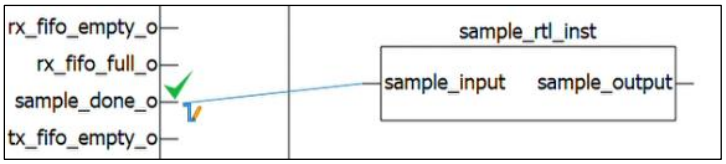

Figure 7.16. Connecting to Existing Modules

5. Click **Save**.

7.2. Lattice Diamond

This section discusses the changes that users can make to the reference design using Lattice Diamond. This section is only applicable to projects that use either MachXO3 or MachXO3D. For more information on using Lattice Diamond, refer to the Lattice Diamond software user guide.

7.2.1. Opening the Lattice Diamond Project

1. Unzip the reference design zip file.
2. Open Lattice Diamond 3.14. Note that the reference design has been tested using this version and does not guarantee backwards compatibility with previous Lattice Diamond versions.
3. Go to **File** > **Open** > **Project**.
4. Select the project file (*.ldf) found in *<Reference_Design_Folder>/Projects/<Design_Folder>*. In this section, the design used as an example is MachXO3D_SCM.

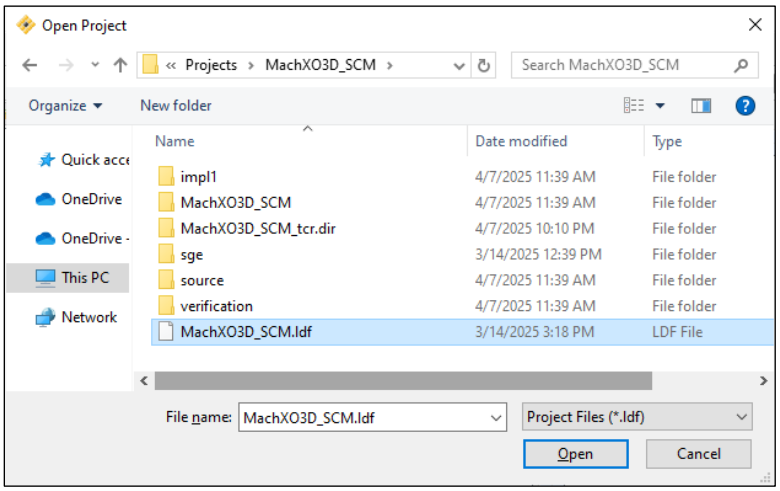

Figure 7.17. Selecting the Lattice Diamond Project

7.2.2. Changing the Target Device

This section explains how to change the project's target device. While it is possible to change to other device families using these steps, it is recommended to use the project dedicated to that device family and then use these steps to adjust the specifications for that device.

Note that changing to other device families may result in incompatibility with some modules in the design.

1. Double-click the current target device or right-click then select Edit.

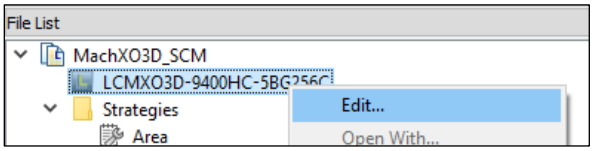

Figure 7.18. Selecting the Lattice Diamond Project

2. Choose the new target device.

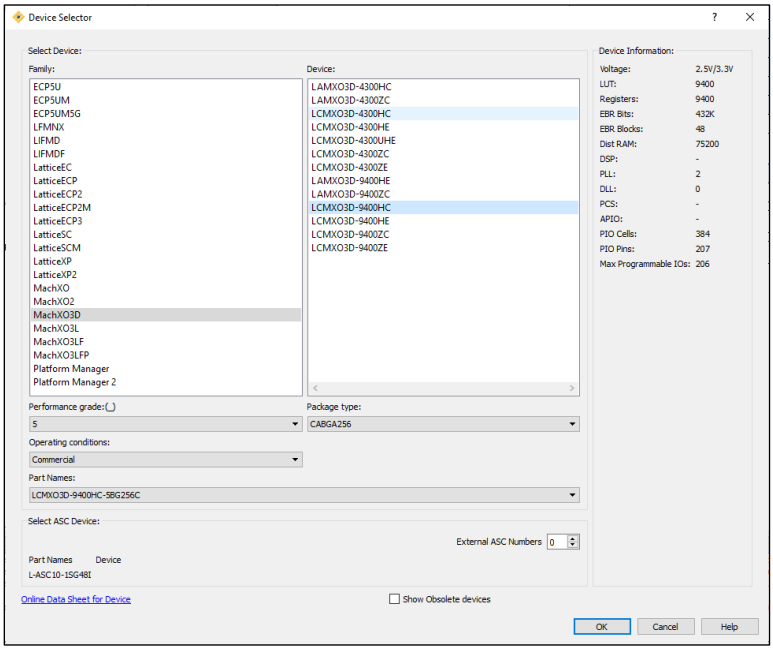

Figure 7.19. Device Selector Window

7.2.3. Adding Modules to the Design

Below are the steps for adding a new module to the design.

1. Drag the file from Windows Explorer to the File List Window or,

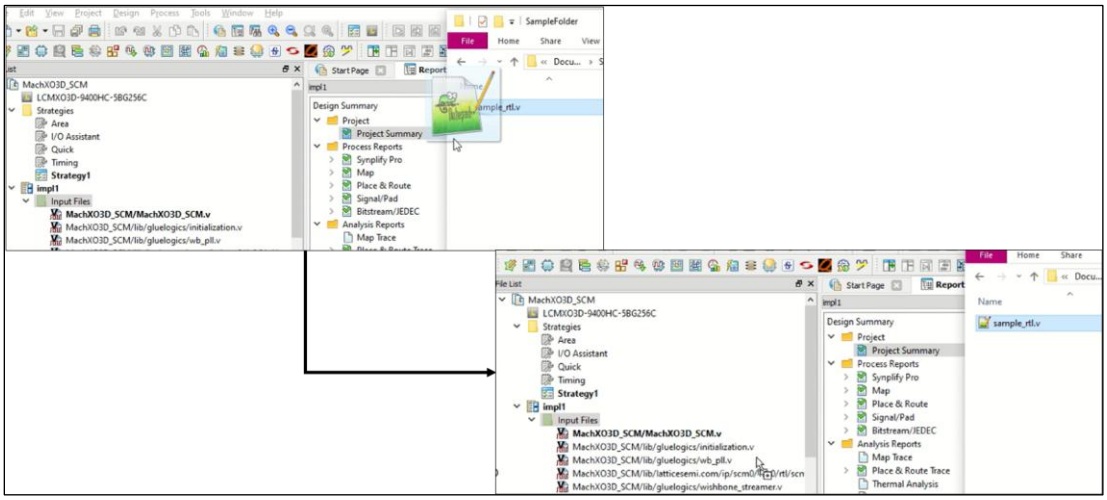

Figure 7.20. Adding a Module through Drag and Drop

2. Right-click the Input Files folder under the implementation, and select **Add** > **Existing File**.

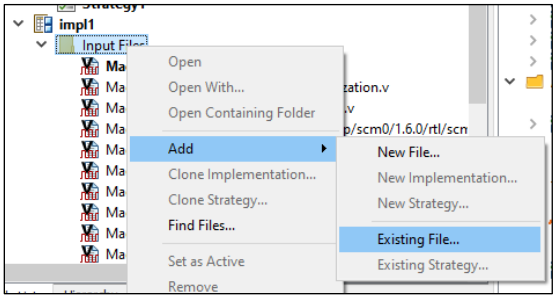

Figure 7.21. Selecting the Lattice Diamond Project

3. Select the file to be added and click **Add**.

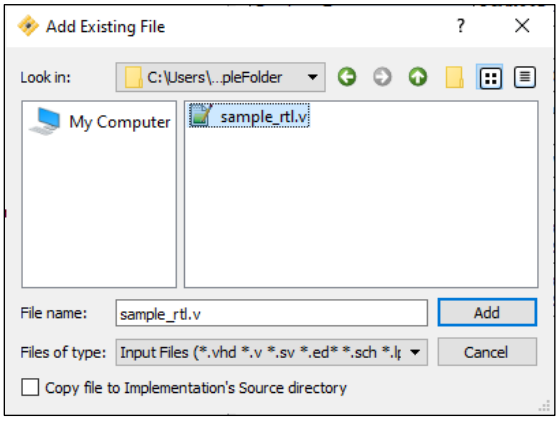

Figure 7.22. Add Existing File Window

Note that for these modules to be used in the design, any of the following must be met:

- The added module is instantiated by one of the existing modules. This is especially useful for modules that are instantiated by wrapper modules in the Lattice Propel project. This is the recommended method of adding modules. See Adding a Custom RTL Module section for instructions on how to add wrapper modules.
- The added module is manually instantiated into one of the existing modules. This requires modification of the existing modules. This method is not recommended as any changes to the top-level module may be overwritten by changes to the Lattice Propel project.

7.2.4. Updating the Pin Assignments

There are two ways to change the pin assignment:

- Changing the constraint file (*.lpf)
- Using spreadsheet view

7.2.4.1. Updating Using Constraint File

1. Open the LPF file found under LPF constraint files.

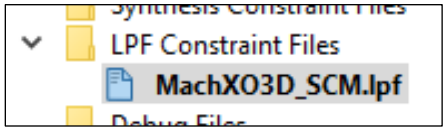

Figure 7.23. Constraint File (*.lpf)

2. Define or modify the I/O properties of the port.

```
IOBUF PORT "clk_i" IO_TYPE=LVCMOS33 PULLMODE=NONE ;
```

Figure 7.24. I/O Properties Example

3. Define or modify the location (pin assignment) of the port. For specific pin assignment requirements, refer to the Lattice FPGA Requirements section.

```
LOCATE COMP "clk_i" SITE "C8" ;
```

Figure 7.25. I/O Location Example

4. Save the file.

7.2.4.2. Updating Using Spreadsheet View

1. Run synthesis.

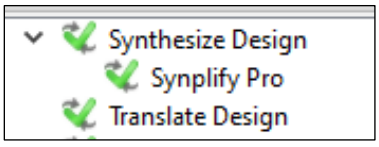

Figure 7.26. Lattice Diamond Synthesis

2. Open the spreadsheet view by clicking the **Spreadsheet View** icon or by clicking **Tools** > **Spreadsheet View**.

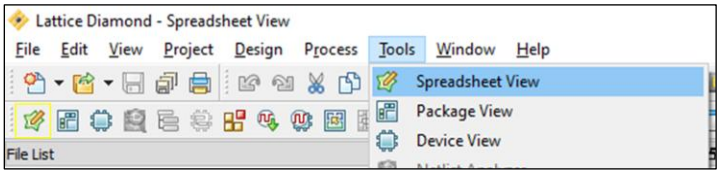

Figure 7.27. Open Spreadsheet View

3. Update the necessary properties. For specific pin assignment requirements, refer to the Lattice FPGA Requirements section.

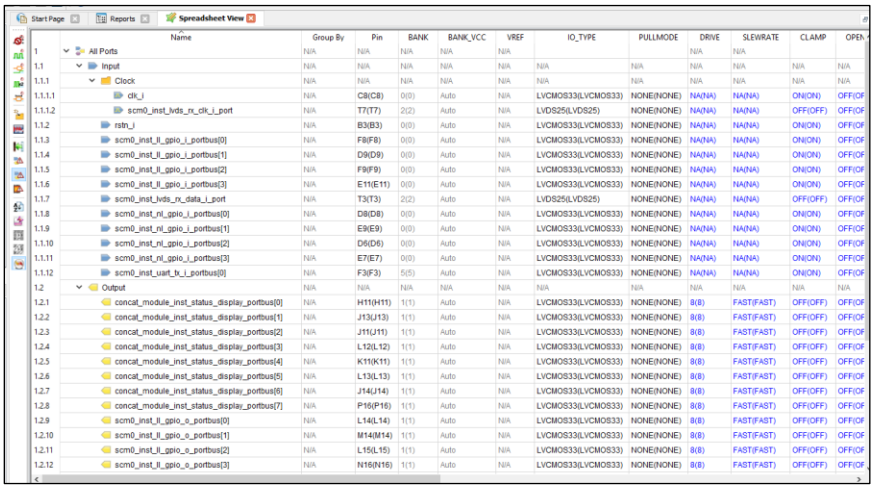

Figure 7.28. Spreadsheet View

4. Save the changes.

7.3. Lattice Radiant

This section discusses changes the user can make to the reference design using Lattice Radiant. This section is only applicable for the projects that use MachXO5-NX.

For more information on Lattice Radiant usage, refer to the Lattice Radiant software user guide.

7.3.1. Opening the Lattice Radiant Project

1. Unzip the reference design zip file.
2. Open Lattice Radiant 2024.2. Note that the reference design is tested using this version and does not guarantee backward compatibility with previous Lattice Radiant versions.
3. Go to **File** > **Open** > **Project**.
4. Select the project file (*.rdf) found in *<Reference_Design_Folder>/Projects/<Design_Folder>.* In this section, the design used as an example is **MachXO5NX_SCM**.

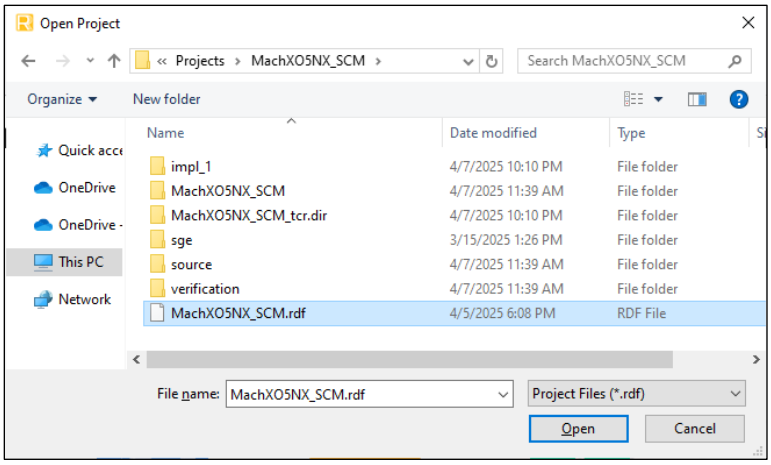

Figure 7.29. Selecting the Lattice Radiant Project

7.3.2. Adding Modules to the Design

Below are the steps for adding a new module to the design:

1. Drag the file from Windows Explorer to the File List window or,

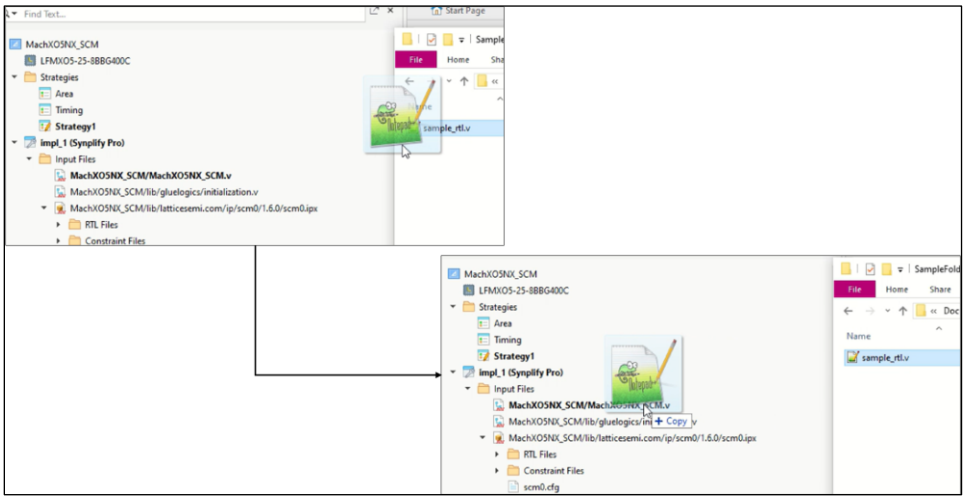

Figure 7.30. Adding a Module through Drag and Drop

2. Right-click the input files folder under the implementation, and select **Add** > **Existing File**.

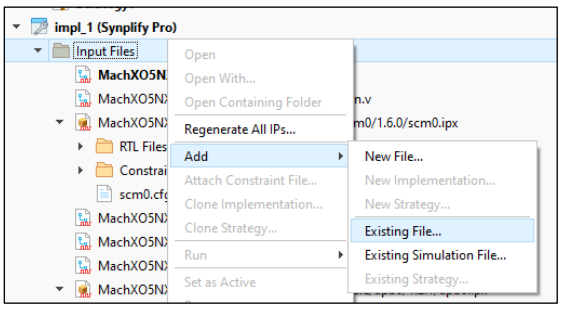

Figure 7.31. Selecting the Lattice Diamond Project

3. Select the file to be added and click **Add**.

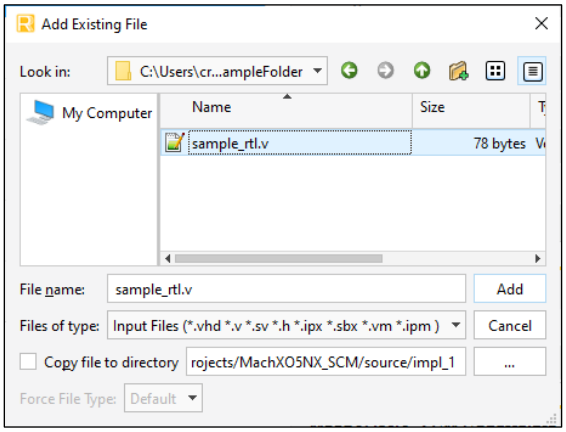

Figure 7.32. Add Existing File Window

Note that for these modules to be used in the design, any of the following conditions must be met:

- The added module is instantiated by one of the existing modules. This is especially useful for modules that are instantiated by wrapper modules in the Lattice Propel project. This is the recommended method of adding modules. See the Adding a Custom RTL Module section on how to add wrapper modules.
- The added module is manually instantiated into one of the existing modules. This requires modification of the existing modules. This method is not recommended as any changes to the top-level module may be overwritten by changes to the Lattice Propel project.

7.3.3. Updating the Pin Assignments

There are two ways to change the pin assignment:

- Changing the constraint file (*.pdc)
- Using the device constraint editor

7.3.3.1. Updating Using Constraint File

1. Open the *.pdc file found under **Post-Synthesis Constraint Files**.

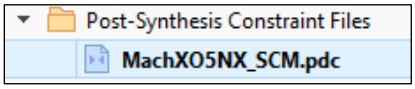

Figure 7.33. Lattice Radiant Constraint File (*.pdc)

2. Define or modify the I/O properties of the port.

```
ldc_set_port -iobuf {IO_TYPE=LVCMOS33 PULLMODE=NONE} [get_ports clk_i]
```

Figure 7.34. Lattice Radiant I/O Properties Example

3. Define or modify the location (pin assignment) of the port. For specific pin assignment requirements, refer to the Lattice FPGA Requirements section.

```
ldc_set_location -site {V1} [get_ports clk_i]
```

Figure 7.35. Lattice Radiant I/O Location Example

4. Save the file.

7.3.3.2. Updating Using the Device Constraint Editor

1. Run synthesis.

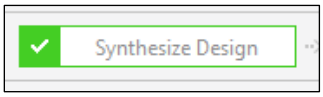

Figure 7.36. Lattice Radiant Synthesis

2. Open the device constraint editor by clicking the **Device Constraint Editor** icon or by clicking **Tools** > **Device Constraint Editor**.

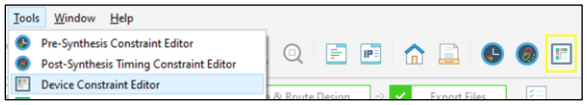

Figure 7.37. Open Device Constraint Editor

3. Update the necessary properties. For specific pin assignment requirements, refer to the Lattice FPGA Requirements section.

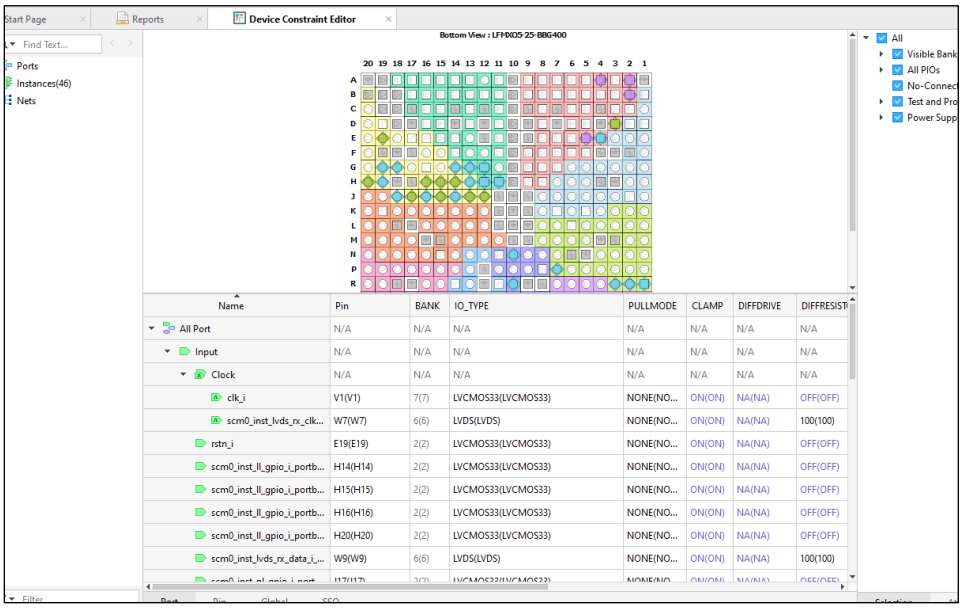

Figure 7.38. Device Constraint Editor

4. Save the changes.

8. Simulating the Reference Design

8.1. Setting Up the Simulation

To simulate using the default configurations of the designs, perform the following steps:

1. Unzip the reference design zip file.
2. Open the simulation **do** file to be used. For this walkthrough, **SCM_XO3D_HPM_XO3D.do** will be used. Note that the succeeding steps apply regardless of the chosen **do** file.
3. On line 2 of the **do** file, replace the default path with the simulation folder path.

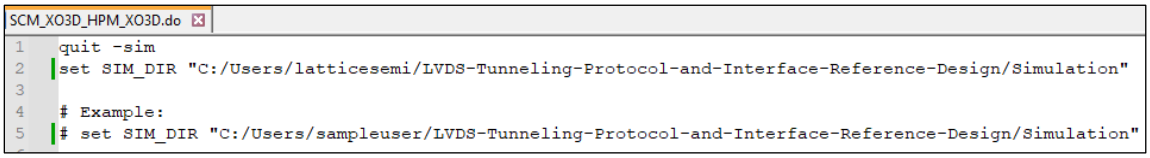

Figure 8.1. Simulation Do File

4. Click **Save**.
5. Open the simulator tool. For this walkthrough, **Questa Lattice OEM Edition-64 2024.2** will be used.
6. Go to **Tools** > **Td** > **Execute Macro**.

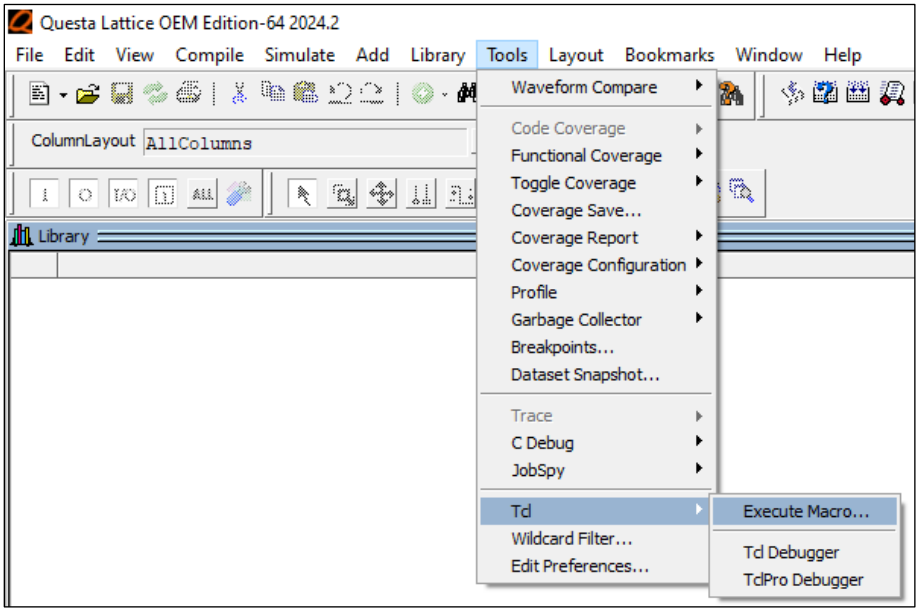

Figure 8.2. Execute Macro

7. Open the simulation **do** file. Close and overwrite any existing projects when prompted.

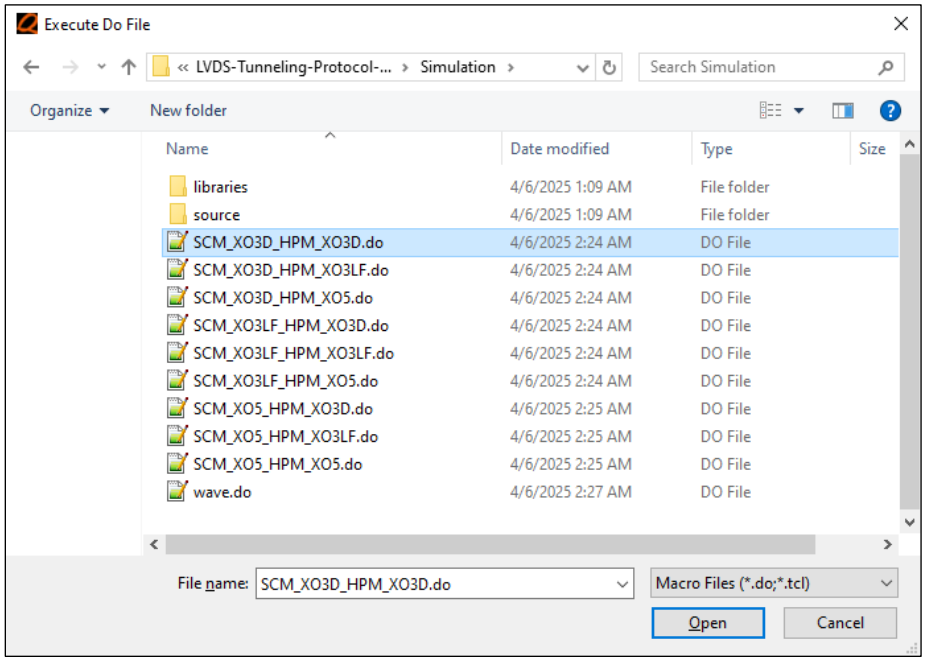

Figure 8.3. Execute Macro

8.2. Simulation Details and Results

The simulation is executed in six parts: Link Training, GPIO Test, OEM Test, UART Test, I2C Test, and CSR Test.

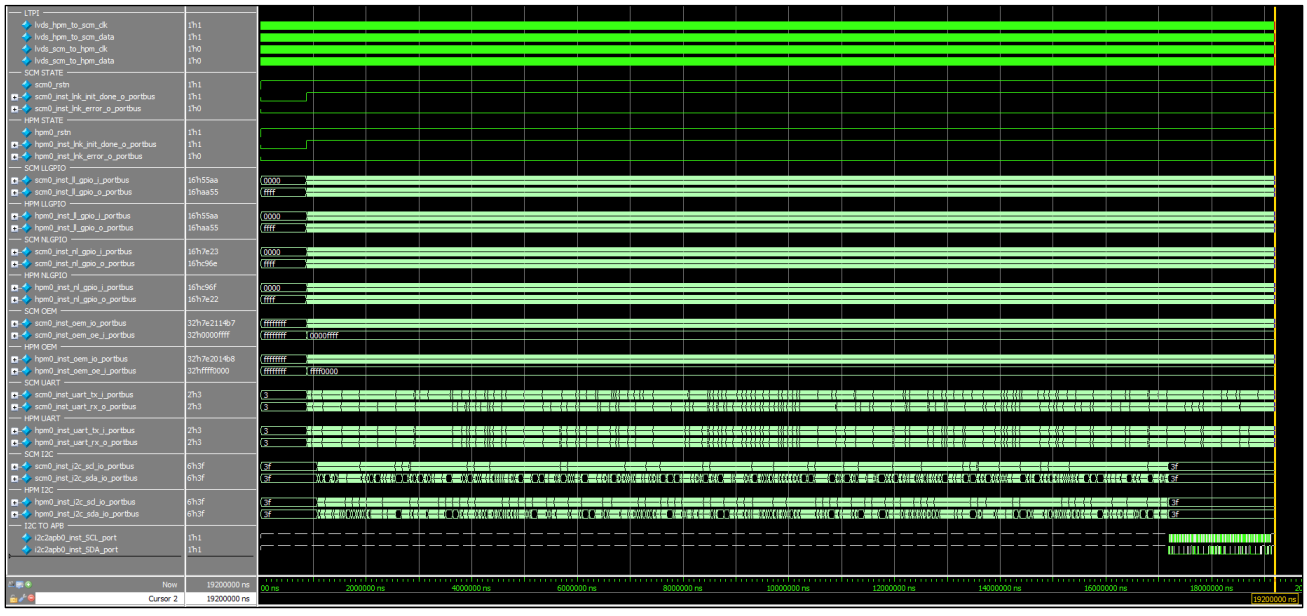

Figure 8.4. Simulation Wave View

8.2.1. Link Training

1. After the reset is released at 100 ns, the LTPI connection between the SCM and the HPM starts transmitting data at 25 MHz SDR (LTPI Base Speed).

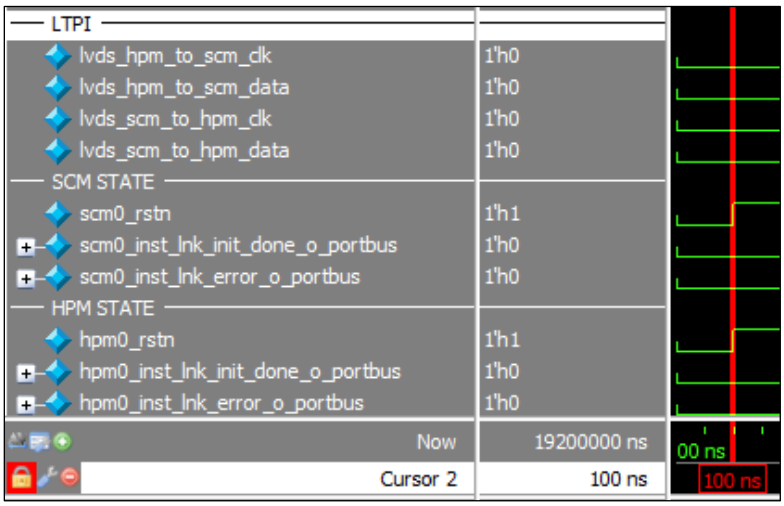

Figure 8.5. Reset Release

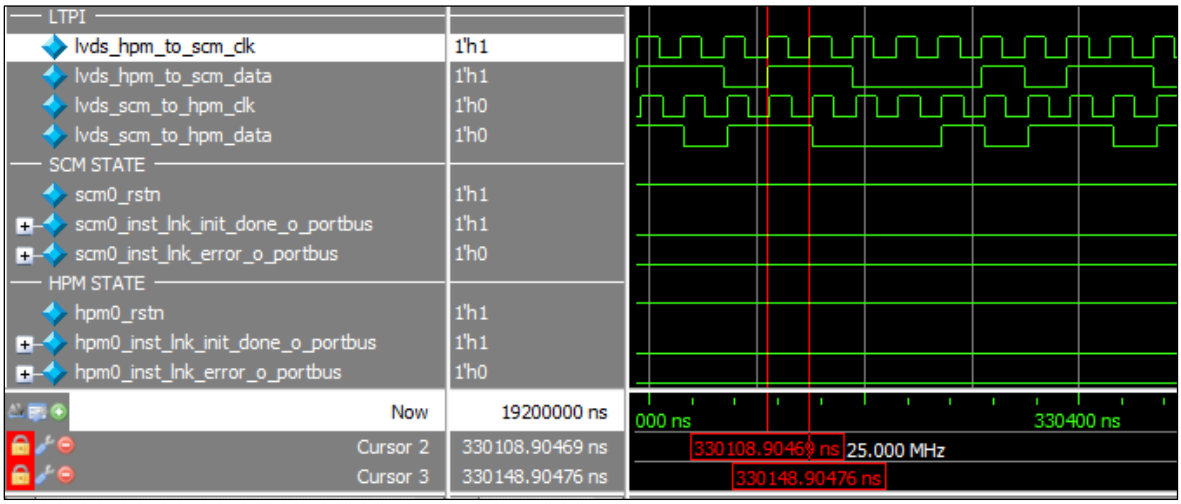

Figure 8.6. LTPI Base Speed

2. Once the Base Speed clock is stable, the system enters the Link Detect state. The transitions between the internal states are not explicitly shown in the simulation and are only observable through CSR access.
3. The transition between Base Speed and Target Speed is observable in the simulation. The default Target Speed of the reference design is 200 MHz DDR.

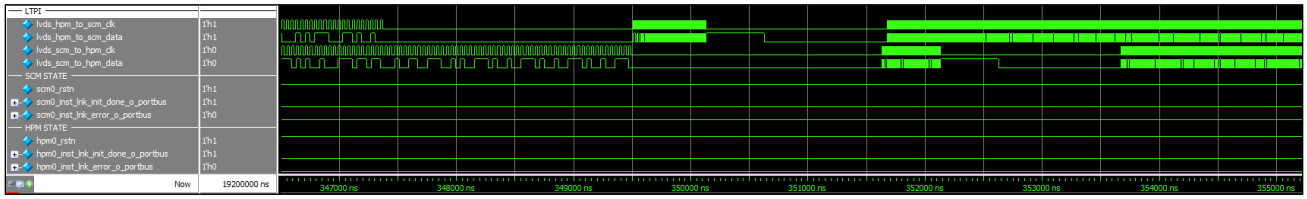

Figure 8.7. Base Speed to Target Speed Transition

4. Both SCM and HPM realign their PHY using the new clock frequency and go through the Advertise, Configuration (SCM only), Accept (HPM only) and Active states.
5. A message is printed on the simulator console once the link is established (scm0_inst_lnk_init_done_o_portbus= 1 and hpm0_inst_lnk_init_done_o_portbus = 1).

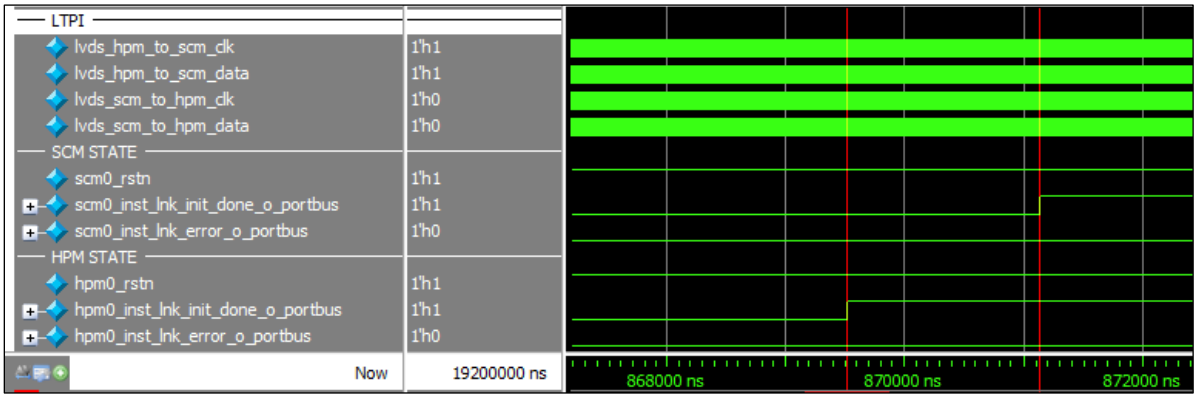

Figure 8.8. Link Training Done

```
2905295ns  : ########## Link Established ##########
```

Figure 8.9. Link Established Message

8.2.2. GPIO Test

This test evaluates both Low Latency GPIO and Normal Latency GPIO, in the following order:

1. The Low Latency Input of SCM is set to 0xAA55.
2. The test bench checks for changes in the Low Latency Output of HPM. If the change matches the set value of 0xAA55, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
3. The Low Latency Input of HPM is set to 0x55AA.
4. The test bench checks for changes in the Low Latency Output of SCM. If the change matches the set value of 0x55AA, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
5. The Normal Latency Input of SCM is set to 0x5A5A.
6. The test bench checks for changes in the Normal Latency Output of HPM. If the change matches the set value of 0x5A5A, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
7. The Normal Latency Input of HPM is set to 0xA5A5.
8. The test bench checks for changes in the Normal Latency Output of SCM. If the change matches the set value of 0xA5A5, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.

After step 8, a message is printed to indicate that the GPIO test is complete. The test bench continues to stimulate all Low Latency and Normal Latency inputs by inversion and increments of one, respectively. However, all outputs are no longer checked by the test bench. This continuous activity helps simulate active LLGPIO and NLGPIO channels while the next channels are being tested.

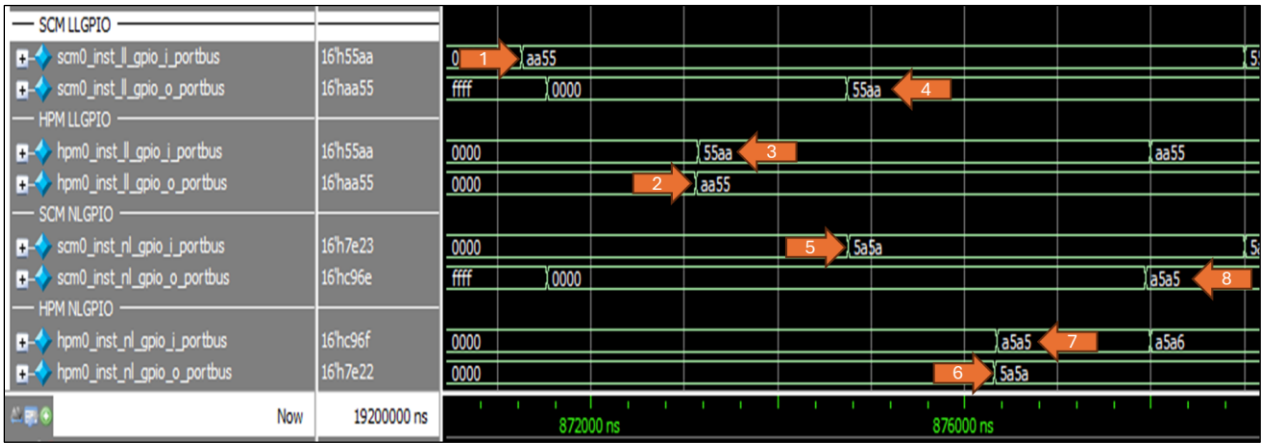

Figure 8.10. GPIO Test Flow

```
871243ns  : <<<<<GPIO Test - START>>>>>


871253ns  : SCM to HPM LLGPIO Test - START
871258ns  : Setting SCM LLGPIO Input to 0xAA55
873135ns  : HPM LLGPIO Output received 0xaa55
873140ns  : SCM to HPM LLGPIO Test - Passed
873140ns  : SCM to HPM LLGPIO Test - END


873150ns  : HPM to SCM LLGPIO Test - START
873155ns  : Setting SCM LLGPIO Input to 0x55AA
874743ns  : SCM LLGPIO Output received 0x55aa
874748ns  : HPM to SCM LLGPIO Test - Passed
874748ns  : HPM to SCM LLGPIO Test - END


874758ns  : SCM to HPM NLGPIO Test - START
874763ns  : Setting SCM NLGPIO Input to 0x5A5A
876335ns  : HPM NLGPIO Output received 0x5a5a
876340ns  : SCM to HPM NLGPIO Test - Passed
876340ns  : SCM to HPM NLGPIO Test - END


876350ns  : HPM to SCM NLGPIO Test - START
876355ns  : Setting SCM NLGPIO Input to 0xA5A5
877943ns  : SCM NLGPIO Output received 0xa5a5
877948ns  : HPM to SCM NLGPIO Test - Passed
877948ns  : HPM to SCM NLGPIO Test - END


877958ns  : GPIO Test - Passed
```

Figure 8.11. GPIO Test Console Messages

8.2.3. OEM Test

This test evaluates the OEM channel in the following order:

1. SCM is set as input and HPM as output.
2. Fixed data (0x5A5A0F0F) is transmitted through SCM OEM.
3. The test bench checks for changes in HPM OEM. If the change matches the set value of 0x5A5A0F0F, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
4. HPM is set as input and SCM as output.
5. Fixed data (0xA5A5F0F0) is transmitted through HPM OEM.
6. The test bench checks for changes in SCM OEM. If the change matches the set value of 0xA5A5F0F0, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.

After step 6, a message is printed to indicate that the GPIO test is complete. The test bench continues to stimulate the upper 16 bits of SCM OEM and lower 16 bits of HPM OEM by increments of one. However, all outputs are no longer checked by the test bench. This continuous activity helps simulate an active OEM channel while the next channels are being tested.

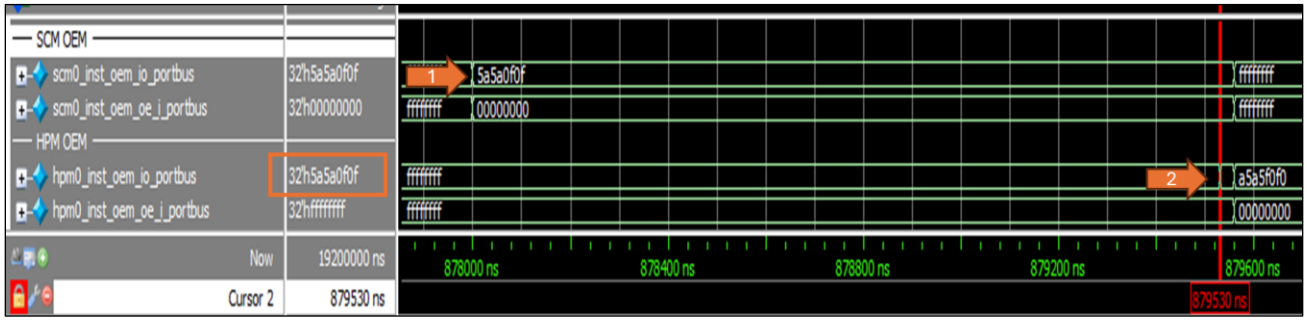

Figure 8.12. OEM Test Flow (SCM to HPM)

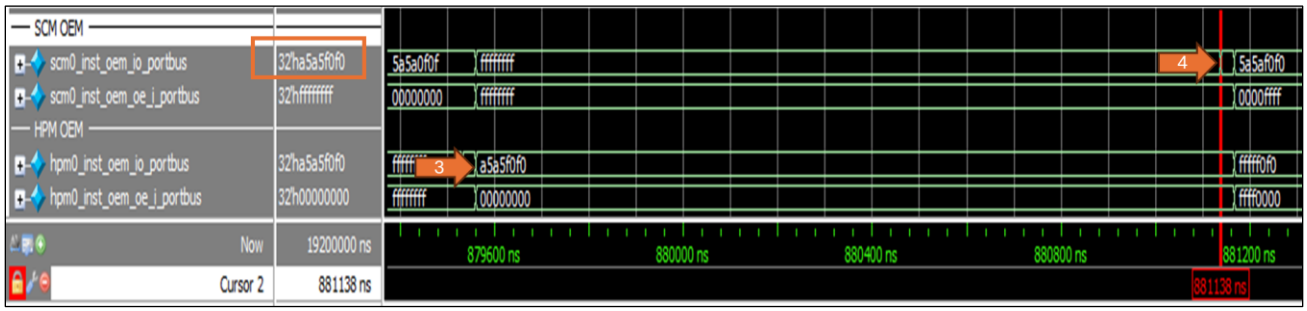

Figure 8.13. OEM Test Flow (SCM to HPM)

```
877978ns  : <<<<<OEM Test - START>>>>>


877988ns  : SCM to HPM OEM Test - START
877993ns  : Setting SCM OEM as input and HPM OEM as output
877998ns  : Transmitting 0x5a5a0f0f
879535ns  : HPM OEM received 0x5a5a0f0f
879540ns  : SCM to HPM OEM Test - Passed
879540ns  : SCM to HPM OEM Test - END


879550ns  : HPM to SCM OEM Test - START
879555ns  : Setting HPM OEM as input and SCM OEM as output
879560ns  : Transmitting 0xa5a5f0f0
881143ns  : SCM OEM received 0xa5a5f0f0
881148ns  : HPM to SCM OEM Test - Passed
881148ns  : HPM to SCM OEM Test - END


881158ns  : OEM Test - Passed


881168ns  : <<<<<OEM Test - END>>>>>
```

Figure 8.14. OEM Test Console Messages

8.2.4. UART Test

This test evaluates the UART channel at a baud rate of 115,200.

1. Fixed data (0xA5) is sent through the Tx port of SCM Bus 0.
2. On the HPM side, the Tx and Rx are tied together, transmitting the exact data received from SCM back to the source.
3. Data is checked at the Rx port of SCM Bus 0. If the received data matches the transmitted data, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
4. Steps 1–3 are repeated for Bus 1, sending 0x5A instead.

After completion, a message is printed to indicate that the UART test is complete. The test bench continues to stimulate the Tx ports of SCM Bus 0 and Bus 1 by increments of one. However, the Rx ports of SCM Bus 0 and Bus 1 are no longer checked by the test bench. This continuous activity helps simulate an active UART channel while the next channel is being tested.

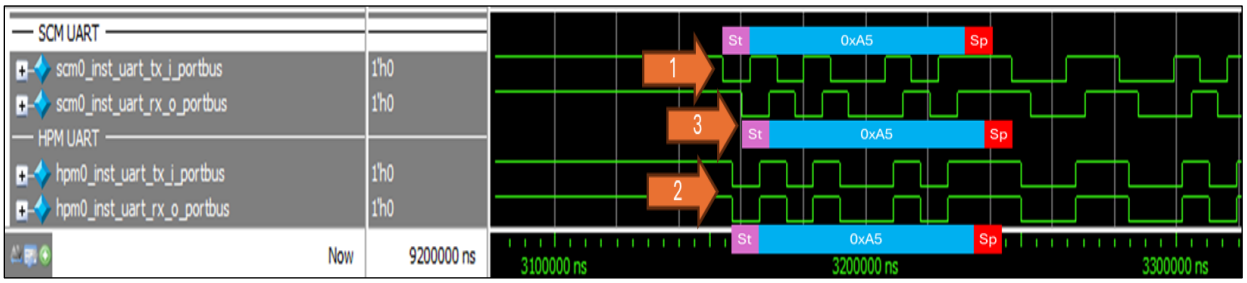

Figure 8.15. UART Test Flow

```
 881178ns  : <<<<<UART Test - START>>>>>


 881188ns  : Sending 0xA5 on Bus 0
 971930ns  : SCM UART Bus 0 RX received 0xa5
 971935ns  : UART Bus 0 Test - Passed


 971945ns  : Sending 0x5A on Bus 1
1062580ns  : SCM UART Bus 1 RX received 0x5a
1062585ns  : UART Bus 1 Test - Passed


1062595ns  : UART Test - Passed
```

Figure 8.16. UART Test Console Messages

8.2.5. I2C Test

This test evaluates the I2C channel in standard mode (100 kHz). The channels are configured such that an external I2C controller is connected to SCM, while an external I2C target is connected to HPM. The external I2C target has a memory block for simulating the writing and reading of internal registers through I2C.

1. The external controller sends an initial 32-bit read command to address 0x00. The contents of this address are irrelevant and this step is performed only to show the initial status of the target register.
2. The external controller sends a 32-bit write command to address 0x00 with a fixed data.
3. To check if the write was successful, another 32-bit read command is sent to address 0x00. If the read data matches the written data, a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
4. Steps 1–3 are repeated for Bus 1 through Bus 5. Each previously tested I2C bus is idle.

After step 4, a message is printed to indicate that the I2C test is complete. The I2C channels are no longer stimulated by the test bench and remain in a pull-up state.

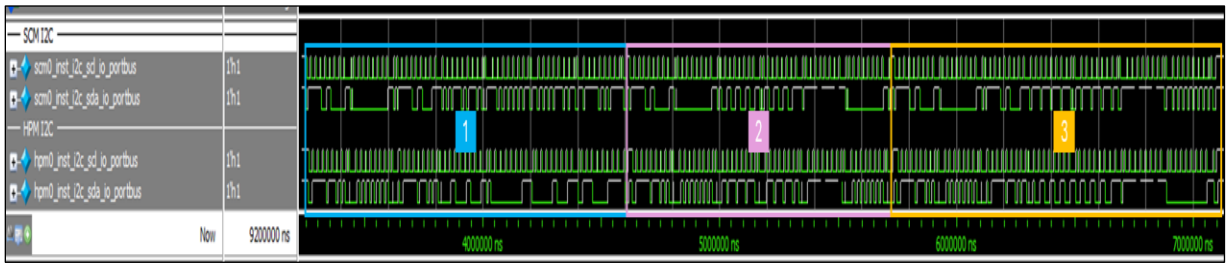

Figure 8.17. I2C Test Flow

```
 1062615ns  : <<<<<I2C Test - START>>>>>


 1062625ns  : I2C Bus 0 Initial 32-bit Data Read from Address 0x00
 2012453ns  : I2C Bus 0 Received Data is 0x24810963
 2012458ns  : I2C Bus 0 Write Data 0xAA55FF00 to Address 0x00
 2826055ns  : I2C Bus 0 Read Data from Address 0x00
 3776453ns  : I2C Bus 0 Received Data is 0xaa55ff00
 3776458ns  : I2C Bus 0 Test - Passed


 3776468ns  : I2C Bus 1 Initial 32-bit Data Read from Address 0x00
 4722453ns  : I2C Bus 1 Received Data is 0x88ae9b92
 4722458ns  : I2C Bus 1 Write Data 0x55FF00AA to Address 0x00
 5536055ns  : I2C Bus 1 Read Data from Address 0x00
 6486453ns  : I2C Bus 1 Received Data is 0x55ff00aa
 6486458ns  : I2C Bus 1 Test - Passed


 6486468ns  : I2C Bus 2 Initial 32-bit Data Read from Address 0x00
 7416453ns  : I2C Bus 2 Received Data is 0x4ab91313
 7416458ns  : I2C Bus 2 Write Data 0xFF00AA55 to Address 0x00
 8226055ns  : I2C Bus 2 Read Data from Address 0x00
 9160453ns  : I2C Bus 2 Received Data is 0xff00aa55
 9160458ns  : I2C Bus 2 Test - Passed


 9160468ns  : I2C Bus 3 Initial 32-bit Data Read from Address 0x00
10086453ns  : I2C Bus 3 Received Data is 0xc782cec7
10086458ns  : I2C Bus 3 Write Data 0x00AA55FF to Address 0x00
10896055ns  : I2C Bus 3 Read Data from Address 0x00
11830453ns  : I2C Bus 3 Received Data is 0x00aa55ff
11830458ns  : I2C Bus 3 Test - Passed


11830468ns  : I2C Bus 4 Initial 32-bit Data Read from Address 0x00
12756453ns  : I2C Bus 4 Received Data is 0x62e92e8d
12756458ns  : I2C Bus 4 Write Data 0x55AA00FF to Address 0x00
13566055ns  : I2C Bus 4 Read Data from Address 0x00
14500453ns  : I2C Bus 4 Received Data is 0x55aa00ff
14500458ns  : I2C Bus 4 Test - Passed


14500468ns  : I2C Bus 5 Initial 32-bit Data Read from Address 0x00
15426453ns  : I2C Bus 5 Received Data is 0x79cd9245
15426458ns  : I2C Bus 5 Write Data 0x00FF55AA to Address 0x00
16236055ns  : I2C Bus 5 Read Data from Address 0x00
17170453ns  : I2C Bus 5 Received Data is 0x00ff55aa
17170458ns  : I2C Bus 5 Test - Passed


17170468ns  : I2C Test - Passed


17170478ns  : <<<<<I2C Test - END>>>>>
```

Figure 8.18. I2C Test Console Messages

8.2.6. CSR Test

This test evaluates the integrated I2C-to-APB module's capability to access both SCM and HPM CSR. It also serves as a test of the data channel, which is the method by which the module can access the HPM CSR. An external I2C controller is connected to the module to execute the testing.

Note that by default, all HPM IDs were set to 0xBF17 and all SCM IDs were set to 0xCA26 during IP generation. This is arbitrary and can be removed or changed as necessary. These IDs were set to help differentiate the SCM and HPM CSR.

1. The external controller sends a 32-bit read command to address 0x0000020C, effectively reading offset 0x0C (Platform ID Local) of SCM CSR (mapped to 0x200-0x2FF by default).
2. If the read data matches the expected SCM ID (0x0000CA26), a message is printed indicating the test passed. Otherwise, the message indicates the test failed.
3. The external controller sends a 32-bit read command to address 0x0000120C, effectively reading offset 0x00 (Platform ID Local) of HPM CSR (mapped to 0x200-0x2FF by default, accessed through Data Channel).
4. If the read data matches the expected HPM ID (0x0000BF17), a message is printed indicating the test passed. Otherwise, the message indicates the test failed.

After step 4, a message is printed to indicate that the CSR test is complete. At this point, the simulation is finished.

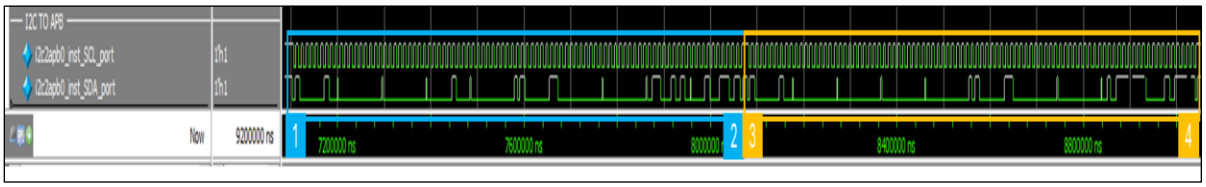

Figure 8.19. CSR Test Flow

```
17170488ns  : <<<<<CSR Access via I2C Test - START>>>>>


17170498ns  : SCM CSR Test - START
17170503ns  : Read Unique ID for SCM (Offset = 0x0000020C), expecting 0x0000CA26
18148555ns  : Received Data is 0x0000ca26
18148560ns  : SCM CSR Test - Passed
18148565ns  : SCM CSR Test - END
18148570ns  : HPM CSR Test - START
18148575ns  : Read Unique ID for HPM (Offset = 0x0000120C), expecting 0x0000BF17
19124555ns  : Received Data is 0x0000bf17
19124560ns  : HPM CSR Test - Passed
19124565ns  : HPM CSR Test - END


19124575ns  : <<<<<CSR Access via I2C Test - END>>>>>
```

Figure 8.20. CSR Test Console Messages

9. Hardware Implementation

9.1. Requirements

9.1.1. DC-SCM 2.0 Requirements

This section discusses the hardware requirements as defined by the DC-SCM 2.0 Specification.

It defines the LVDS link as AC-coupled. As such, the following components are required on the hardware:

- DC Blocking Capacitor – placed on SCM only.
- Voltage Bias Circuit – placed close to the LVDS RX I/O Buffer.

To ensure that these specifications are met within the board design, it is highly recommended that both DC blocking capacitors and the voltage bias circuit be added externally on any user board design.

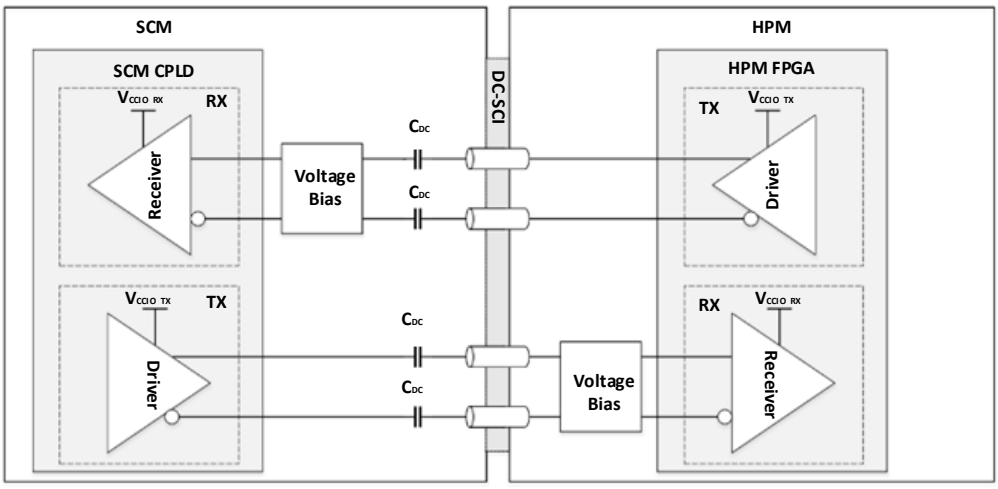

Figure 9.1. LVDS Link

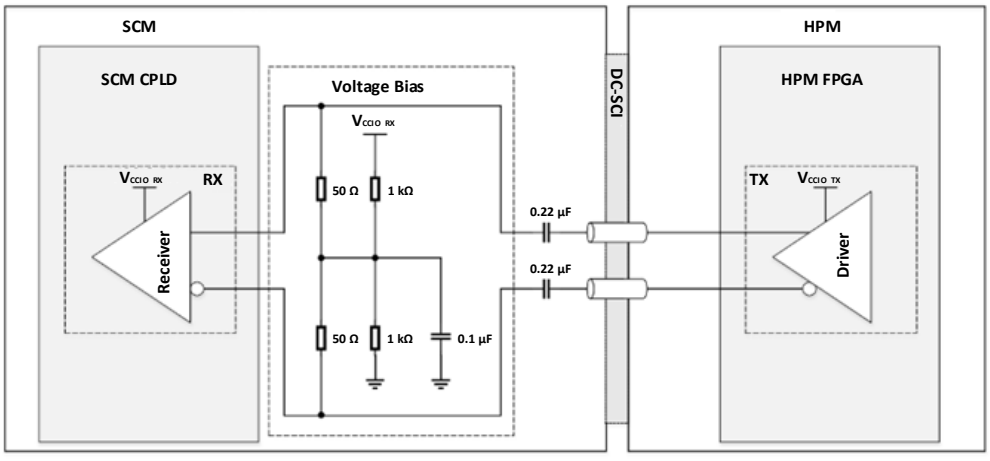

Figure 9.2. LVDS Link

Table 9.1. LVDS Electrical Requirements

Parameter	Min	Max	Description
V_{OD}	100 mV	800 mV	Output Differential Voltage Swing.
V_{ID}	100 mV	800 mV	Input Differential Voltage Swing.
$V_{CCIO\ RX/TX}$	N/A	3.3 V	LVDS VCCIO Voltage (Refer to MachXO3/MachXO3D/MachXO5-NX documentation).
V_{ICM}	N/A	($V_{CCIO\ RX}$ / 2) + 20%	Input common mode voltage on LVDS I/O pins.
C_{DC}	0.22 µF	N/A	DC blocking capacitor.
R_{TT}	100 Ω – 5%	100 Ω + 5%	LVDS Termination Resistor (can be set internally on the device).

9.1.2. Lattice FPGA Requirements

This section describes the device-specific requirements for implementing the reference design on hardware. Note that by default, these requirements are already covered by the reference design.

Below is the guidance for the pin assignments of input and output LVDS for MachXO3 and MachXO3D:

- Input LVDS: These signals can only be assigned to A/B pairs (PBxxA/PBxxB) on Bank 2. In addition, the LVDS clock should be assigned to a PCLK pin (PCLKT2_x /PCLKC2_x).
- Output LVDS: These signals can only be assigned to A/B pairs (PTxxA/PTxxB) on Bank 0.

For MachXO5-NX, the LVDS pairs (both input and output) should be mapped to high-performance I/O banks (bottom banks). For Dual Node implementation, each node must be on different banks.

Disclaimer: This is not a comprehensive list of requirements for each device. For more information, refer to the supporting documents for each device available on the following web pages:

- MachXO3 web page
- MachXO3D web page
- MachXO5-NX web page

10. Resource Utilization

The resource utilization depends on the DC-SCM LTPI IP configuration. The following tables show the resource utilization of the reference design based on device family.

Note: Resource utilization values may vary depending on whether additional or fewer channel buses are used.

Table 10.1. MachXO3 Device Resource Utilization for Demo Designs

Configuration	LUT4	PFU Register	EBR	Slice
DC-SCM LTPI IP – SCM				
LTPI Reference Design	4072	2950	2	2461
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	3738	2611	2	2246
I2C-to-APB Bridge	208	232	0	119
Miscellaneous LTPI Modules	126	107	0	97
DC-SCM LTPI IP – HPM				
LTPI Reference Design	4277	3070	2	2510
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	3780	2727	2	2250
I2C-to-APB Bridge	214	228	0	119
Miscellaneous LTPI Modules	283	115	0	142

Table 10.2. MachXO3D Device Resource Utilization for Demo Designs

Configuration	LUT4	PFU Register	EBR	Slice
DC-SCM LTPI IP – SCM				
LTPI Reference Design	4072	2950	2	2461
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	3738	2611	2	2246
I2C-to-APB Bridge	208	232	0	119
Miscellaneous LTPI Modules	126	107	0	97
DC-SCM LTPI IP – HPM				
LTPI Reference Design	5957	3734	4	3396
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	3807	2726	2	2251
I2C-to-APB Bridge	211	228	0	115
I3C Controller IP	1606	662	2	836
Miscellaneous LTPI Modules	333	118	0	196

Table 10.3. MachXO5-NX Device Resource Utilization for Demo Designs

Configuration	LUT4	PFU Register	EBR	Slice
DC-SCM LTPI IP – SCM (Single Node)				
LTPI Reference Design	5124	2948	3	—
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	4629	2598	2	—
I2C-to-APB Bridge	205	232	0	—
Miscellaneous LTPI Modules	290	118	1	—
DC-SCM LTPI IP – SCM (Dual Node)				
LTPI Reference Design	10369	5633	6	—
DC-SCM LTPI IP LLGPIO: 16-bit (per node) NLGPIO: 128-bit (per node) I2C/SMBus: 6 bus (per node) UART : 2 channels (per node) OEM : 32-bit (per node)	9609	5169	4	—
I2C-to-APB Bridge	217	232	0	—
Miscellaneous LTPI Modules	543	232	2	—
DC-SCM LTPI IP – HPM				
LTPI Reference Design	7284	4037	5	—
DC-SCM LTPI IP LLGPIO: 16-bit NLGPIO: 128-bit I2C/SMBus: 6 bus UART : 2 channels OEM : 32-bit	4539	2709	2	—
I2C-to-APB Bridge	213	228	0	—
I3C Controller IP	2038	971	2	—
Miscellaneous LTPI Modules	494	129	1	—

11. Common Issues and Troubleshooting

11.1. Link Not Reaching Active State

The most common issue observed in the design is that the LTPI does not reach the Active State (status_display[0] = 0). This can be caused by any of the following:

1. IP Feature Capability Mismatch
   - This occurs when the feature capability of HPM and SCM does not match, meaning that one or more of the enabled or disabled interfaces do not match. This can be checked using the **Capabilities** tab of the IP (as shown in Figure 11.1).

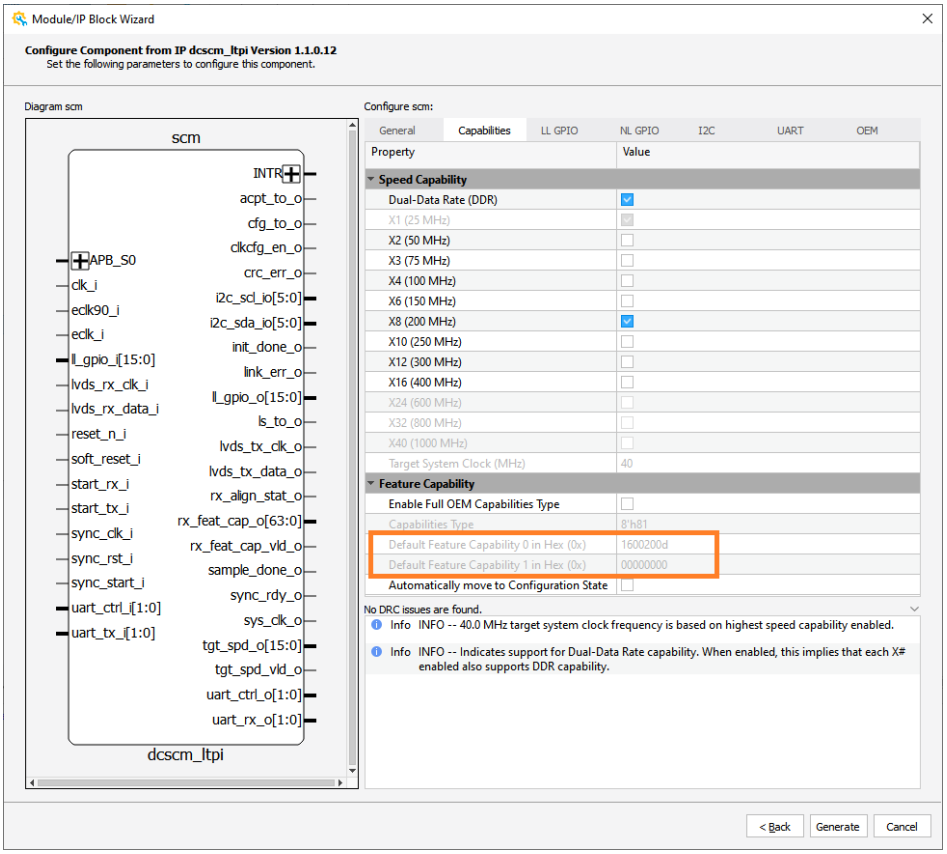

Figure 11.1. Feature Capabilities

2. Floating Inputs
   - This issue occurs when interface inputs, LL inputs, NL inputs, I2C bus, and UART TX do not have valid logic values and are floating or in tri-state. It is indicated by consecutive crc_err_o assertions during start-up. This can be resolved by placing initial values for LL, NL, and UART, as well as pull-up for the I2C bus.

11.2. I2C Bus Hang

This issue sometimes occurs when the I2C transaction is abruptly ended or interrupted. In this scenario, the SCL or SDA of the LTPI I2C channel will remain indefinitely on logic low. This can be resolved through the following methods:

1. Inducing an external reset – this, however, does not guarantee that the issue will not recur.
2. Enabling the I2C timer for both SCM and HPM – this is discussed in Modifying the Reference Design section.
3. Sending an I2C bus reset through the offset 0x3C – Figure 11.2 describes the register format of offset 0x3C.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
IP Control						
0x3C	I2C_BUS_RST	RO	32'h0	[31:24]	RSVD	Reserved bits.
		RW		[23:0]	i2c_bus_rst	I²C bus controller reset. When asserted, IP releases the local bus and resets the interface controller of the corresponding bus link number. Each bit index is mapped to each I²C bus link reset with bus link 0 occupying index 0. [0] – I²C bus 0 reset [1] – I²C bus 1 reset ... [23] – I²C bus 23 reset

Figure 11.2. I2C_BUS_RST Register

References

- DC-SCM LTPI IP User Guide (FPGA-IPUG-02200)
- Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057)
- Implementing High-Speed Interfaces with MachXO3D Usage Guide (FPGA-TN-02065)
- MachXO5-NX High-Speed I/O Interface (FPGA-TN-02286)
- MachXO3 sysCLOCK PLL Design and User Guide (FPGA-TN-02058)
- MachXO3D sysCLOCK PLL Usage Guide (FPGA-TN-02070)
- DC-SCM 2.0 Specification
- Open Compute Project web page
- LVDS Tunneling Protocol and Interface Reference Design web page
- MachXO3 web page
- MachXO3D web page
- MachXO5-NX web page
- Lattice Diamond Software web page
- Lattice Propel Design Environment web page
- Lattice Radiant Software web page
- Lattice Insights web page 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, please refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.6, April 2026

Section	Change Summary
All	Made editorial fixes.

Introduction	• Updated Data Datacenter-ready Secure Control Module (DC-SCM) version to 2.2 and LTPI version to 1.2.
	• Table 1.1. Summary of the Reference Design:
	  • updated Lattice Radiant version to 2025.2;
	  • updated Lattice Propel version to 2025.2;
	  • updated the DC-SCM LTPI version to 3.0.0 (via Propel Builder IP Catalog);
	  • updated I2C-to-APB Bridge Reference Design to 1.3.0 (via Propel Builder IP Catalog);
	  • newly added *APB Streamer 1.0.0 (via IPK folder)*;
	  • updated Board and Cable information to the current.
	• Features:
	  • updated the supported DC SCM version to 2.2, LTPI version to 1.2 (Oct. 2025);
	  • newly added the last two items to the Signal Names section.

Directory Structure and Files	• Updated Figure 2.1. Directory Structure. • Made global update to Table 2.1. File List.
Functional Description	Made global updates to the Design Components section including all the descriptions, figures and tables, except the GPIO Latency section.
Signal Description	• Made global updates to the whole section including table. • Removed the original Table 4.2. HPM Top Module Signals.
Register Description	Newly added this section.
Timing Constraints	Made global updates to the whole section.
Modifying the Reference Design	Reworked the whole section except the Lattice Diamond and Lattice Radiant sections.
Hardware Implementation	Newly added *For Dual Node implementation, each node must be on different banks* to the Lattice FPGA Requirements section.
Resource Utilization	Made global updates to Table 10.1. MachXO3 Device Resource Utilization for Demo Designs, Table 10.2. MachXO3D Device Resource Utilization for Demo Designs, and Table 10.3. MachXO5-NX Device Resource Utilization for Demo Designs.

Revision 1.5, September 2025

Section	Change Summary
All	• Added the OCP logo to the document cover page. • Updated all instances of *APB Slave Error* to *APB Completer Error*. • Made editorial fixes.
Abbreviations in This Document	Added *AC*, *AMBA*, *FPGA*, *GDDR*, *GUI*, *HDL*, *IPK*, *LUT*, *N/A*, *OCP*, *PFU*, *PLL*, *SCM*, and *USB*.
Introduction	• Added the *OCP Ready™* statement to the Introduction section. • Updated *Software Requirements* and *Hardware Requirements* in Table 1.1. Summary of the Reference Design.
Directory Structure and Files	• Updated Figure 2.1. Directory Structure. • Added *IPK Subfolder* to Table 2.1. File List.
Functional Description	• Removed the 25 MHz Generator module description in the Functional Description section. • Updated Figure 3.2. SCM Block Diagram, Figure 3.3. HPM Block Diagram, Figure 3.4. LTPI (SCM) Block Diagram, and Figure 3.5. LTPI (HPM) Block Diagram. • Updated Table 3.1. LTPI Clock Source Signals, Table 3.2. Reset Timer Signals, Table 3.4. PLL Streamer Logic Signals, and Table 3.4. PLL Streamer Logic Signals. • Removed the *25 MHz Generator*, *Initialization Logic*, *Clock Generator*, *APB Interconnect (SCM*

Section	Change Summary
	Only), and *APB Feedthrough (HPM Only)* sections. • Updated the descriptions in the Reset Timer, LTPI Clock Source, Clocking Scheme, and DC-SCM LTPI IP Control and Status Registers sections.. • Removed *Table 3.15. DC-SCM LTPI IP Registers*.
Signal Description	Updated Table 4.1. SCM Top Module Signals and Table 4.2. HPM Top Module Signals.
Timing Constraints	Updated this section.

Modifying the Reference Design	• In the Changing the DC-SCM LTPI IP Settings section:
	  • Updated Figure 7.3. Check Generated Result Window – Figure 7.6. Frame Format Tab.
	  • Updated the General Tab and Frame Format Tab sections.
	  • Removed the *Capabilities Tab*, *LL GPIO Tab*, *NL GPIO Tab*, *I2C Tab*, *UART Tab*, and *OEM Tab* sections.
	  • Added the Channels Tab and Debug: IP Parameters sections.
	• Removed the *Removing the 25 MHz Generator*, *Changing the 25 MHz Generator Input*, and *Changing the 25 MHz Generator Input* sections.

Simulating the Reference Design	• Updated the following sections:
	  • Simulation Details and Results
	  • Link Training (including all figures)
	  • GPIO Test (including all figures)
	  • UART Test (including Figure 8.16. UART Test Console Messages)
	  • I2C Test (including Figure 8.18. I2C Test Console Messages)
	  • CSR Test (including Figure 8.20. CSR Test Console Messages)
	• Added the OEM Test section.

Hardware Implementation	• Removed *Table 8.1. Pin Assignments* and its description. • Updated the disclaimer in the Lattice FPGA Requirements section.
Common Issues and Troubleshooting	Removed the *Clock Mismatch* information in the Link Not Reaching Active State section.
Resource Utilization	Updated this section.
References	Added the *Open Compute Project* web page and removed *Lattice Propel 2024.2 Builder User Guide (FPGA-UG-02219)*, *Lattice Radiant Software 2024.2 User Guide*, and *Lattice Diamond 3.14 User Guide*.

Revision 1.4, June 2025

Section	Change Summary
All	Minor editorial fixes.

Introduction	• Removed the following subsections:
	  • Limitations
	  • Conventions
	  • Attribute Names
	• Added new subsections:
	  • Quick Facts
	  • Naming Conventions
	• Removed support for Mach-NX.

Directory Structure and Files	• Added this section. • Added new directory structure overview and description.

Functional Description	• Removed the Pin Description section and replaced it with the *Functional Description* section.
	• Removed the following subsections:
	  • Overview
	  • Reference Design Specification
	  • Target Speed Setting
	  • Clock Recovery
	  • Additional Requirement for Mach-NX

Section	Change Summary

	• Added the new subsections:
	  • Design Components
	  • Clocking Scheme
	  • Reset Scheme
	• Reworked subsection contents of LL GPIO Latency and renamed to GPIO Latency.

Signal Description	• Removed the Reference Design Macros section and replaced it with the *Signal Description* section. • Added top-level ports for SCM and HPM.
Timing Constraints	• Removed the *Diamond Design* and *Radiant Design* subsections. • Added Clock Definitions subsection.

Modifying the Reference Design	• Removed Generating the IP section and replaced it with *Modifying the Reference Design* section.
	• Added the following subsections:
	  • Lattice Propel
	  • Lattice Diamond
	  • Lattice Radiant

Simulating the Reference Design	• Removed HDL Simulation and Verification section and replaced it with *Simulating the Reference Design* section.
	• Added the following subsections:
	  • Setting Up the Simulation
	  • Simulation Details and Results
	• Added new simulation screenshots.

Hardware Implementation	Removed the Hardware Setup section and Hardware Validation section and replaced it with the *Hardware Implementation* section.
Resource Utilization	Updated the resource utilization numbers to match utilization for LTPI IP version 1.6.0.
Common Issues and Troubleshooting	Added I2C Bus Hang subsection.
Packaged Design	Removed this section.

Revision 1.3, August 2024

Section	Change Summary
All	Added *User Guide* to the document title.
Disclaimers	Updated boilerplate.
Inclusive Language	Added boilerplate.
Abbreviations in This Document	• Replaced the word *acronyms* with *abbreviations*. • Added *Management Component Transport Protocol* (*MCTP*) to the list of abbreviations.

Introduction	• Updated the following items in the Features section:
	  • Added an introductory sentence.
	  • Updated the *LTPI version*.
	  • Removed the sentence: *for I2C interface, each can be configured as Controller, Target or Controller/Target (for multi-controller/main)*.
	  • Added the descriptions for *I²C interface Single-node* and *Multi node* features.
	• Updated the versions of the *Lattice Diamond*, *Lattice Radiant*, and *Lattice Propel* software in the Limitation section.

Generating the IP	Added section 6.3. Sample Reference Design Setup.
References	Added the following references: • Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057) • Implementing High-Speed Interfaces with MachXO3D Usage Guide (FPGA-TN-02065) • Implementing High-Speed Interfaces with Mach-NX Usage Guide (FPGA-TN-02234) • MachXO5-NX High-Speed I/O Interface (FPGA-TN-02286)

Section	Change Summary
	• Lattice Insights web page for Lattice Semiconductor training courses and learning plans

Revision 1.2, June 2023

Section	Change Summary
Acronyms in This Document	Added CDC and removed SCM.
Introduction	• Updated LTPI version and supported data rates in the Features section. • Added MachXO5-NX to supported devices and updated software versions in the Limitations section.
Functional Description	• Updated the Overview section to add the DC-SCM. • General update to the Reference Design Specification (previously Reference Design Variations) and the Target Speed Setting sections. • Removed the Clock Recovery Diagram figure from the Clock Recovery section. Marked statement as Note. • Updated information on the latency of LL GPIO in the LL GPIO Latency section. • Deleted statement regarding a separate demo from the Additional Requirement for Mach-NX section.

Pin Description	• Removed the statement Note that some pins may not appear on every variation of the reference design from introductory paragraph.
	• In Table 3.1. Pin Descriptions:
	  • Moved int_o to the System group.
	  • Updated tgt_spd_o description.
	  • Updated values and description of rx_feat_cap_o.

Reference Design Macros	General update to this section (previously *Parameters and Synthesis Directives*)
Timing Constraints	• Added timing constraints guidelines for MachXO5-NX. • Added the Diamond Design, Lattice Radiant Design, Clocks, Resets, and False Paths subsections.
Generating the IP	Updated steps and figures in the main section and in the Sample SCM IP Configuration (Default Reference Design) subsection.
HDL Simulation and Verification	• Provided detailed information regarding Figure 7.1. Initialization after Reset. • Updated figures.
Hardware Setup	General update to this section.
Hardware Validation	Updated hardware validation information for MachXO3D and Mach-NX. Added information on MachXO5-NX.
Packaged Design	Updated the design directory structure and added note.
Resource Utilization	Updated section to show only the resource utilization of the Diamond and the Lattice Radiant demos.
References	Added references to product web pages.
Technical support Assistance	Added reference to the Lattice Answer Database on the Lattice website.

Revision 1.1, November 2022

Section	Change Summary

Introduction	• In the Limitations section:
	  • added Mach-NX to the Supported Device Families;
	  • updated the Supported Lattice Propel version to 2.2.

Functional Description	• General update to the Target Speed Setting section. • Added the Using a Single PLL section. • Added the Additional Requirement for Mach-NX section. • In the section Clock Recovery:

Section	Change Summary
	• added a description for Figure 2.4. Clock Recovery Diagram; • added information about additional clock recovery function of the IP.

Pin Descriptions	• Added information of the Initialization Interface.
	• In Table 3.1. Pin Descriptions:
	  • updated the Description for the sync_clk_i Port;
	  • updated the note of the Initialization Interface.

Timing Constraints	Added the current timing constraints applied to the design.

Generating the IP	• Added information about the usage of RTL files in step 8 of the IP generation process.
	• Updated Figure 6.9. Sample RTL File Path.
	• Updated step 9 of the IP generation process.
	• Added Figure 6.11. Sample Key File Path.
	• Added the Setting-up Diamond Software section.
	• Added information about the Enable Clock Compensation option in the section Sample SCM IP Configuration (Design 1) section.
	• Added the following tables:
	  • Table 6.1. General Tab Attributes
	  • Table 6.2. General Tab Attributes Description
	  • Table 6.5. Capabilities Tab Attributes
	  • Table 6.6. Capabilities Tab Attributes Description
	  • Table 6.7. LL GPIO Tab Attributes
	  • Table 6.8. LL GPIO Tab Attributes Description
	  • Table 6.9. NL GPIO Tab Attributes
	  • Table 6.10. NL GPIO Tab Attributes Description
	  • Table 6.11. I2C Tab Attributes
	  • Table 6.12. I2C Tab Attributes Description
	  • Table 6.13. UART Tab Attributes
	  • Table 6.14. UART Tab Attributes Description
	  • Table 6.15. OEM Tab Attributes
	  • Table 6.16. OEM Tab Attributes Description
	• Added Figure 6.20. NL GPIO Tab and Figure 6.23. OEM Tab.

HDL Simulation and Verification	Added the Running the Simulation section.
Hardware Setup	• In Table 8.1. Diamond Demo Pin Assignments, updated the Pin Assignment information of the following Ports of the LVDS Interface: lvds_tx_clk_o, lvds_tx_data_o, lvds_rx_clk_i, and lvds_rx_data_i. Added the LVDS Setup Considerations section.
Hardware Validation	Added information about the hardware validation for Mach-NX device.
References	Updated this section.

Revision 1.0, August 2022

Section	Change Summary
All	Production release
Introduction	Updated the features to match the new IP implementation.
Functional Description	• Updated section content to match the new IP implementation, including Figure 2.1. Block Diagram and Figure 2.2. State Machine. • Updated Table 2.1. Design to change Clock Recovery value for Design 1 (SCM), Design 2 (SCM), and Design 3 SCM to No.
Pin Descriptions	Updated pin list to match the new reference design top module in Table 3.1. Pin Descriptions.
Parameters and Synthesis Directives	Changed section name from *Parameters* to *Parameters and Synthesis Directives*. Added Table 4.1. Design Parameters and Table 4.2. Synthesis Directives.
Timing Constraints	Added this section.

Section	Change Summary
Generating the IP	Added this section.
HDL Simulation and Verification	Updated section content to match the new IP implementation, including Figure 7.1. Initialization after Reset and Figure 7.2. Interface Activities.
Hardware Setup	Added this section.
Hardware Validation	Updated section content to include validation for MachXO3D.
Common Issues and Troubleshooting	Added this section.
Packaged Design	Updated section content to match the new reference design package, including Figure 11.1. Packaged Design Directory Structure.
Resource Utilization	Updated section content to match the new IP implementation, including adding Table 12.1. Resource Utilization for Design 1 and Table 12.2. Resource Utilization for Added Channels.

Revision 0.80, February 2022

Section	Change Summary
All	First preliminary release

www.latticesemi.com