LVDS Tunneling Protocol and Interface (LTPI) User Guide

Reference Design

FPGA-RD-02247-1.4

June 2025

Disclaimers

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
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
GPIO	General Purpose Input/Output
HPM	Host Processor Module
I/O	Input/Output
I^2C	Inter-Integrated Circuit
IP	Intellectual Property
LED	Light-Emitting Diode
LLGPIO	Low Latency General Purpose Input/Output
LTPI	LVDS Tunneling Protocol and Interface
LVDS	Low Voltage Differential Signaling
MCSI	Multi-Channel Serial Interface
MCTP	Management Component Transport Protocol
NLGPIO	Normal Latency General Purpose Input/Output
OEM	Original Equipment Manufacturer
RTL	Register Transfer Level
Rx	Receiver
SDR	Single Data Rate
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter

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) 2.0 LTPI 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.

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 2024.2 for the MachXO5-NX devices Lattice Propel™ version 2024.2 Synplify Pro® Questa Lattice OEM Edition-64 2024.2
	IP version (if applicable)	DC-SCM LTPI 1.6.0 APB Interconnect 1.2.1 APB Feedthrough 1.1.0 I2C to APB Bridge RD 1.2.0
Hardware Requirements	Board	MachXO3LF Starter Kit MachXO3D Breakout Board MachXO5-NX Development Board
	Cable	USB Mini-B to USB-A Cable (provided for all boards except the LTPI board)

1.2. Features

Key features of the LTPI reference design include:

- Support for DC-SCM 2.0 LTPI version 1.1 (Oct 2023)
- 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

2. Directory Structure and Files

Figure 2.1 shows the directory structure.

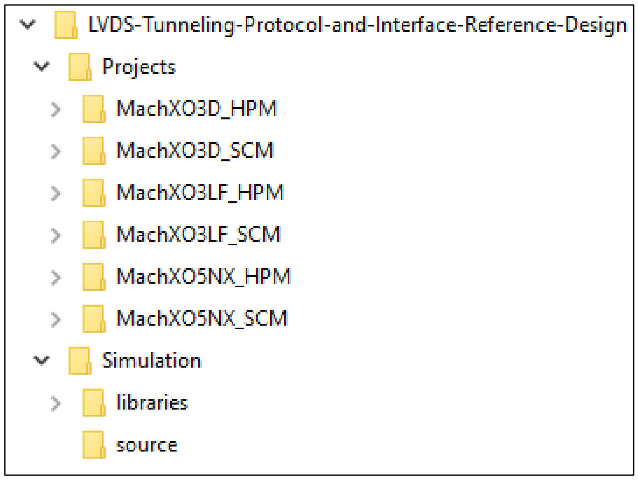

Figure 2.1. Directory Structure

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

Table 2.1. File List

Subfolder	Description
Projects	This subfolder contains pre-built projects, subcategorized by the function and the Lattice FPGA used. • MachXO3D_HPM • MachXO3D_SCM • MachXO3LF_HPM • MachXO3LF_SCM • MachXO5NX_HPM • MachXO5NX_SCM Each project has been tested via simulation and on hardware.
Simulation	This subfolder contains the 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* • SCM_XO3D_HPM_XO3D.do • SCM_XO3D_HPM_XO3LF.do • SCM_XO3D_HPM_XO5.do • SCM_XO3LF_HPM_XO3D.do • SCM_XO3LF_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 *.do files to add preset signals on the Wave view of the simulator

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 via 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 the 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 25 MHz clock from the 25 MHz Generator module serves as the primary system clock unless specified otherwise.

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

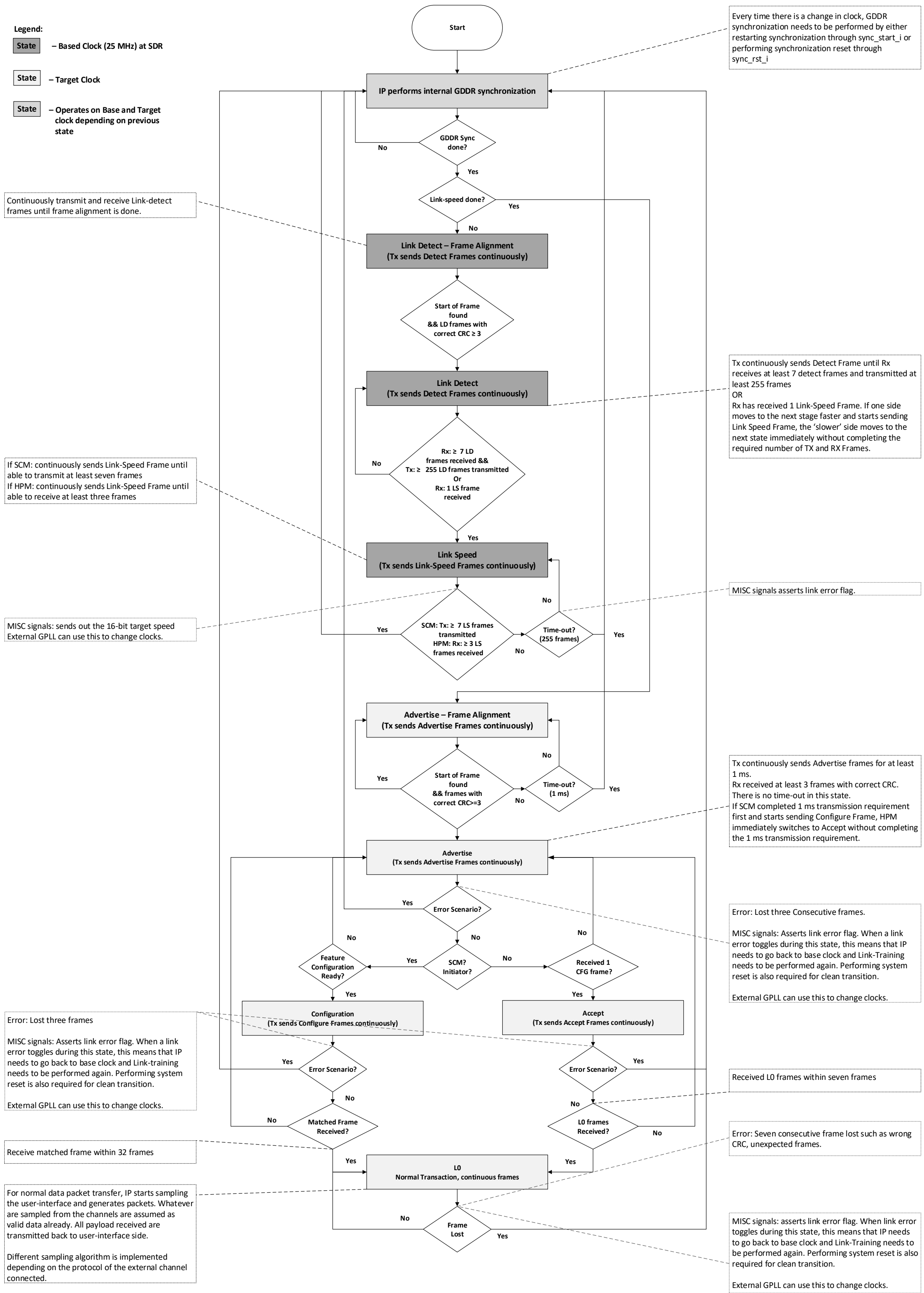

Figure 3.1. LTPI State Flow

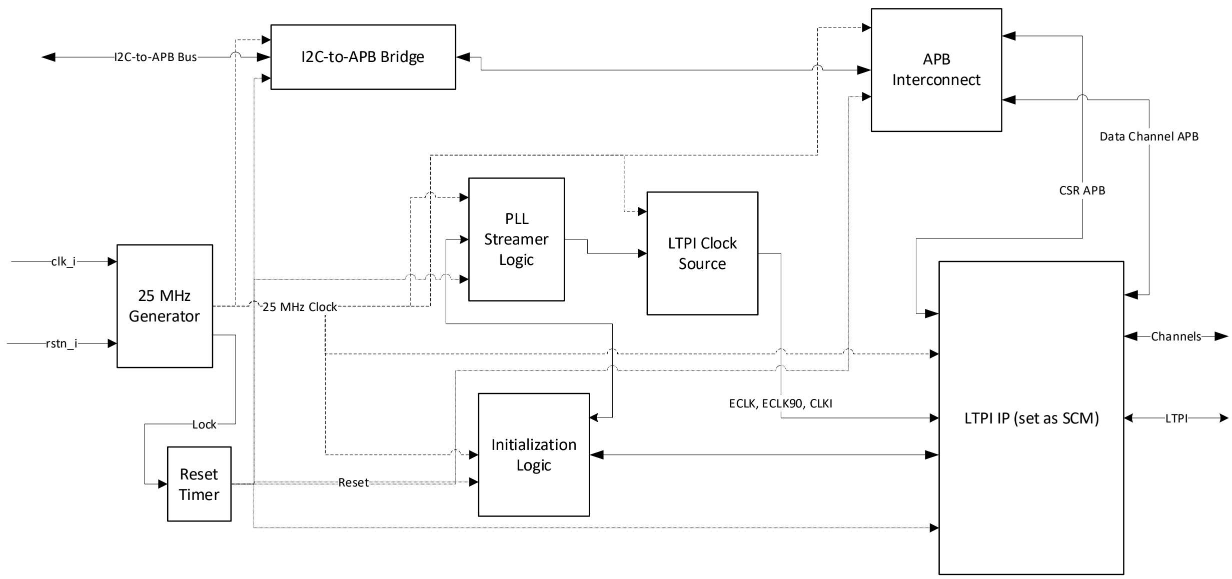

Figure 3.2. SCM Block Diagram

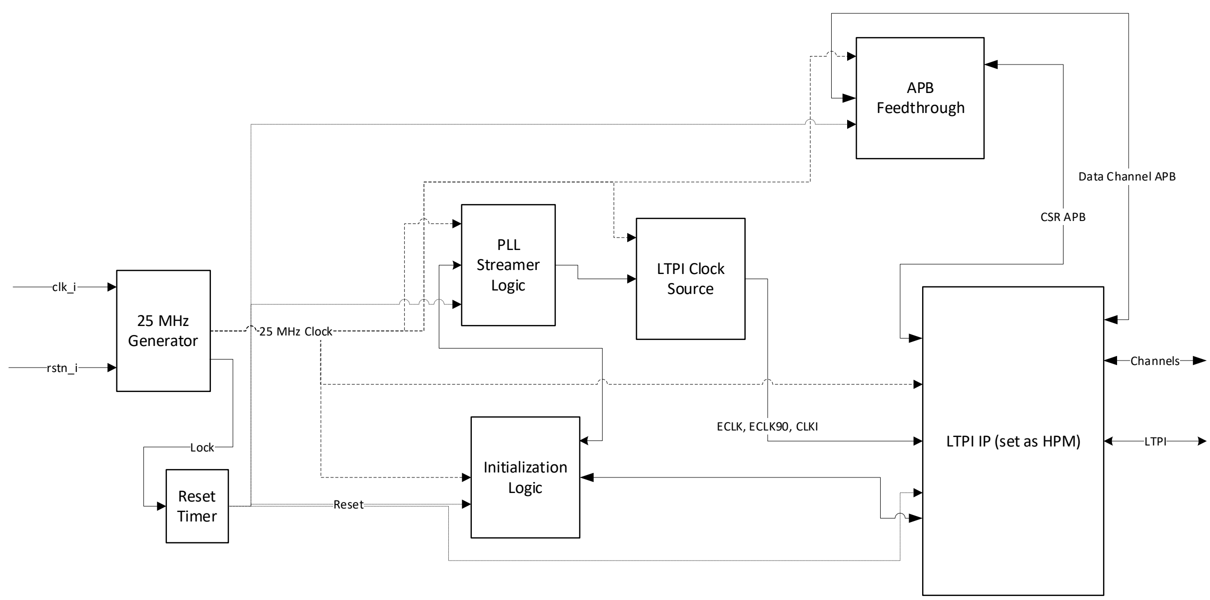

Figure 3.3. HPM Block Diagram

3.1. Design Components

This section outlines the module components of the top-level module in the reference design. For the signal description of the top-level module, refer to the Signal Description section.

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 (for SCM instance) or hpm0 (for HPM instance).

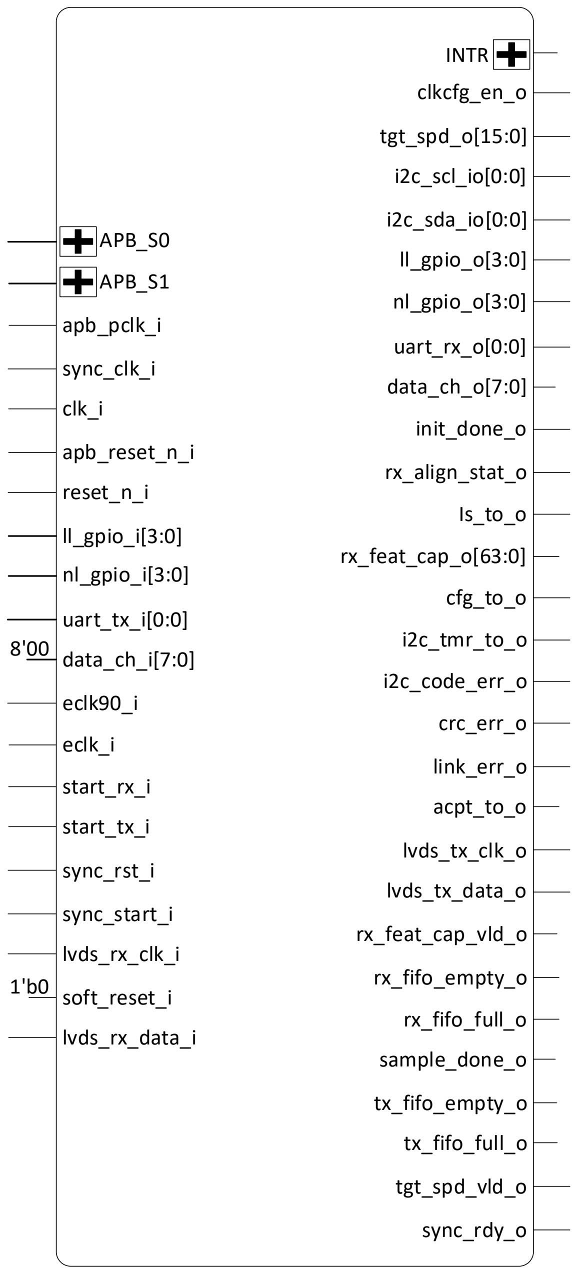

Figure 3.4. LTPI (SCM) Block Diagram

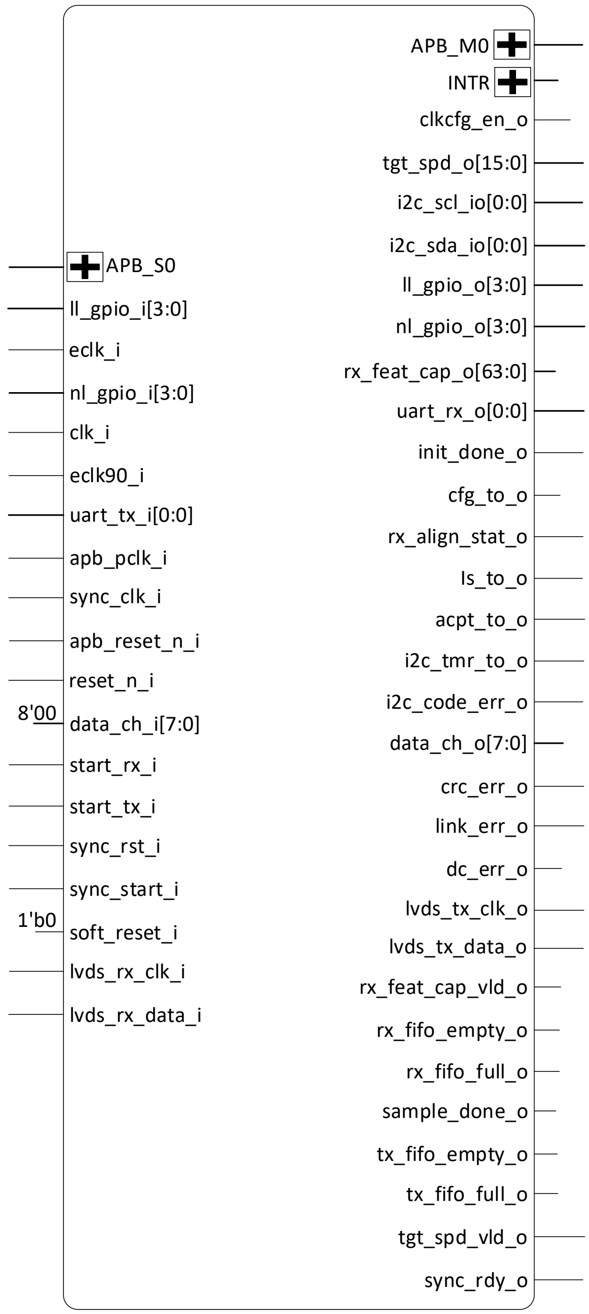

Figure 3.5. LTPI (HPM) Block Diagram

Table 3.1. LTPI Clock Source Signals

Port Name	Input/Output	Width	Default Value	Description
System Signals				
clk_i	Input	1	N/A	The LTPI main clock connects to either CLKOS2 of LTPI clock source or sys_clk_o of clock generator.
reset_n_i	Input	1	N/A	Active low reset.
soft_reset_i	Input	1	1’b0	LTPI Active high soft reset.
eclk_i	Input	1	N/A	The ECLK input connects to either CLKOP or clkop_o of the LTPI clock source.

Port Name	Input/Output	Width	Default Value	Description
eclk90_i	Input	1	N/A	The ECLK90 input connects to either CLKOS or clkos_o of the LTPI clock source.
INTR	Interrupt	N/A	N/A	Interrupt interface. Unused.
PHY Initialization Signals				
sync_clk_i	Input	1	N/A	PHY synchronization clock. 25 MHz clock input.
start_rx_i	Input	1	N/A	Initialize Rx PHY. Connects to start_rx_o of initialization logic.
start_tx_i	Input	1	N/A	Initialize Tx PHY. Connects to start_tx_o of initialization logic.
sync_rst_i	Input	1	N/A	PHY reset. Connects to sync_rst_o of initialization logic.
sync_start_i	Input	1	N/A	The PHY synchronization start trigger connects to sync_start_o of the initialization logic.
sync_rdy_o	Output	1	1’b0	The PHY synchronization ready connects to sync_rdy_i of the initialization logic.
LTPI Signals				
lvds_rx_clk_i	Input	1	N/A	LVDS Receive Clock. Connects to scm0_inst_lvds_rx_clk_i_port or hpm0_inst_lvds_rx_clk_i_port (Top Module).
lvds_rx_data_i	Input	1	N/A	LVDS Receive Data. Connects to scm0_inst_lvds_rx_data_i_port or hpm0_inst_lvds_rx_data_i_port (Top Module).
lvds_tx_clk_o	Output	1	N/A	LVDS Transmit Clock. Connects to scm0_inst_lvds_tx_clk_o_port or hpm0_inst_lvds_tx_clk_o_port (Top Module).
lvds_tx_data_o	Output	1	N/A	LVDS Transmit Data. Connects to scm0_inst_lvds_tx_data_o_port or hpm0_inst_lvds_tx_data_o_port (Top Module).
Low Latency GPIO Channel				
ll_gpio_i	Input	4	N/A	Low Latency Input. Connect scm0_inst_ll_gpio_i_portbus or hpm0_inst_ll_gpio_i_portbus (Top Module).
ll_gpio_o	Output	4	N/A	Low Latency Output. Connect scm0_inst_ll_gpio_o_portbus or hpm0_inst_ll_gpio_o_portbus (Top Module).
Normal Latency GPIO Channel				
nl_gpio_i	Input	4	N/A	Low Latency Input. Connect scm0_inst_nl_gpio_i_portbus or hpm0_inst_nl_gpio_i_portbus (Top Module).
nl_gpio_o	Output	4	N/A	Low Latency Output. Connect scm0_inst_nl_gpio_o_portbus or hpm0_inst_nl_gpio_o_portbus (Top Module).
UART Channel				
uart_tx_i	Input	1	N/A	UART Tx pin. Connect scm0_inst_uart_tx_i_portbus or hpm0_inst_uart_tx_i_portbus.
uart_rx_o	Output	1	N/A	UART Rx pin. Connect scm0_inst_uart_rx_o_portbus or hpm0_inst_uart_rx_o_portbus (Top Module).
I2C Channel				
i2c_scl_io	Inout	1	N/A	I2C Clock pin. Connects to scm0_inst_i2c_scl_io_portbus or hpm0_inst_i2c_scl_io_portbus (Top Module).
i2c_sda_io	Inout	1	N/A	I2C Data pin. Connects to scm0_inst_i2c_sda_io_portbus or hpm0_inst_i2c_sda_io_portbus (Top Module).
Data Channel				
data_ch_i	Input	8	8’h00	Data channel input tag. Unused.
data_ch_o	Output	8	8’h00	Data channel output tag. Unused.
APB_S1	APB IF	N/A	N/A	Data Channel Completer Interface (SCM only). Connects to APB_M00 of APB Interconnect.
APB_M0	APB IF	N/A	N/A	Data Channel APB Requester Interface. Connects to APB_S0 of APB Feedthrough.

Port Name	Input/Output	Width	Default Value	Description
APB Interface				
apb_pclk_i	Input	1	N/A	APB interface clock. 25 MHz clock input.
apb_reset_n_i	Input	1	N/A	Active-low reset.
APB_S0	APB IF	N/A	N/A	CSR APB Completer Interface. Connects to APB_M01 of APB Interconnect (SCM only) or APB_M0 of APB Feedthrough (HPM only).
Miscellaneous Signals				
clkcfg_en_o	Output	1	N/A	Clock Reconfiguration Enabled Flag. It indicates that the LTPI clocks can now change from Base Speed to Target Speed. It Connects to clkcfg_en_i of Initialization Logic and PLL Streamer Logic.
tgt_spd_o	Output	32	N/A	Target Speed Status. It Indicates the highest common available frequency between SCM and HPM. Note that only one bit can be high for bits 0-9. Bit 15: DDR if 1, SDR if 0 Bit 14-10: Reserved Bit 9: 600 MHz Bit 8: 400 MHz Bit 7: 300 MHz Bit 6: 250 MHz Bit 5: 200 MHz Bit 4: 150 MHz Bit 3: 100 MHz Bit 2: 75 MHz Bit 1: 50 MHz Bit 0: 25 MHz It Connects to tgt_spd_i of PLL Streamer Logic and Clock Generator.
rx_feat_cap_o	Output	64	N/A	Received Feature Capabilities. Unconnected.
init_done_o	Output	1	N/A	Link Established Flag. Connects to concat_module_inst_status_display_portbus[0] (Top Module).
cfg_to_o	Output	1	N/A	Configuration State Timeout. Only active for SCM. Connects to concat_module_inst_status_display_portbus[3] (Top Module, SCM only).
rx_align_stat_o	Output	1	N/A	PHY Rx Alignment Done Flag. Connects to concat_module_inst_status_display_portbus[1] (Top Module).
ls_to_o	Output	1	N/A	Link Speed State Timeout. Connects to concat_module_inst_status_display_portbus[2] (Top Module).
acpt_to_o	Output	1	N/A	Accept State Timeout. Only active for HPM. Connects to concat_module_inst_status_display_portbus[3] (Top Module, HPM only).

Port Name	Input/Output	Width	Default Value	Description
i2c_tmr_to_o	Output	1	N/A	I2C Timer Timeout/Reset Flag. Inactive (low) when the I2C channel is disabled. When the I2C channel is enabled, this is only valid if any of the following is true: • Enable Timer for Bus # option of any of the available I2C bus is checked; or • Generate STOP on Bus # Reset option of any of the available Target I2C bus is checked. This indicates that a timeout or bus reset occurred on any of the active I2C channels and that the bus recovery is successful. It Connects to concat_module_inst_status_display_portbus[4] (Top Module).
i2c_code_err_o	Output	1	N/A	Invalid I2C Event Received Flag. Connects to concat_module_inst_status_display_portbus[5] (Top Module).
crc_err_o	Output	1	N/A	Frame CRC Error. The following number of consecutive CRC Errors will result in a link error: • Advertise State – 3 • Active State – 7 Connects to concat_module_inst_status_display_portbus[6] (Top Module).
link_err_o	Output	1	N/A	Link Error Flag. It indicates that the link between SCM and HPM is lost. It Connects to concat_module_inst_status_display_portbus[7] (Top Module).
dc_err_o	Output	1	N/A	Data Channel Error Flag. Not connected.
rx_feat_cap_vld_o	Output	1	N/A	Received Feature Capability Valid Flag. Not connected.
rx_fifo_empty_o	Output	1	N/A	Rx FIFO Empty Flag. Not connected.
sample_done_o	Output	1	N/A	Sample Done Flag. This indicates that the IP is done sampling a frame. Not connected.
tx_fifo_empty_o	Output	1	N/A	Tx FIFO Empty Flag. Not connected.
tgt_spd_vld_o	Output	1	N/A	Target Speed Valid Flag. Connects to tgt_spd_vld_i of PLL Streamer Logic.

3.1.2. 25 MHz Generator

This module serves as a wrapper for a PLL instance that accepts an external clock input and converts it into a 25 MHz clock. This module can be omitted if an external 25 MHz clock is available.

Table 3.3 for the difference in implementation across the three devices.

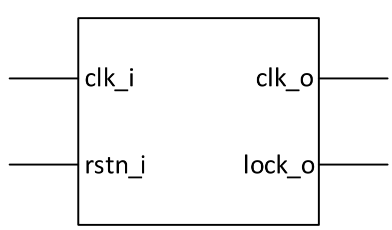

Figure 3.6. 25 MHz Generator (MachXO3) Block Diagram

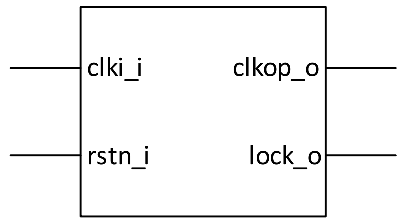

Figure 3.7. 25 MHz Generator (MachXO5-NX) Block Diagram

Table 3.2. 25 MHz Generator Signals

Port Name	Input/Output	Width	Default Value	Description
MachXO3/MachXO3D				
clk_i	Input	1	N/A	External clock input. 12 MHz for MachXO3 and MachXO3D. 125 MHz for MachXO5-NX Connects to clk_i (Top Module).
rstn_i	Input	1	N/A	Active-low reset. Coonects to rstn_i (Top Module).
clk_o	Output	1	N/A	25 MHz clock output.
lock_o	Output	1	1’b0	PLL lock. Connects to reset_n_i input of the reset timer.
MachXO5-NX				
clki_i	Input	1	N/A	External clock input. 12 MHz for MachXO3 and MachXO3D. 125 MHz for MachXO5-NX Connects to clk_i (Top Module).
rstn_i	Input	1	N/A	Active-low reset. Coonects to rstn_i (Top Module).
clkop_o	Output	1	N/A	25 MHz clock output.
lock_o	Output	1	1’b0	PLL lock. Connects to reset_n_i input of the reset timer.

Table 3.3. 25 MHz Generator Implementation

Target Device	Module Name	PLL Instance
MachXO3	pll_25MHz	upll
MachXO3D	pll_25MHz	upll
MachXO5-NX	pll_25MHz_xo5	upll_xo5

3.1.3. Reset Timer

This module utilizes both the PLL lock from the 25 MHz Generator (or external reset if 25 MHz Generator is not used) and the output of an internal watchdog timer to generate the system’s reset net. If the timer is disabled (timeout_en_i = 0), the output reset source is equal to the input reset.

Module Name: reset_timeout

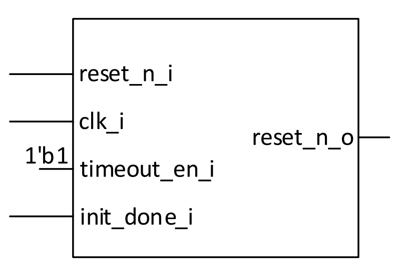

Figure 3.8. Reset Timer Block Diagram

Table 3.4. Reset Timer Signals

Port Name	Input/Output	Width	Default Value	Description
reset_n_i	Input	1	N/A	Active-low reset. It is connected to rstn_i (Top Module) if the 25 MHz generator is disabled. Otherwise, it is connected to lock_o of the 25 MHz generator.
clk_i	Input	1	N/A	25 MHz clock input.
timeout_en_i	Input	1	1'b1	Enables the timeout timer. The timer is active when timeout_en_i = 1 and init_done_i = 0.
init_done_i	Input	1	N/A	Connects to init_done_o of LTPI. The timer is active when timeout_en_i = 1 and init_done_i = 0.
reset_n_o	Output	1	1'b0	System reset source.

3.1.4. Initialization Logic

This module manages the PHY initialization of the LTPI and controls the signal to trigger the switch between Base Speed and Target Speed.

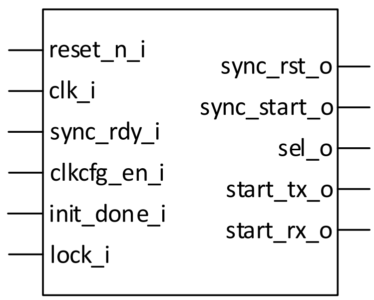

Figure 3.9. Initialization Logic Block Diagram

Table 3.5. Initialization Logic Signals

Port Name	Input/Output	Width	Default Value	Description
reset_n_i	Input	1	N/A	Active-low reset.
clk_i	Input	1	N/A	25 MHz clock input.
sync_rdy_i	Input	1	N/A	PHY synchronization ready. Connects to sync_rdy_o of LTPI.
clkcfg_en_i	Input	1	N/A	Clock Reconfiguration Enable. Connects to clkcfg_en_o of LTPI.
init_done_i	Input	1	N/A	Link Established Flag. Connects to init_done_o of LTPI.
lock_i	Input	1	N/A	Dynamic PLL Lock. Connects to LOCK or lock_o of LTPI Clock Source.
sync_rst_o	Output	1	1'b0	PHY reset. Connects to sync_rst_i of LTPI.
sync_start_o	Output	1	1'b0	PHY synchronization start trigger. Connects to sync_start_i of LTPI.
sel_o	Output	1	1'b0	Control signal for frequency change: 1'b0 = Base Speed 1'b1 = Target Speed Connects to sel_i of clock generator. Unused for MachXO3 and MachXO3D.
start_tx_o	Output	1	1'b0	Initialize Tx PHY. Connects to start_tx_i of LTPI.
start_rx_o	Output	1	1'b0	Initialize Rx PHY. Connects to start_rx_i of LTPI.

3.1.5. PLL Streamer Logic

This module manages the process of changing the LTPI Clock Source frequency from Base Speed to Target Speed and vice versa. Table 3.7 outlines the difference in implementation for the three devices.

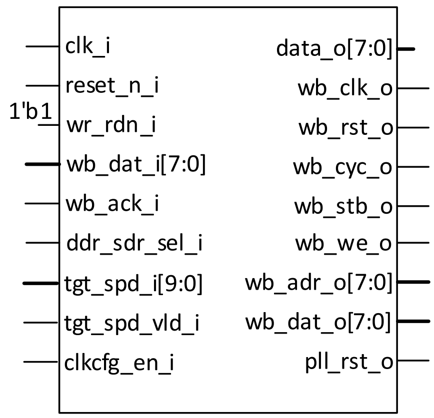

Figure 3.10. PLL Streamer Logic (Wishbone) Block Diagram

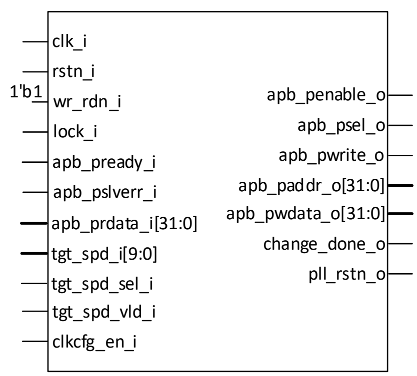

Figure 3.11. PLL Streamer Logic (APB) Block Diagram

Table 3.6. PLL Streamer Logic Implementation

Target Device	Module Name
MachXO3	wishbone_streamer
MachXO3D	wishbone_streamer
MachXO5-NX	apb_streamer

Table 3.7. PLL Streamer Logic Signals

Port Name	Input/Output	Width	Default Value	Description
Wishbone				
clk_i	Input	1	N/A	25 MHz clock input.
reset_n_i	Input	1	N/A	Active-low reset.
wr_rdn_i	Input	1	1’b1	Write (High) / Read (Low) Enable. **DO NOT CHANGE.**
wb_dat_i	Input	8	N/A	Wishbone Data In. Connects to wb_dat_o of EFB.
wb_ack_i	Input	1	N/A	Wishbone Ack Bit. Connects to wb_ack_o of EFB.

Port Name	Input/Output	Width	Default Value	Description
ddr_sdr_sel_i	Input	1	N/A	Flag for SDR/DDR selection: 1'b0 = SDR 1'b1 = DDR Connects to tgt_spd_o[15] of LTPI.
tgt_spd_i	Input	10	N/A	Status flag for Target Speed Selection: 10'h001 = 25 MHz 10'h002 = 50 MHz 10'h004 = 75 MHz 10'h008 = 100 MHz 10'h010 = 150 MHz 10'h020 = 200 MHz 10'h040 = 250 MHz 10'h080 = 300 MHz 10'h100 = 400 MHz Connects to tgt_spd_o[9:0] of LTPI.
tgt_spd_vld_i	Input	1	N/A	It indicates that the value on the tgt_spd_o of LTPI is valid. Connects to tgt_spd_vld_o of LTPI.
clkcfg_en_i	Input	1	N/A	Clock Reconfiguration Enable. Connects to clkcfg_en_o of LTPI.
data_o	Output	8	8'h00	Module debug port. Unused.
wb_clk_o	Output	1	1'b0	Wishbone Clock. Connects to wb_clk_i of EFB.
wb_rst_o	Output	1	1'b0	Wishbone Reset. Connects to wb_rst_i of EFB.
wb_cyc_o	Output	1	1'b0	Wishbone Cycle. Connects to wb_cyc_i of EFB.
wb_stb_o	Output	1	1'b0	Wishbone Strobe. Connects to wb_stb_i of EFB.
wb_we_o	Output	1	1'b0	Wishbone Write Enable. Connects to wb_we_i of EFB.
wb_adr_o	Output	8	8'h00	Wishbone Address. Connects to wb_adr_i of EFB.
wb_dat_o	Output	8	8'h00	Wishbone Data Out. Connects to wb_dat_i of EFB.
pll_rst_o	Output	1	1'b0	Dynamic PLL Reset Source. Connects to RST of LTPI Clock Source.
APB				
clk_i	Input	1	N/A	25 MHz clock input.
rstn_i	Input	1	N/A	Active-low reset.
wr_rdn_i	Input	1	1'b1	Write (High) / Read (Low) Enable. **DO NOT CHANGE.**
lock_i	Input	1	N/A	Dynamic PLL Lock. Connects to lock_o of LTPI Clock Source.
apb_pready_i	Input	1	N/A	APB Ready. Connects to apb_pready_o of LTPI Clock Source.
apb_pslverr_i	Input	1	N/A	APB Slave Error. Connects to apb_pslverr_o of LTPI Clock Source.
apb_prdata_i	Input	32	N/A	APB Read Data. Connects to apb_prdata_o of LTPI Clock Source.
tgt_spd_i	Input	10	N/A	Status flag for Target Speed Selection: 10'h001 = 25 MHz 10'h002 = 50 MHz 10'h004 = 75 MHz 10'h008 = 100 MHz 10'h010 = 150 MHz 10'h020 = 200 MHz 10'h040 = 250 MHz 10'h080 = 300 MHz 10'h100 = 400 MHz 10'h200 = 800 MHz Connects to tgt_spd_o[9:0] of LTPI.

Port Name	Input/Output	Width	Default Value	Description
ddr_sdr_sel_i	Input	1	N/A	Flag for SDR/DDR selection: 1’b0 = SDR 1’b1 = DDR Connects to tgt_spd_o[15] of LTPI.
tgt_spd_vld_i	Input	1	N/A	It indicates that the value on the tgt_spd_o of LTPI is valid. Connects to tgt_spd_vld_o of LTPI.
clkcfg_en_i	Input	1	N/A	Clock Reconfiguration Enable. Connects to clkcfg_en_o of LTPI.
apb_penable_o	Output	1	1’b0	APB Enable. Connects to apb_penable_i of LTPI Clock Source.
apb_psel_o	Output	1	1’b0	APB Select. Connects to apb_psel_i of LTPI Clock Source.
apb_pwrite_o	Output	1	1’b0	APB Write Enable. Connects to apb_pwrite_i of LTPI Clock Source.
apb_paddr_o	Output	32	32’h00000000	APB Address. Connects to apb_paddr_i of LTPI Clock Source.
apb_pwdata_o	Output	32	32’h00000000	APB Write Data. Connects to apb_pwdata_i of LTPI Clock Source.
change_done_o	Output	1	1’b0	Frequency Change Done Flag. Connects to change_done_i of Clock Generator.
pll_rstn_o	Output	1	1’b1	Dynamic PLL Reset Source. Connects to rstn_i of LTPI Clock Source.

3.1.6. EFB

This module serves as a wrapper for an EFB instance, providing access to the Wishbone interface of the LTPI Clock Source PLL. This module is used exclusively for MachXO3 and MachXO3D devices.

Module Name: wb_pll

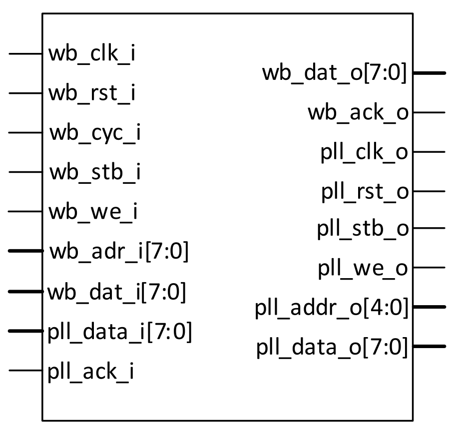

Figure 3.12. EFB Block Diagram

Table 3.8. EFB Signals

Port Name	Input/Output	Width	Default Value	Description
wb_clk_i	Input	1	N/A	Wishbone Clock. Connects to wb_clk_o of PLL Streamer Logic.
wb_rst_i	Input	1	N/A	Wishbone Reset. Connects to wb_rst_o of PLL Streamer Logic.
wb_cyc_i	Input	1	N/A	Wishbone Cycle. Connects to wb_cyc_o of PLL Streamer Logic.
wb_stb_i	Input	1	N/A	Wishbone Strobe. Connects to wb_stb_o of PLL Streamer Logic.
wb_we_i	Input	1	N/A	Wishbone Write Enable. Connects to wb_we_o of PLL Streamer Logic.
wb_adr_i	Input	8	N/A	Wishbone Address. Connects to wb_adr_o of PLL Streamer Logic.
wb_dat_i	Input	8	N/A	Wishbone Data In. Connects to wb_dat_o of PLL Streamer Logic.
pll_dat_i	Input	8	N/A	PLL Wishbone Data In. Connects to PLLDATO of LTPI Clock Source.
pll_ack_i	Input	1	N/A	PLL Wishbone Ack Bit. Connects to PLLACK of LTPI Clock Source.
wb_dat_o	Output	8	8'h00	Wishbone Data Out. Connects to wb_dat_i of PLL Streamer Logic.
wb_ack_o	Output	1	1'b0	Wishbone Ack Bit. Connects to wb_ack_i of PLL Streamer Logic.
pll_clk_o	Output	1	N/A	PLL Wishbone Clock. Connects to PLLCLK of LTPI Clock Source.
pll_rst_o	Output	1	1'b0	PLL Wishbone Reset. Connects to PLLRST of LTPI Clock Source.
pll_stb_o	Output	1	1'b0	PLL Wishbone Strobe. Connects to PLLSTB of LTPI Clock Source.
pll_we_o	Output	1	1'b0	PLL Wishbone Write Enable. Connects to PLLWE of LTPI Clock Source.
pll_addr_o	Output	5	5'h00	PLL Wishbone Address. Connects to PLLADDR of LTPI Clock Source.
pll_data_o	Output	8	8'h00	PLL Wishbone Data Out. Connects to PLLDATI of LTPI Clock Source.

3.1.7. LTPI Clock Source

This module serves as a wrapper for a PLL instance, providing the necessary clocks for the LTPI.

Table 3.3 outlines the difference in implementation for the three devices.

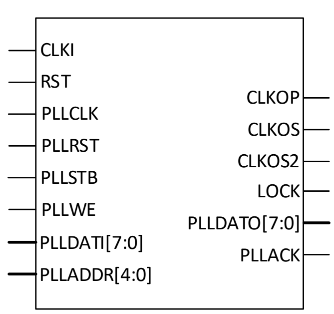

Figure 3.13. LTPI Clock Source (MachXO3/MachXO3D) Block Diagram

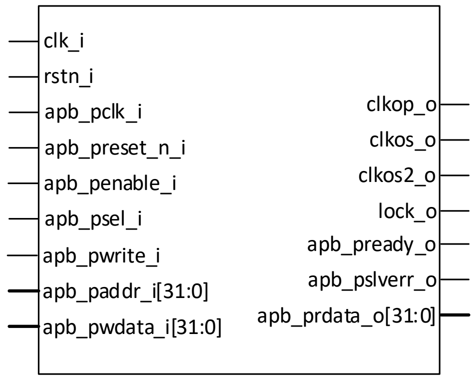

Figure 3.14. LTPI Clock Source (MachXO5-NX) Block Diagram

Table 3.9. LTPI Clock Source Signals

Port Name	Input/Output	Width	Default Value	Description
MachXO3/MachXO3D				
CLKI	Input	1	N/A	25 MHz clock input.
RST	Input	1	N/A	Active-high reset. Connects to pll_rst_o of PLL Streamer Logic.
PLLCLK	Input	1	N/A	PLL Wishbone Clock. Connects to pll_clk_o of LTPI Clock Source.
PLLRST	Input	1	N/A	PLL Wishbone Reset. Connects to pll_rst_o of LTPI Clock Source.
PLLSTB	Input	1	N/A	PLL Wishbone Strobe. Connects to pll_stb_o of LTPI Clock Source.
PLLWE	Input	1	N/A	PLL Wishbone Write Enable. Connects to pll_we_o of LTPI Clock Source.
PLLDATI	Input	8	N/A	PLL Wishbone Data In. Connects to pll_data _o of LTPI Clock Source.
PLLADDR	Input	5	N/A	PLL Wishbone Address. Connects to pll_addr_o of LTPI Clock Source.
CLKOP	Output	1	N/A	LTPI ECLK source. Connects to eclk_i of LTPI.
CLKOS	Output	1	N/A	LTPI ECLK 90 source. Connects to eclk90_i of LTPI.
CLKOS2	Output	1	N/A	LTPI main clock source. Connects to clk_i of LTPI.
LOCK	Output	1	1’b0	Dynamic PLL Lock. Connects to lock_i of Initialization Logic.
PLLDATO	Output	8	8’h00	PLL Wishbone Data Out. Connects to pll_dat_i of EFB.
PLLACK	Output	1	1’b0	PLL Wishbone Ack Bit. Connects to pll_ack_i of EFB.
MachXO5-NX				
clk_i	Input	1	N/A	25 MHz clock input.
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	25 MHz clock input.
apb_preset_n_i	Input	1	N/A	Active-low reset.
apb_penable_i	Input	1	N/A	APB Enable. Connects to apb_penable_o of PLL Streamer Logic.
apb_psel_i	Input	1	N/A	APB Select. Connects to apb_psel_o of PLL Streamer Logic.
apb_pwrite_i	Input	1	N/A	APB Write Enable. Connects to apb_pwrite_o of PLL Streamer Logic.
apb_paddr_i	Input	32	N/A	APB Address. Connects to apb_paddr_o of PLL Streamer Logic.

Port Name	Input/Output	Width	Default Value	Description
apb_pwdata_i	Input	32	N/A	APB Write Data. Connects to apb_pwdata_o of PLL Streamer Logic.
clkop_o	Output	1	N/A	LTPI ECLK source. Connects to eclk_i of LTPI.
clkos_o	Output	1	N/A	LTPI ECLK 90 source. Connects to eclk90_i of LTPI.
clkos2_o	Output	1	N/A	LTPI Target Speed System Clock. Connects to clk_systarget_i of Clock Generator.
lock_o	Output	1	1'b0	Dynamic PLL Lock. Connects to lock_i of Initialization Logic and lock_i of Clock Generator.
apb_pready_o	Output	1	1'b0	APB Ready. Connects to apb_pready_i of PLL Streamer Logic.
apb_pslverr_o	Output	1	1'b0	APB Slave Error. Connects to apb_pslverr_i of PLL Streamer Logic.
apb_prdata_o	Output	32	32'h00000000	APB Read Data. Connects to apb_prdata_i of PLL Streamer Logic.

Table 3.10. LTPI Clock Source Implementation

Target Device	Module Name	PLL Instance
MachXO3	dynamic_pll	dpll
MachXO3D	dynamic_pll	dpll
MachXO5-NX	dynamic_pll_xo5	dpll_xo5

3.1.8. Clock Generator

This module generates a 2.5 MHz clock that cannot be natively produced by the LTPI Clock Source. It also includes the switch logic to toggle between this 2.5 MHz clock and clkos2_o, the output of the LTPI Clock Source, to generate the required input clock, clk_i, for LTPI. This module is exclusively used for MachXO5-NX devices.

Module Name: clock_generator

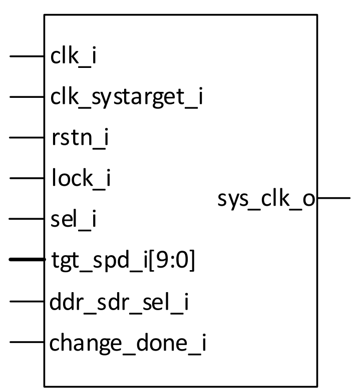

Figure 3.15. Clock Generator Block Diagram

Table 3.11. Clock Generator Signals

Port Name	Input/Output	Width	Default Value	Description
clk_i	Input	1	N/A	25 MHz clock input.
clk_systarget_i	Input	1	N/A	LTPI Target Speed System Clock. Connects to clkos2_o port of LTPI Clock Source.
rstn_i	Input	1	N/A	Active-low reset.
lock_i	Input	1	N/A	Dynamic PLL Lock. Connects to lock_o of LTPI Clock Source.

Port Name	Input/Output	Width	Default Value	Description
sel_i	Input	1	N/A	Control signal for frequency change: 1'b0 = Base Speed 1'b1 = Target Speed Connects to sel_i of Initialization Logic.
ddr_sdr_sel_i	Input	1	N/A	Flag for SDR/DDR selection: 1'b0 = SDR 1'b1 = DDR Connects to tgt_spd_o[15] of LTPI.
tgt_spd_i	Input	10	N/A	Status flag for Target Speed Selection: 10'h001 = 25 MHz 10'h002 = 50 MHz 10'h004 = 75 MHz 10'h008 = 100 MHz 10'h010 = 150 MHz 10'h020 = 200 MHz 10'h040 = 250 MHz 10'h080 = 300 MHz 10'h100 = 400 MHz 10'h200 = 800 MHz Connects to tgt_spd_o[9:0] of LTPI.
change_done_i	Input	1	N/A	Frequency Change Done Flag. Connects to change_done_o of PLL Streamer Logic.
sys_clk_o	Output	1	N/A	LTPI main clock source. Connects to clk_i of LTPI.

3.1.9. APB Interconnect (SCM Only)

This module utilizes the APB Interconnect Module from Lattice Propel. It arbitrates APB commands from the I2C-to-APB Bridge to either the CSR APB Completer Interface or the Data Channel APB Completer Interface of LTPI.

Module Name: apb0

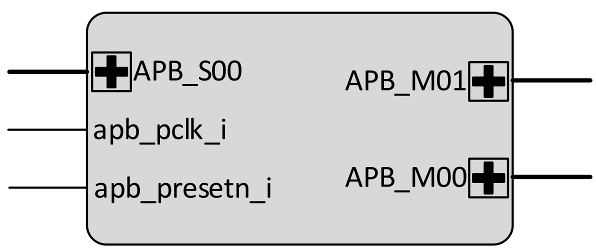

Figure 3.16. APB Interconnect Block Diagram

Table 3.12. APB Interconnect Signals

Port Name	Input/Output	Width	Default Value	Description
APB_S00	APB IF	N/A	N/A	APB Completer Interface 0. Connects to APB_M0 of I2C-to-APB Bridge.
apb_pclk_i	Input	1	N/A	25 MHz clock input.
apb_presetn_i	Input	1	N/A	Active-low reset.
APB_M00	APB IF	N/A	N/A	APB Requester Interface 0. Connects to APB_S1 of I2C-to-APB Bridge.
APB_M01	APB IF	N/A	N/A	APB Requester Interface 1. Connects to APB_S0 of I2C-to-APB Bridge.

3.1.10. APB Feedthrough (HPM Only)

This module utilizes the APB Feedthrough Module from Lattice Propel. It arbitrates APB commands from the Data Channel APB Requester Interface to the CSR APB Completer Interface of LTPI.

Module Name: apbf0

Figure 3.17. APB Feedthrough Block Diagram

Table 3.13. APB Feedthrough Signals

Port Name	Input/Output	Width	Default Value	Description
APB_S0	APB IF	N/A	N/A	APB Completer Interface 0. Connects to APB_M0 of LTPI.
APB_M0	APB IF	N/A	N/A	APB Requester Interface 0. Connects to APB_S0 of LTPI.

3.1.11. I2C-to-APB Bridge (SCM Only)

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

Module Name: i2c2apb0

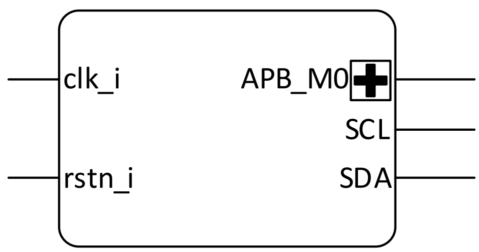

Figure 3.18. I2C-to-APB Bridge Block Diagram

Table 3.14. I2C-to-APB Bridge Signals

Port Name	Input/Output	Width	Default Value	Description
clk_i	Input	1	N/A	25 MHz clock input.
rstn_i	Input	1	N/A	Active-low reset.
APB_M0	APB IF	N/A	N/A	APB Requester Interface 0. Connects to APB_S00 of APB Interconnect.
SCL	Inout	1	Tri-state	I2C Clock port. Connects to i2c2apb0_inst_SCL_port (Top Module).
SDA	Inout	1	Tri-state	I2C Data port. Connects to i2c2apb0_inst_SDA_port (Top Module).

3.2. Clocking Scheme

The reference design uses a 25 MHz generator (via PLL) that accepts either a 12 MHz input for MachXO3 and MachXO3D devices or a 125 MHz input for MachXO5-NX devices.

The input values are determined by the available external oscillators on each device's boards, which are used as the target hardware for this design.

The 25 MHz source is required to generate the necessary clocks for the DC-SCM LTPI IP (ECLK, ECLK90, and CLK). This allows one set of computed values to generate the supported frequencies as defined by the LTPI specifications, up to 400 MHz for MachXO3 and MachXO3D devices and up to 800 MHz for MachXO5-NX devices.

If an external 25 MHz source is available, the 25 MHz generator can be removed, connecting the external clock directly to the system and the rstn_i port to the reset timer module instead of the PLL's lock port.

3.3. Reset Scheme

The reference design uses the Reset Timer module to generate the system's reset net.

Note that the LTPI Clock Source module does not use the system reset. Its reset is provided by the PLL streamer logic, allowing the module to toggle the PLL's output when changing from Base Speed to Target Speed and vice versa.

3.4. DC-SCM LTPI IP Control and Status Registers

This information is based on section 2.4 of the DC-SCM LTPI IP User Guide (FPGA-IPUG-02200). Any changes made by future IP updates may not be reflected immediately in this document.

Table 3.15. DC-SCM LTPI IP Registers

Offset	Register Name	Access	Default Value	Register Description		
				Bit	**Field**	**Field Description**
Attribute Registers						
0x0	ID	RO	{depends on ID Number set in Attributes}	[31:16]	RSVD	Reserved bits.
				[15:0]	id	Identification Number.
0x4	LTPI_VER	RO	{depends on *LTPI Version (Major)* and *LTPI Version (Minor)* set in Attributes}	[31:8]	RSVD	Reserved bits.
				[7:0]	ltpi_ver	Specifies the LTPI Version. [7:4] – Major version [3:0] – Minor version
0x8	SPEED_CAP	RO	{depends on Speed Capability set in Attributes}	[31:16]	RSVD	Reserved bits.
				[15:0]	speed_cap	Defines the speed capability of the IP. Base frequency is 25 MHz. [7:0] = {X12,X10,X8,X6,X4,X3,X2,X1} [15:8] = {DDR,3'h00,X40,X32,X24,X16}
0x0C	PLATFORM_TYPE	RO	{depends on ID Number set in Attributes}	[31:16]	RSVD	Reserved bits.
				[15:0]	platform_type	Platform type.
0x10	FEATURE_CAP3_0	RO	{depends on IP and Channel Capability set in Attributes}	[31:0]	feature_cap3_0	Feature capability (byte3-0) of the IP. If *Enable Full OEM Capabilities Type* == Checked, it gets value from *OEM Feature Capability 0 in Hex (0x)*.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	**Field**	**Field Description**
0x14	FEATURE_CAP7_4	RO	{depends on IP and Channel Capability set in Attributes}	[31:0]	feature_cap7_4	Feature capability (byte7-4) of the IP. If *Enable Full OEM Capabilities Type* == Checked, it gets value from *OEM Feature Capability 1 in Hex (0x)*.
0x18	FEATURE_CAP11_8	RO	32'h0	[31:0]	feature_cap11_8	Feature capability (byte11-8) of the IP. Currently is unused and only serves as a placeholder for future use.
0x1C	REQ_FEATURE3_0	RW	{depends on Feature Capability 0 set in Attributes}	[31:0]	req_feature3_0	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded.
0x20	REQ_FEATURE7_4	RW	{depends on Feature Capability 1 set in Attributes}	[31:0]	req_feature7_4	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded.
0x24	REQ_FEATURE 11_8	RW	32'h0	[31:0]	req_feature11_8	Request features of the IP when *IP MODE == SCM*. When *IP MODE == HPM*, this register is discarded. Currently is unused and only serves as a placeholder for future use.
0x28	TGT_SPEED	RO	32'h0	[31:16]	RSVD	Reserved bits.
				[15:0]	tgt_spd	Defines the target speed of the IP based on Link training and negotiation. Base frequency is 25 MHz. [7:0] = {X12,X10,X8,X6,X4,X3,X2,X1} [15:8] = {DDR,3'h00,X40,X32,X24,X16}
0x2C	RX_FEATURE3_0	RO	32'h0	[31:0]	rx_feat_cap3_0	Received Feature Capability from Remote IP during Link-training and negotiation.
0x30	RX_FEATURE7_4	RO	32'h0	[31:0]	rx_feat_cap7_4	Received Feature Capability from Remote IP during Link-training and negotiation.
0x34	RX_FEATURE 11_8	RO	32'h0	[31:0]	rx_feat_cap11_8	Received Feature Capability from Remote IP during Link-training and negotiation.

<table>
<tr><th rowspan="2">Offset</th><th rowspan="2">Register Name</th><th rowspan="2">Access</th><th rowspan="2">Default Value</th><th colspan="3">Register Description</th></tr>
<tr><th>Bit</th><th>Field</th><th>Field Description</th></tr>
<tr><td colspan="7">IP Control</td></tr>
<tr><td rowspan="7">0x38</td><td rowspan="7">IP_CTRL</td><td rowspan="3">RO</td><td rowspan="7">{23’h0, depends on Automatically move to Configuration State set in Attributes, 8’h0}</td><td>[31:9]</td><td>RSVD</td><td>Reserved bits.</td></tr>
<tr><td>[8]</td><td>auto_cfgen</td><td>When set and IP Mode == SCM, the IP automatically goes to Configuration state after completing the required frame transmission for Advertise State.</td></tr>
<tr><td>[7:4]</td><td>RSVD</td><td>Reserved bits.</td></tr>
<tr><td rowspan="4">RW</td><td>[3]</td><td>data_ch_rst</td><td>Data channel reset register.
When asserted, resets the data channel interface controller of the IP.
When IP Mode==HPM, this also resets the APB requester interface output ports.</td></tr>
<tr><td>[2]</td><td>software_rst</td><td>LTPI link software reset. Reset only applies to interface controller and Link-training and negotiation FSM-related logic.</td></tr>
<tr><td>[1]</td><td>resync_link</td><td>Link retraining request register. When asserted, internally resets the IP except for IP CSR related interface. The IP goes back to Link-training state. Needs to be unset to proceed with retraining after active trigger.</td></tr>
<tr><td>[0]</td><td>reqcfg_rdy</td><td>Used to indicate access to CSR is done and requested features (for SCM) is ready for Advertise state. Bit is only checked during Advertise state and automatically move to Configuration State==Unchecked during IP configuration.
Only when this is set that SCM goes to Configuration State.
When IP Mode==HPM, this register is discarded.</td></tr>
<tr><td rowspan="2">0x3C</td><td rowspan="2">I2C_BUS_RST</td><td>RO</td><td rowspan="2">32’h0</td><td>[31:24]</td><td>RSVD</td><td>Reserved bits.</td></tr>
<tr><td>RW</td><td>[23:0]</td><td>i2c_bus_rst</td><td>I2C bus controller reset. This can be used to perform reset and recovery of I2C buses during timeout or bus hang-up.
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</td></tr>
</table>

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
IP Settings						
0x40	SP_SYMBOL	RO	K28.7 (8'hFC)	[31:8]	RSVD	Reserved bits.
				[7:0]	sp_symbol	Symbol used to indicate the start of normal frames (L0 state).
0x44	LT_SYMBOL	RO	K28.5 (8'hBC)	[31:8]	RSVD	Reserved bits.
				[7:0]	lt_symbol	Symbol used to indicate start of Link Training related frames (Link-Detect and Link-Speed states).
0x48	CFG_SYMBOL	RO	K28.6 (8'hDC)	[31:8]	RSVD	Reserved bits.
				[7:0]	cfg_symbol	Symbol used to indicate start of Advertise, Configure, and Accept frames.
0x4C	LDFT_SYMBOL	RO	D0.0 (8'h00)	[31:8]	RSVD	Reserved bits.
				[7:0]	ldft_symbol	Symbol used to indicate frame type for Link-Detect frames during Link Training.
0x50	LSFT_SYMBOL	RO	D1.0 (8'h01)	[31:8]	RSVD	Reserved bits.
				[7:0]	lsft_symbol	Symbol used to indicate frame type for Link-Speed frames during Link Training.
0x54	IDLE_SYMBOL	RO	D0.0 (8'h00)	[31:8]	RSVD	Reserved bits.
				[7:0]	idle_symbol	Symbol used to indicate IDLE bytes for Link-Detect and Link-Speed frames during Link Training.
0x58	ADVFT_SYMBOL	RO	D0.0 (8'h00)	[31:8]	RSVD	Reserved bits.
				[7:0]	advft_symbol	Symbol used to indicate frame type for Advertise frames during Link Configuration.
0x5C	REQFT_SYMBOL	RO	D1.0 (8'h01)	[31:8]	RSVD	Reserved bits.
				[7:0]	reqft_symbol	Symbol used to indicate frame type for Request Configuration frames for SCM during Link Configuration.
0x60	ACCFT_SYMBOL	RO	D2.0 (8'h02)	[31:8]	RSVD	Reserved bits.
				[7:0]	accft_symbol	Symbol used to indicate frame type for Accept frames for HPM during Link Configuration.
0x64	DFT_SYMBOL	RO	24’h0, depends on *Data Frame Type* set in Attributes	[31:8]	RSVD	Reserved bits.
				[7:0]	dft_symbol	Symbol used to indicate frame type for Default or Custom I/O Frame. 0x00 – Default I/O 0x10 – Custom I/O
0x68	DCFT_SYMBOL	RO	D1.0 (8'h01)	[31:8]	RSVD	Reserved bits.
				[7:0]	dcft_symbol	Symbol used to indicate frame type for Data Frame.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
IP Debug Registers						
0x6C	LINK_STATE_TX	RO	32'h1	[31:7]	RSVD	Reserved bits.
				[6:0]	link_state_tx	Local LTPI link state. 0x01 – Idle/Link-Detect –Frame Align 0x02 – Link-Detect 0x04 – Link-Speed 0x08 – Advertise 0x10 – Configuration 0x20 – Accept 0x40 – L0 (Operational)
0x70	LINK_STATE_RX	RO	32'h1	[31:7]	RSVD	Reserved bits.
				[6:0]	link_state_rx	Remote LTPI link state. 0x01 – Idle/Link-Detect –Frame Align 0x02 – Link-Detect 0x04 – Link-Speed 0x08 – Advertise 0x10 – Configuration 0x20 – Accept 0x40 – L0 (Operational)
Interrupt Registers						
0x74	INT_STATUS	RW1C	32'h0	[31:13]	RSVD	Reserved bits.
				[12]	i2c_code_err_int	I2C invalid code error. Asserts if invalid I2C event is observed in any of the active I2C buses.
				[11]	i2c_tmr_to_int	I2C timeout. Asserts if any of the active buses encounter timeout.
				[10]	acpt_to_int	Accept state timeout.
				[9]	cfg_to_int	Configuration state timeout.
				[8]	ls_to_int	Link-Speed state timeout.
				[7]	rx_feat_vld_int	Received feature capability is valid.
				[6]	clk_cfg_int	Clock can be reconfigured for target frequency based. This is asserted every time the IP is in Advertise state.
				[5]	tgt_spd_vld_int	Target speed information from link training and negotiation is already valid.
				[4]	link_err_int	Link error is encountered during normal operation.
				[3]	rxfifo_full_int	Rx FIFO full asserted unexpectedly during normal operation.
				[2]	txfifo_full_int	Tx FIFO full asserted unexpectedly during normal operation.
				[1]	lol_rx_int	Loss of enable: detected deassertion of start_rx_i.
				[0]	lol_tx_int	Loss of enable: detected deassertion of start_tx_i.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x78	INT_ENABLE	RW	32’h0	[31:13]	RSVD	Reserved bits.
				[12]	i2c_code_err_en	I2C invalid code error – interrupt enable.
				[11]	i2c_tmr_to_en	I2C timeout – interrupt enable.
				[10]	acpt_to_en	Accept state timeout – interrupt enable.
				[9]	cfg_to_en	Configuration state timeout – interrupt enable.
				[8]	ls_to_en	Link-Speed state timeout – interrupt enable.
				[7]	rx_feat_vld_en	Received feature capability is valid – interrupt enable.
				[6]	clk_cfg_en	Clock can be reconfigured for target frequency based – interrupt enable.
				[5]	tgt_spd_vld_en	Target speed information from link training and negotiation is already valid – interrupt enable.
				[4]	link_err_en	Link error is encountered during normal operation – interrupt enable.
				[3]	rxfifo_full_en	Rx FIFO full asserted unexpectedly during normal operation – interrupt enable.
				[2]	txfifo_full_en	Tx FIFO full asserted unexpectedly during normal operation – interrupt enable.
				[1]	lol_rx_en	Loss of enable: detected. deassertion of start_rx_i – interrupt enable.
				[0]	lol_tx_en	Loss of enable: detected. deassertion of start_tx_i – interrupt enable.

Offset	Register Name	Access	Default Value	Register Description		
				Bit	Field	Field Description
0x7C	INT_SET	WO	32'h0	[31:13]	RSVD	Reserved bits.
				[12]	i2c_code_err_set	I2C invalid code error – set.
				[11]	i2c_tmr_to_set	I2C timeout – set.
				[10]	acpt_to_set	Accept state timeout – set.
				[9]	cfg_to_set	Configuration state timeout – set.
				[8]	ls_to_set	Link-Speed state timeout – set.
				[7]	rx_feat_vld_set	Received feature capability is valid – set.
				[6]	clk_cfg_set	Clock can be reconfigured for target frequency based – set.
				[5]	tgt_spd_vld_set	Target speed information from link training and negotiation is already valid – set.
				[4]	link_err_set	Link error is encountered during normal operation – set.
				[3]	rxfifo_full_set	Rx FIFO full asserted unexpectedly during normal operation – set.
				[2]	txfifo_full_set	Tx FIFO full asserted unexpectedly during normal operation – set.
				[1]	lol_rx_set	Loss of enable: detected. deassertion of start_rx_i – set.
				[0]	lol_tx_set	Loss of enable: detected. deassertion of start_tx_i – set.

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). We measure this 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, which are shown in Table 4.1 for SCM and Table 4.2 for HPM.

Table 4.1. SCM Top Module Signals

Port Name	Input/Output	Width	Default Value	Description
System				
clk_i	Input	1	N/A	External clock input. 12 MHz for MachXO3 and MachXO3D. 125 MHz for MachXO5-NX.
rstn_i	Input	1	N/A	Active-low reset. This is used to reset the 25 MHz Generator module.
LVDS Interface				
scm0_inst_lvds_tx_clk_o_port	Output	1	N/A	LVDS Tx PHY clock
scm0_inst_lvds_tx_data_o_port	Output	1	N/A	LVDS Tx PHY data
scm0_inst_lvds_rx_clk_i_port	Input	1	N/A	LVDS Rx PHY clock
scm0_inst_lvds_rx_data_i_port	Input	1	N/A	LVDS Rx PHY data
Low Latency GPIO Channel Interface				
scm0_inst_ll_gpio_i_portbus	Input	4	N/A	Low Latency Input pins
scm0_inst_ll_gpio_o_portbus	Output	4	N/A	Low Latency Output pins
Normal Latency GPIO Channel Interface				
scm0_inst_nl_gpio_i_portbus	Input	4	N/A	Normal Latency Input pins
scm0_inst_nl_gpio_o_portbus	Output	4	N/A	Normal Latency Output pins
UART Channel Interface				
scm0_inst_uart_tx_i_portbus	Input	1	N/A	UART Tx pin
scm0_inst_uart_rx_o_portbus	Output	1	N/A	UART Rx pin
I2C Channel Interface				
scm0_inst_i2c_scl_io_portbus	Inout	1	Tri-state	I2C Clock pin. Requires external pull-up.
scm0_inst_i2c_sda_io_portbus	Inout	1	Tri-state	I2C Data pin. Requires external pull-up.
I2C-to-APB Interface				
i2c2apb0_inst_SCL_port	Inout	1	Tri-state	I2C-to-APB Clock pin. Requires external pull-up.
i2c2apb0_inst_SDA_port	Inout	1	Tri-state	I2C-to-APB Data pin. Requires external pull-up.
Protocol Information				
concat_module_inst_status_display_portbus	Output	8	1	Status Flags from LTPI: Bit 0: init_done_o Bit 1: rx_align_stat_o Bit 2: ls_to_o Bit 3: cfg_to_o Bit 4: i2c_tmr_to_o Bit 5: i2c_code_err_o Bit 6: crc_err_o Bit 7: link_error_o

Table 4.2. HPM Top Module Signals

Port Name	Input/Output	Width	Default Value	Description
System				
clk_i	Input	1	N/A	External clock input. 12 MHz for MachXO3 and MachXO3D. 125 MHz for MachXO5-NX.
rstn_i	Input	1	N/A	Active-low reset. This is used to reset the 25 MHz Generator module.
LVDS Interface				
hpm0_inst_lvds_tx_clk_o_port	Output	1	N/A	LVDS Tx PHY clock
hpm0_inst_lvds_tx_data_o_port	Output	1	N/A	LVDS Tx PHY data
hpm0_inst_lvds_rx_clk_i_port	Input	1	N/A	LVDS Rx PHY clock
hpm0_inst_lvds_rx_data_i_port	Input	1	N/A	LVDS Rx PHY data
Low Latency GPIO Channel Interface				
hpm0_inst_ll_gpio_i_portbus	Input	4	N/A	Low latency GPIO input pins
hpm0_inst_ll_gpio_o_portbus	Output	4	N/A	Low latency GPIO output pins
Normal Latency GPIO Channel Interface				
hpm0_inst_nl_gpio_i_portbus	Input	4	N/A	Normal latency GPIO input pins.
hpm0_inst_nl_gpio_o_portbus	Output	4	N/A	Normal latency GPIO output pins.
UART Channel Interface				
hpm0_inst_uart_tx_i_portbus	Input	1	N/A	UART Tx pin
hpm0_inst_uart_rx_o_portbus	Output	1	N/A	UART Rx pin
I2C Channel Interface				
hpm0_inst_i2c_scl_io_portbus	Inout	1	N/A	I2C Clock pin. Requires external pull-up.
hpm0_inst_i2c_sda_io_portbus	Inout	1	N/A	I2C Data pin. Requires external pull-up.
Protocol Information				
concat_module_inst_status_display_portbus	Output	8	1	Status Flags from LTPI IP: Bit 0: init_done_o Bit 1: rx_align_stat_o Bit 2: ls_to_o Bit 3: acpt_to_o Bit 4: i2c_tmr_to_o Bit 5: i2c_code_err_o Bit 6: crc_err_o Bit 7: link_error_o

5. Timing Constraints

For timing constraint guidelines, refer to the Timing Analysis for High-Speed GDDR Interfaces' section of the Implementing High-Speed Interfaces with MachXO3 Devices (FPGA-TN-02057) for MachXO3 family, Implementing High-Speed Interfaces with MachXO3D Usage Guide (FPGA-TN-02065) for MachXO3D family, and MachXO5-NX High-Speed I/O Interface (FPGA-TN-02286) for the MachXO5-NX family.

5.1. 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.

- External Clock
    - 12 MHz for MachXO3/MachXO3D
    - 125 MHz for MachXO5-NX
- clk_i : 20 MHz
- sync_clk_i : 25 MHz
- eclk_i : 100 MHz
- eclk90_i : 100 MHz
- apb_pclk_i : 25 MHz
- lvds_rx_clk_i : 100 MHz
- phytx_sclk : 20 MHz
- ddrtx_sclk : 20 MHz
- pclkdiv_out : 10 MHz (used for SDR)
- sys_clk : 20 MHz
- ddrrx_sclk : 20 MHz

5.2. 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 inside the IP for the following clock sources and can be set as false paths as well.

- clk_i
- sync_clk_i
- eclk_i
- eclk90_i
- apb_pclk_i
- lvds_rx_clk_i
- phytx_sclk
- ddrtx_sclk
- pclkdiv_out
- sys_clk
- ddrrx_sclk

6. 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.

6.1. 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).

6.1.1. 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**.

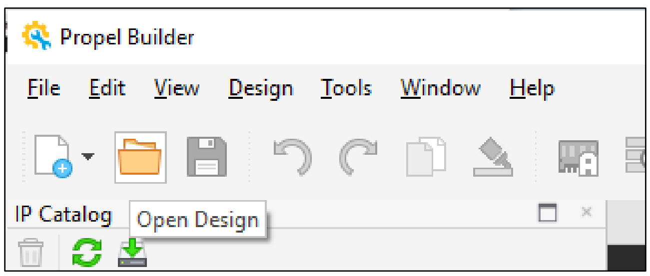

Figure 6.1. Open Design Button

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**.

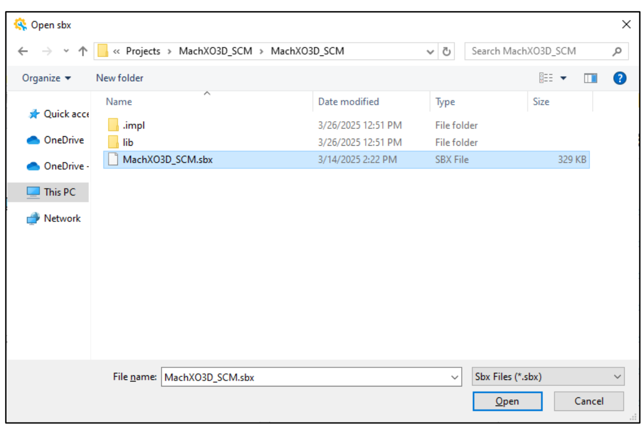

Figure 6.2. Selecting the Lattice Propel™ Project

6.1.2. Add/Remove Top-Level Module Ports

All module ports that are connected to an external port (input, output or inout) will be 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.

6.1.3. 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 will discuss 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's 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**.

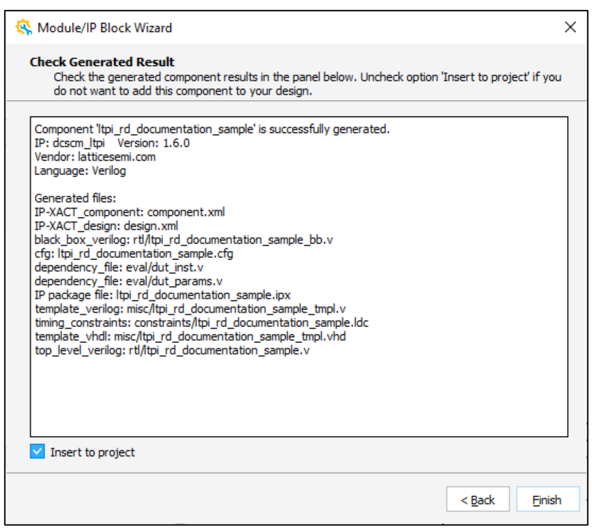

Figure 6.3. Check Generated Result Window

6.1.3.1. General Tab

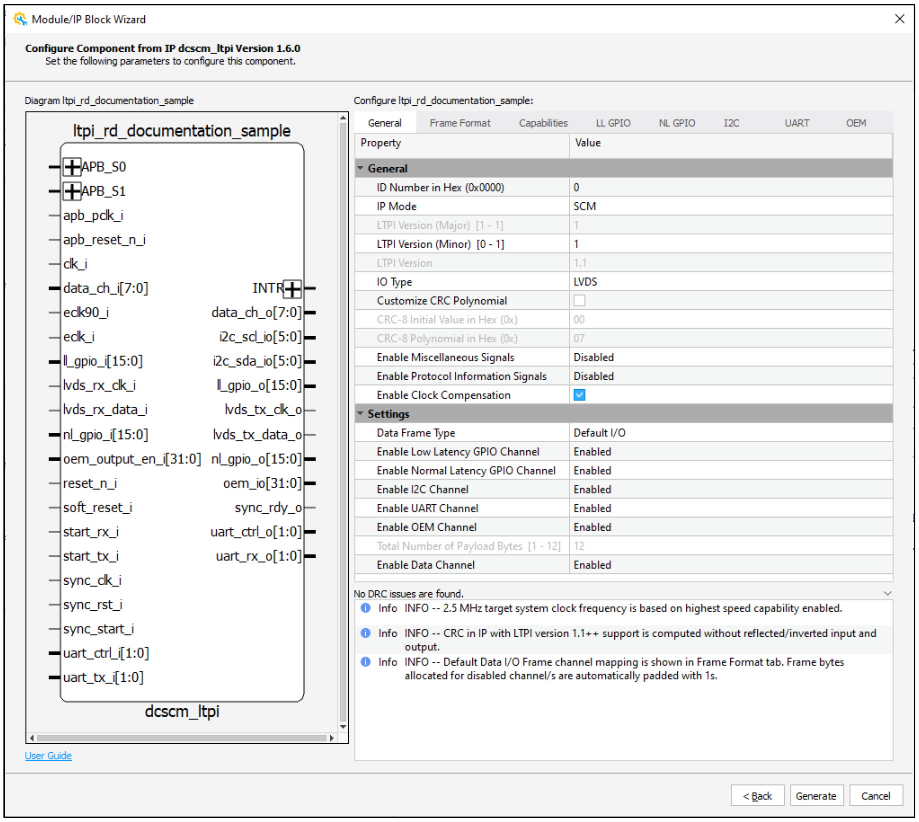

Figure 6.4. General Tab

General Section

- ID Number in Hex (0x0000)
  - This option is set to 0xCA26 for SCM and 0xBF17 for HPM. This is set to create a difference between the contents of the SCM and HPM CSR, which is then used to validate CSR access.
  - This can be removed or changed as necessary.
  - Changes to this option will be written at offset 0x00 of the CSR (ID).
- IP Mode
  - This option determines the function of the IP.
  - When modifying the reference design, do not change this option unless it is necessary.
- LTPI Version (Minor)
  - This option can be set to either 0 or 1. It sets the IP to support either LTPI version 1.0 or version 1.1.
  - It is recommended to use version 1.1 as it is the latest (this is the default setting).
  - There is no backward compatibility with version 1.0 since the change redefined how the CRC is computed in the frames, along with other changes.
  - Do not change this option unless it is necessary.
- IO Type
  - This is a placeholder option in case LTPI supports other IO types in future versions. Do not change.
- Customize CRC Polynomial
  - This is a placeholder option in case LTPI supports other CRC polynomials in future versions. Do not change.

- Enable Miscellaneous Signals
  - This option creates ports for various status flags of the IP, including init_done_o and link_err_o.
  - This option should be enabled for the reference design to work.
  - It is possible to gain access to these flags via CSR.
- Enable Protocol Information Signals
  - This option creates ports for various protocol signals of the IP, including tgt_spd_o and clkcfg_en_o.
  - This option should be enabled for the reference design to work.
  - It is possible to gain access to these signals via CSR.
- Enable Clock Compensation
  - This option allows the IP to perform frame drop or frame repeat if its FIFO experiences an underflow or overflow. This ensures the link remains stable even if there is a recoverable difference between the SCM and HPM clocks.
  - This option should not be changed to ensure link stability.

Settings Section

- Data Frame Type
  - This option allows modification of the frame format if set to Custom.
  - The reference design uses the default setting for this option (Default I/O).
- Enable Low Latency GPIO Channel
  - This option enables or disables the Low Latency GPIO Channel.
  - This option is enabled in the reference design.
  - When disabled, this option will remove the LL GPIO ports from the IP and the top-level module
- Enable Normal Latency GPIO Channel
  - This option enables or disables the Low Latency GPIO Channel.
  - This option is enabled in the reference design.
  - When disabled, this option will remove the LL GPIO ports from the IP and the top-level module.
- Enable I2C Channel
  - This option enables or disables the Low Latency GPIO Channel.
  - This option is enabled in the reference design.
  - When disabled, this option will remove the LL GPIO ports from the IP and the top-level module.
- Enable UART Channel
  - This option enables or disables the UART Channel.
  - This option is enabled in the reference design.
  - When disabled, this option will remove the LL GPIO ports from the IP and the top-level module.
- Enable OEM Channel
  - This option enables or disables the OEM Channel.
  - This option is disabled in the reference design.
- Enable Data Channel
  - This option enables or disables the Data Channel.
  - This option is enabled in the reference design.
  - When disabled, this option will remove the APB_S1 interface (SCM) or APB_M0 interface (HPM). This effectively disconnects the HPM CSR from access via the I2C-to-APB Bridge.
  - Disabling this option would also remove data_ch_i and data_ch_o from the IP ports. However, these are already unused in the reference design.

6.1.3.2. Frame Format Tab

By default, this is not editable. This can only be modified by setting Data Frame Type to Custom. The default setting adheres to the Default I/O Frame Format as defined by the LTPI Specification.

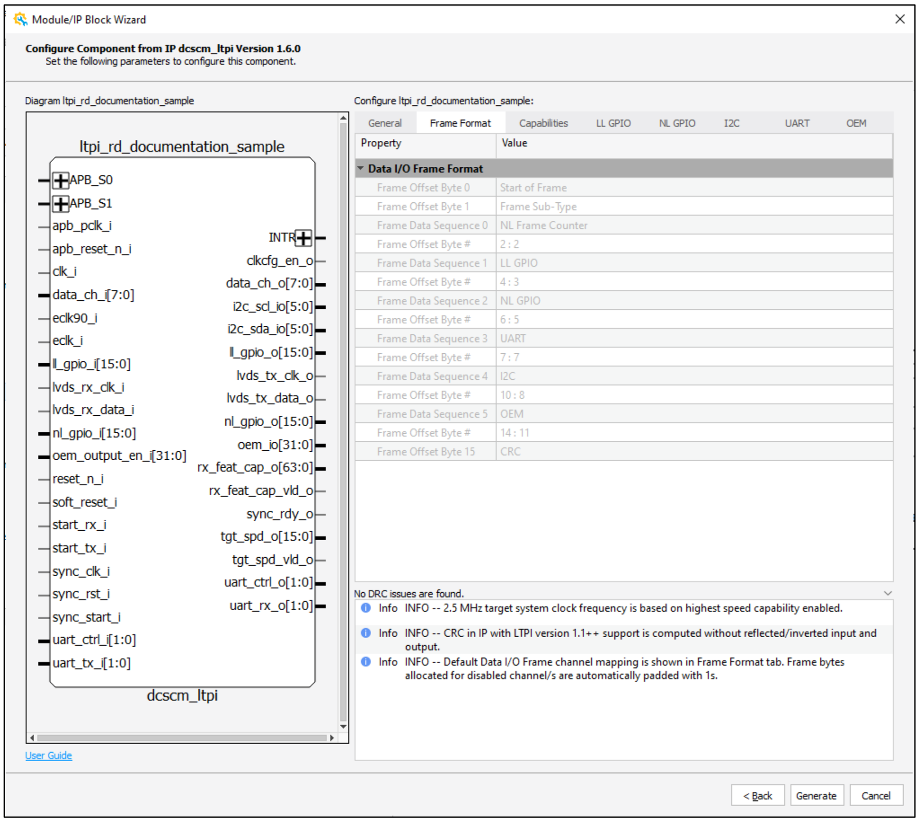

Figure 6.5. Frame Format Tab

6.1.3.3. Capabilities Tab

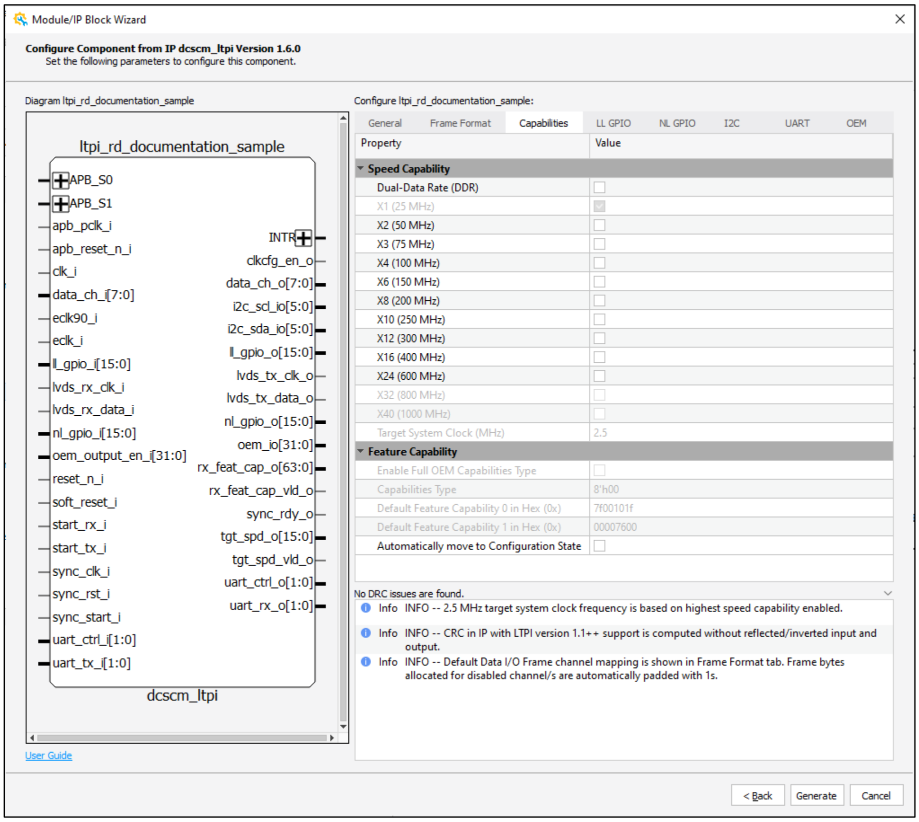

Figure 6.6. Capabilities Tab

Speed Capability Section

Changes to the options in this section will be recorded at offset 0x08 of the CSR (SPEED_CAP). Multiple options can be enabled simultaneously.

- Dual-Data Rate (DDR)
  - SPEED_CAP[15] = 1 when enabled. SPEED_CAP[15] = 0 when disabled.
  - This setting is enabled in the reference design.
- X1 (25 MHz)
  - This setting is always enabled. Mapped to SPEED_CAP[0].
- X2 (50 MHz)
  - SPEED_CAP[1] = 1 when enabled. SPEED_CAP[1] = 0 when disabled.
  - This setting is disabled in the reference design.
- X3 (50 MHz)
  - SPEED_CAP[2] = 1 when enabled. SPEED_CAP[2] = 0 when disabled.
  - This setting is disabled in the reference design.
- X4 (100 MHz)
  - SPEED_CAP[3] = 1 when enabled. SPEED_CAP[3] = 0 when disabled.
  - This setting is enabled in the reference design.
- X6 (150 MHz)
  - SPEED_CAP[4] = 1 when enabled. SPEED_CAP[4] = 0 when disabled.
  - This setting is disabled in the reference design.

- X8 (200 MHz)
  - SPEED_CAP[5] = 1 when enabled. SPEED_CAP[5] = 0 when disabled.
  - This setting is disabled in the reference design.
- X10 (250 MHz)
  - SPEED_CAP[6] = 1 when enabled. SPEED_CAP[6] = 0 when disabled.
  - This setting is disabled in the reference design.
- X12 (300 MHz)
  - SPEED_CAP[7] = 1 when enabled. SPEED_CAP[7] = 0 when disabled.
  - This setting is disabled in the reference design.
- X16 (400 MHz)
  - SPEED_CAP[7] = 1 when enabled. SPEED_CAP[7] = 0 when disabled.
  - This setting is disabled in the reference design.
- X24 (600 MHz)
  - SPEED_CAP[9] = 1 when enabled. SPEED_CAP[9] = 0 when disabled.
  - This setting is disabled in the reference design.
  - This is not accessible for MachXO3 and MachXO3D (grayed out).
- Target System Clock (MHz)
  - This option is non-editable.
  - This is computed to be the system clock (clk_i of LTPI) required, based on the fastest frequency enabled.
  - Changes to this frequency are managed by the reference design logic.

Feature Capability Section

- Enable Full OEM Capabilities Type
  - This enables or disables the usage of the OEM Capability 0 in Hex (0x) and OEM Capability 1 in Hex (0x) options under the OEM Tab.
  - By default, this option is non-editable.
  - This option is unused in the reference design.
- Default Feature Capability 0 in Hex (0x)
  - By default, this option is non-editable.
  - This option represents the lower 32 bits of the default feature capabilities.
  - This value is computed based on the current settings of the IP. It is written to offset 0x10 of the CSR (FEATURE_CAP3_0) when the Enable Full OEM Capabilities Type is disabled.
  - This value should be consistent between SCM and HPM.
- Default Feature Capability 1 in Hex (0x)
  - By default, this option is non-editable.
  - This option represents the upper 32 bits of the default feature capabilities.
  - This value is computed based on the current settings of the IP. It is written to offset 0x14 of the CSR (FEATURE_CAP7_4) when Enable Full OEM Capabilities Type is disabled.
  - This value should be consistent between SCM and HPM.
- Automatically move to Configuration State
  - This option is only available for SCM.
  - This option is enabled in the reference design.
  - This option should be enabled unless the user can trigger the transition from the Advertise state to the Configuration state via CSR access.

6.1.3.4. LL GPIO Tab

The options on this tab are only available if the Enable LL GPIO Channel is enabled.

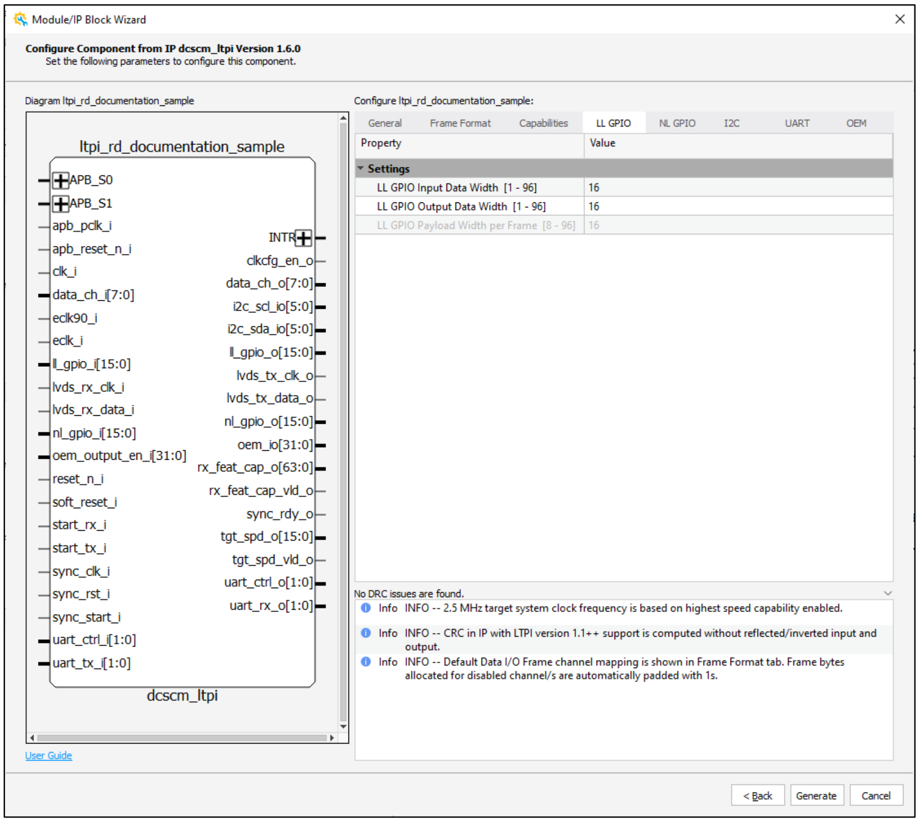

Figure 6.7. LL GPIO Tab

- LL GPIO Input Data Width
  - This option determines the size of the Low Latency Inputs.
  - In the reference design, this option is set to 4.
  - Although up to 96 bits can be set, this is only possible when the Data Frame Type is Custom. When the Data Frame Type is Default I/O, this can only be set up to 16 bits.
- LL GPIO Output Data Width
  - This option determines the size of the Low Latency Outputs.
  - In the reference design, this option is set to 4.
  - Although up to 96 bits can be set, this is only possible when the Data Frame Type is Custom. When the Data Frame Type is Default I/O, this can only be set up to 16 bits.
- LL GPIO Payload Width per Frame
  - This option determines the maximum number of LL GPIO that is sent per frame.
  - The value for either LL GPIO Input Data Width or LL GPIO Output Data Width must not exceed this value.
  - This can only be modified when the Data Frame Type is set to Custom and the Enable Data Channel option is disabled.

6.1.3.5. NL GPIO Tab

The options on this tab are only available if the Enable NL GPIO Channel is enabled.

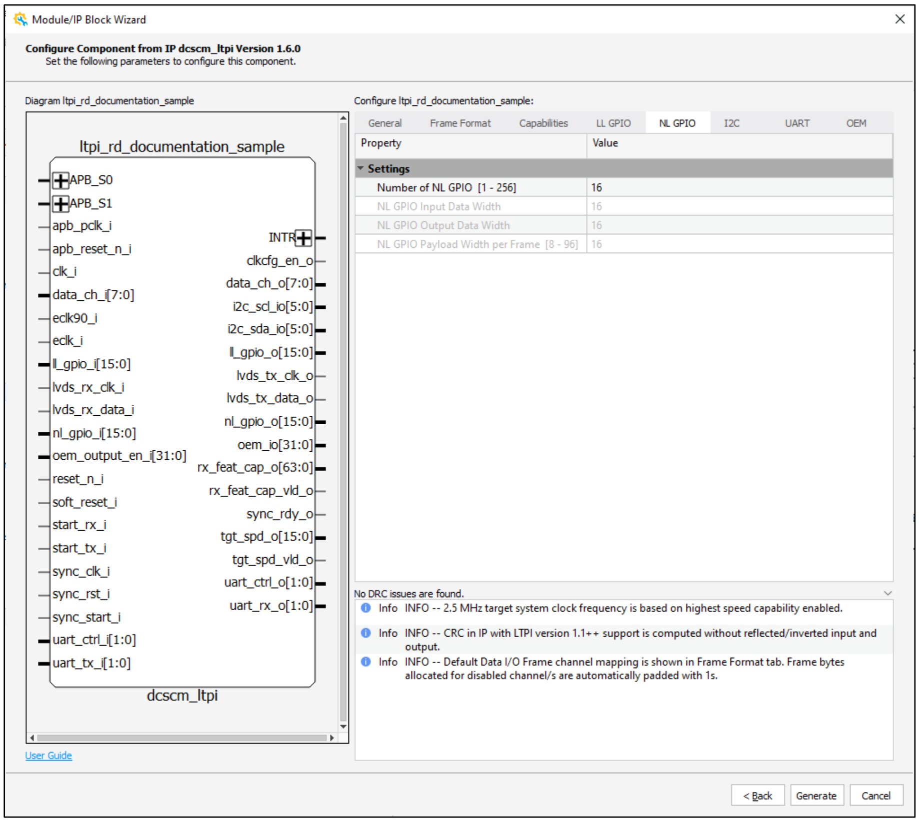

Figure 6.8. NL GPIO Tab

- Number of NL GPIO
  - This option determines the size of the Normal Latency Inputs and Outputs.
  - In this reference design, this option is set to 4.
  - Unlike LL GPIO, this option can be set up to its maximum size of 256 bits. This is because NL GPIO bits are sent through the frame based on the size of the NL GPIO Payload Width per Frame using a round-robin sequence.
  - Note that setting this option beyond the size of the NL GPIO Payload Width per Frame will cause updates on the I/O to be delayed.
- NL GPIO Payload Width per Frame
  - This option determines the maximum number of NL GPIO signal sent per frame.
  - Unlike other channels, the total number of bits that can be set is not influenced by this option.
  - This option can only be modified when Data Frame Type is set to Custom.

6.1.3.6. I2C Tab

The options on this tab are only available if the Enable I2C Channel is enabled.

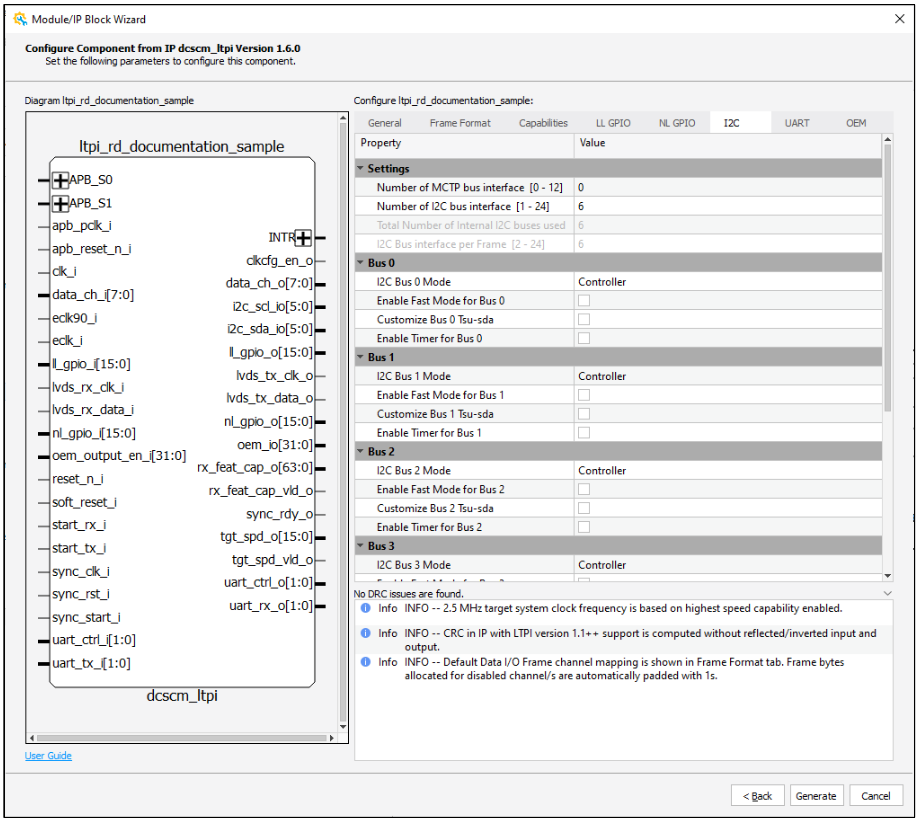

Figure 6.9. I2C Tab

Settings Section

- Number of MCTP bus interface
  - This option allows the IP to generate I2C channels that can support both external I2C controllers and I2C targets simultaneously.
  - Each channel uses two I2C bus instances.
- Number of I2C bus interface
  - This option determines the number of I2C buses enabled by the IP.
  - In the reference design, this option is set to 1.
  - Each number generates a new Bus X section in the GUI.
- I2C Bus Interface per Frame
  - This option determines the maximum number of I2C buses sent per frame. The default value is six (6).
  - This option can only be set in multiples of two (2) because each byte on the frame can store two (2) I2C bus data.
  - This option can only be modified when the Data Frame Type is set to Custom.

Bus X Section

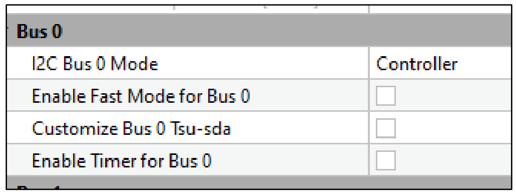

Figure 6.10. Bus X Section for Controller

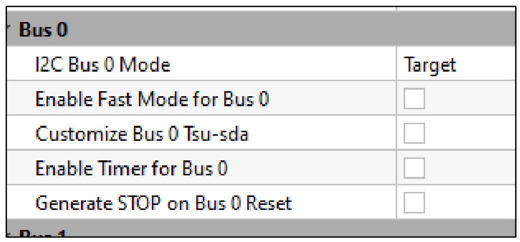

Figure 6.11. Bus X Section for Target

- I2C Bus X Mode
  - This option can be set to either Controller (expecting an external I2C controller to be connected) or Target (expecting an external I2C target to be connected).
  - This option is set to Controller for SCM and Target for HPM.
- Enable Fast Mode for Bus X
  - This option can be enabled to support fast mode I2C (400 kHz) or disabled to support standard mode I2C (100 kHz).
  - In the reference design, this option is disabled.
  - Note that this option affects how the IP regenerates the I2C transaction. While a fast mode enabled bus can support a 100 kHz transaction, it would regenerate 400 kHz on the Target side.
- Customize Bus Tsu-sda
  - This option enables or disables the customization of the SDA setup time.
  - In the reference design, this option is disabled.
- Generate STOP on Bus X Reset
  - This option is only available when the I2C Bus X Mode is set to Target.
  - This option enables or disables the IP to attempt sending a STOP condition on the bus when a I2C bus reset is invoked.
  - This option is not available if the Enable Timer for Bus X is enabled, which is the case in the reference design.
- Enable Timer for Bus X
  - This option enables or disables the timeout timer for the specific I2C bus.
  - In the reference design, this option is enabled.
  - When enabled, this option creates additional options within the section**.** Figure 6.12 shows the additional options.

Enable Timer for Bus 0	☑
Disable Timer Prescaler for Bus 0	☐
Prescaler Ratio for Bus 0	1:256
Timer Size of Bus 0 [1 - 32]	12
Apply Bus 0 Timer Settings to all I2C Buses	☐

Figure 6.12. Additional Bus X Section Options

Additional Bus X Section Options

- Disable Timer Prescaler for Bus X
  - This is disables/enables the use of the timer prescaler.
  - This is unchecked in the reference design.
- Prescaler Ratio for Bus X
  - This determines the timer prescaler value.
  - This is set to 1: 131072 in the reference design.
  - Use the formula in Figure 6.13 to determine the necessary setting for this option.
- Timer Size of Bus X
  - This determines the register size that would be allocated to the I2C timer.
  - This is set to 2 in the reference design.
  - Use the formula in Figure 6.13 to determine the necessary setting for this option.
- Apply Bus X Timer Settings to all Buses
  - This option enables/disables the same timer settings to be used for all other I2C buses.
  - This is only available if there is more than one (1) I2C bus enabled, which is not the case for the reference design.

Threshold Equation:

$$Timeout_{TH} = \text{Prescaler Ratio} \times \text{System Clock Period} \times [(2^{\text{Timer Size}})-1]$$

Sample Computation/s:

Prescaler Ratio = 1:65536
System Clock = 12.5 ns (80 MHz)
Timer Size = 5
$Timeout_{TH} = 65536 \times 12.5\ \text{ns} \times [(2^5)-1]$
= 25.3952 ms

Prescaler Ratio = DISABLED
System Clock = 12.5 ns (80 MHz)
Timer Size = 21
$Timeout_{TH} = 1 \times 12.5\ \text{ns} \times [(2^{21})-1]$
= 26.2144 ms

Figure 6.13. I2C Timeout Formula and Sample Computation

6.1.3.7. UART Tab

The options on this tab are only available if the Enable UART Channel is enabled.

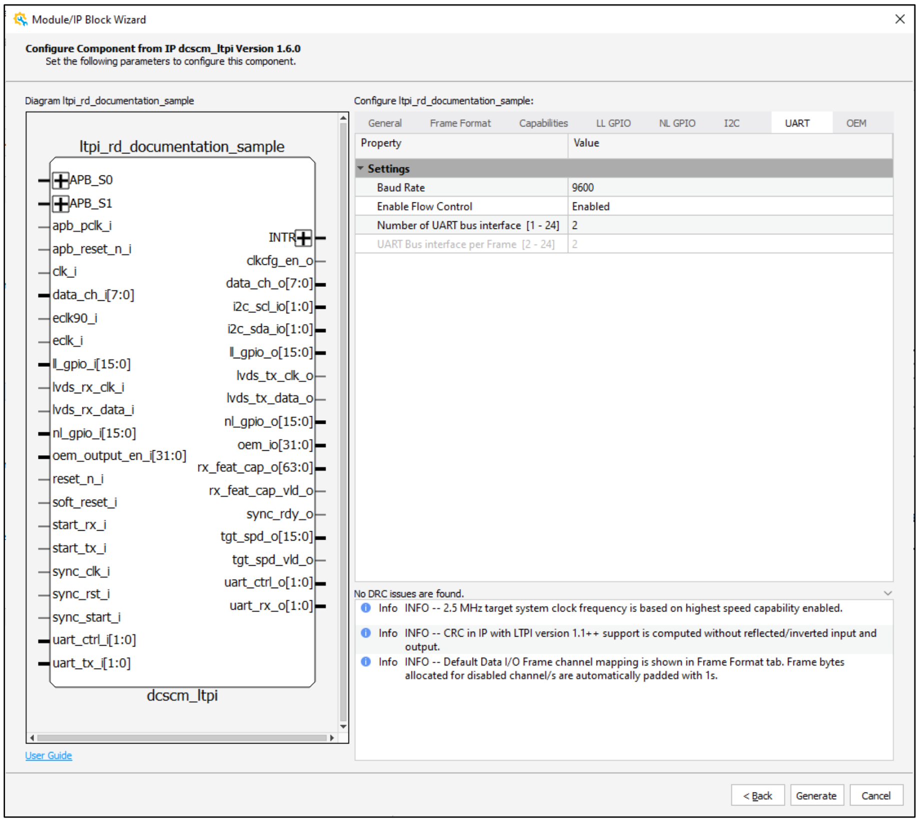

Figure 6.14. UART Tab

- Baud Rate
  - This option determines the maximum baud rate that can be supported.
  - In the reference design, this option is set to 115,200.
- Enable Flow Control
  - This option enables or disables the flow control signals, which are used to support multiple full-duplex UART interfaces tunneling.
  - In the reference design, this option is disabled.
- Number of UART bus interface
  - This option determines the number of UART buses are enabled by the IP.
  - In the reference design, this option is set to 1.
- I2C Bus Interface per Frame
  - This option determines the maximum number of UART buses sent per frame. The default value is two (2).
  - This option can only be set in multiples of two (2) because each byte on the frame can store two (2) I2C bus data.
  - This option can only be modified when the Data Frame Type is set to Custom.

6.1.3.8. OEM Tab

The options on this tab are only available if the Enable OEM Channel is enabled.

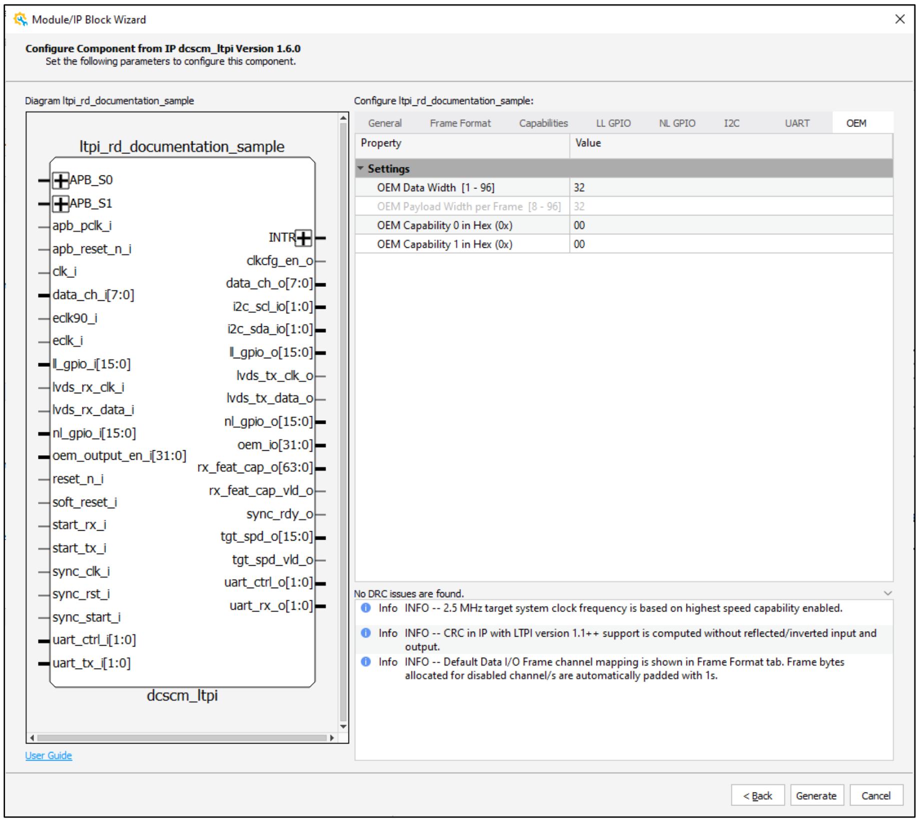

Figure 6.15. UART Tab

- OEM Data Width
  - This option determines the number of OEM bits enabled by the IP.
  - This option is not available in the reference design (Enable OEM Channel = Disabled).
- I2C Bus Interface per Frame
  - This option determines the maximum number of OEM bits that can be sent per frame. The default value is 32.
  - The value of the OEM Data Width cannot exceed this value.
  - This option can only be modified when the Data Frame Type is set to Custom.
  - This option is not available in the reference design (Enable OEM Channel = Disabled).
- OEM Capability 0 in Hex (0x)
  - This option represents the lower 32 bits of custom feature capabilities.
  - The value in this option is written to offset 0x10 of the CSR (FEATURE_CAP3_0) when the Enable Full OEM Capabilities Type is enabled.
  - This option is not available in the reference design (Enable OEM Channel = Disabled).
- OEM Capability 1 in Hex (0x)
  - This option represents the upper 32 bits of custom feature capabilities.
  - The value in this option is written to offset 0x14 of the CSR (FEATURE_CAP7_4) when Enable Full OEM Capabilities Type is enabled.
  - This option is not available in the reference design (Enable OEM Channel = Disabled).

6.1.4. 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 will 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.

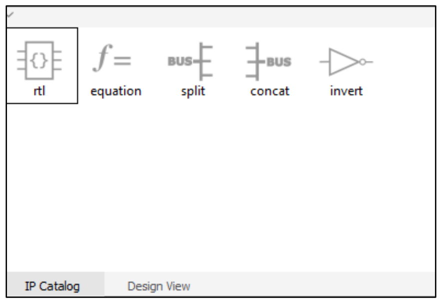

Figure 6.16. RTL Glue Logic

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

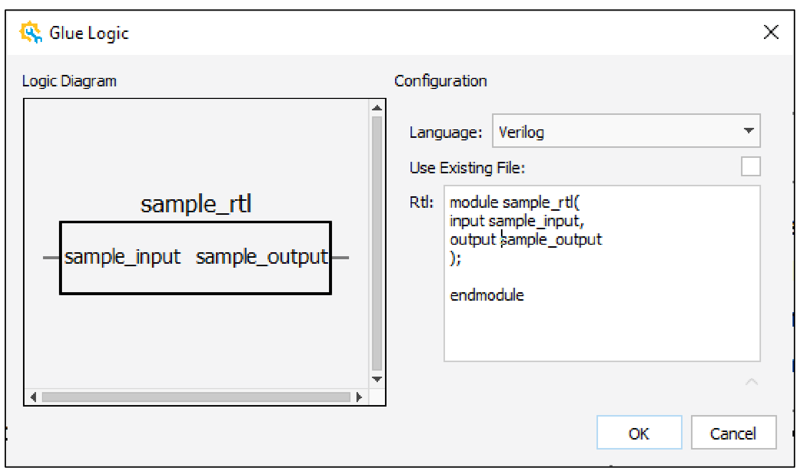

Figure 6.17. Using the RTL Box

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

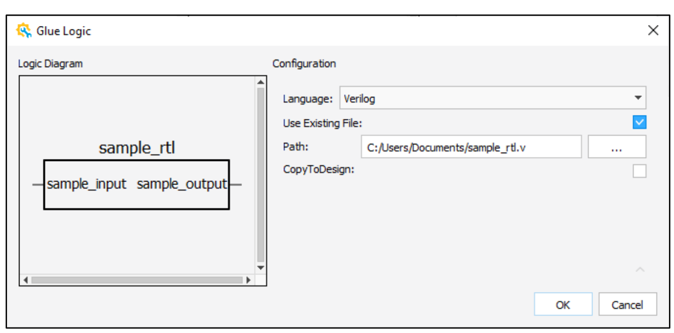

Figure 6.18. Using an Existing RTL File

4. The user 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.

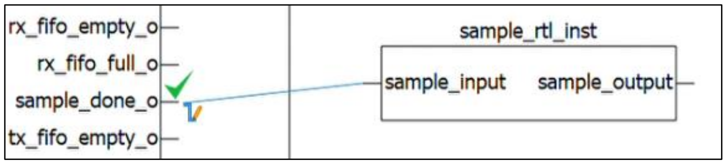

Figure 6.19. Connecting to Existing Modules

5. Click **Save**.

6.1.5. Removing the 25 MHz Generator

For this example, the 25 MHz generator for MachXO3/MachXO3D will be used, but this can also be applied to the MachXO5-NX.

1. Click on the nets connecting the clk_i external port to the clk_i port of the 25 MHz Generator and rstn_i external port to the rstn_i port of the 25 MHz generator, and click **Delete**.

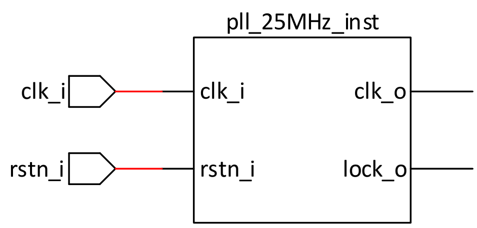

Figure 6.20. Selecting the Nets

2. Hover over the clk_o port until the pencil icon appears. Connect it to the clk_i external port (a green check mark should appear).

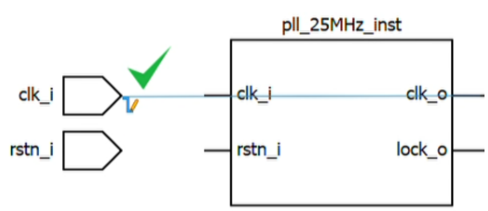

Figure 6.21. Connecting the Existing Nets of clk_o to External clk_i

3. Hover over the lock_o port until the pencil icon appears. Connect it to the rstn_i external port (a green check mark should appear).

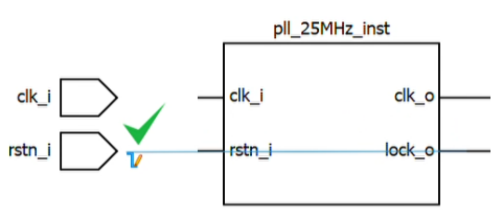

Figure 6.22. Connecting the Existing Net of lock_o to External rstn_i

4. If successful, the nets should be transferred to the external ports.

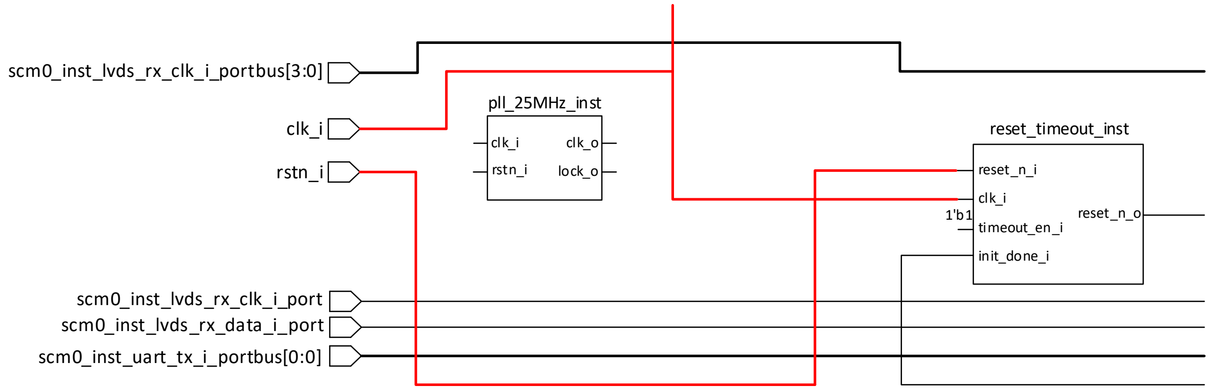

Figure 6.23. Transferred Connections

5. Delete the 25 MHz generator module.
6. Click **Save**.

6.2. 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 3.14 User Guide.

6.2.1. 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.

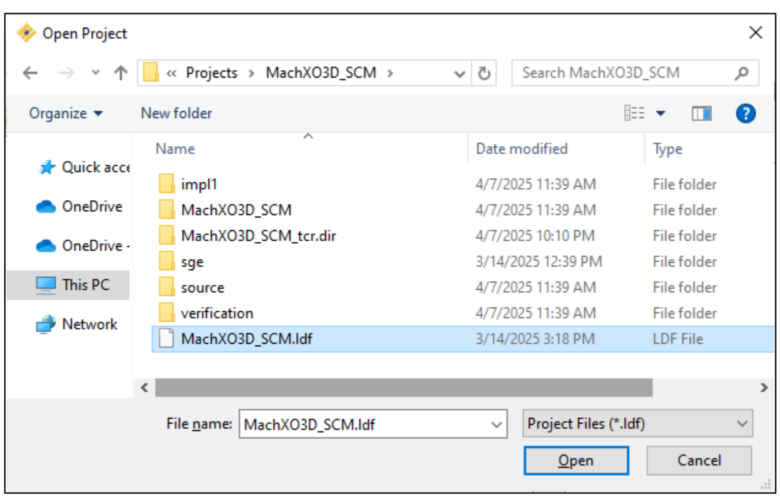

Figure 6.24. Selecting the Lattice Diamond Project

6.2.2. 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.

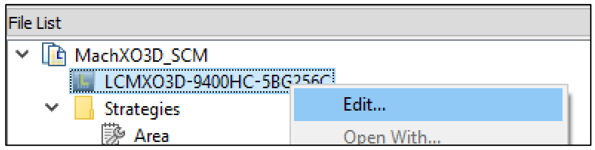

Figure 6.25. Selecting the Lattice Diamond Project

2. Choose the new target device.

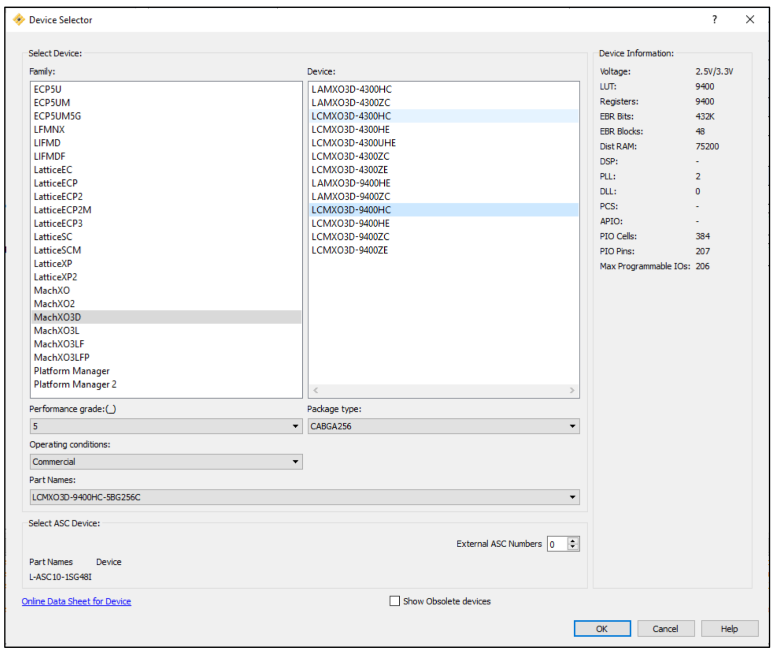

Figure 6.26. Device Selector Window

6.2.3. 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,

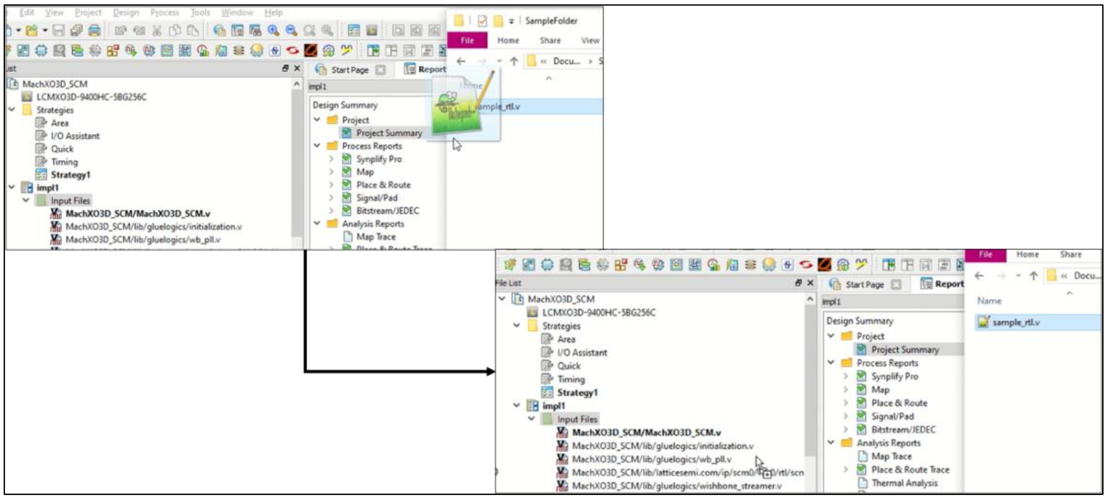

Figure 6.27. Adding a Module via Drag and Drop

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

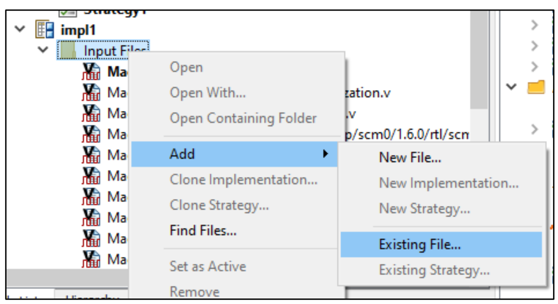

Figure 6.28. Selecting the Lattice Diamond Project

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

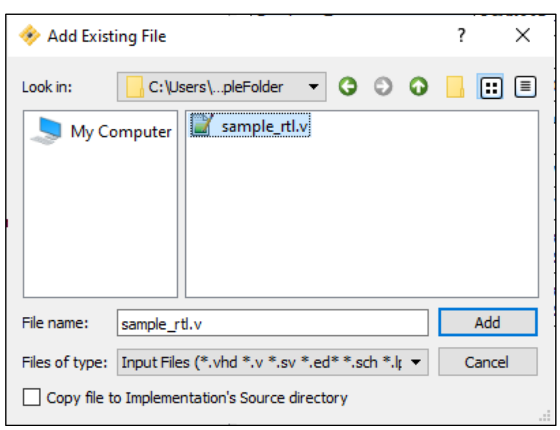

Figure 6.29. Add Existing File Window

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.

6.2.4. Changing the 25 MHz Generator Input

The default requirement for the 25 MHz generator is 12 MHz. These are the steps to change the input frequency required:

1. Double-click the upll.ipx file.

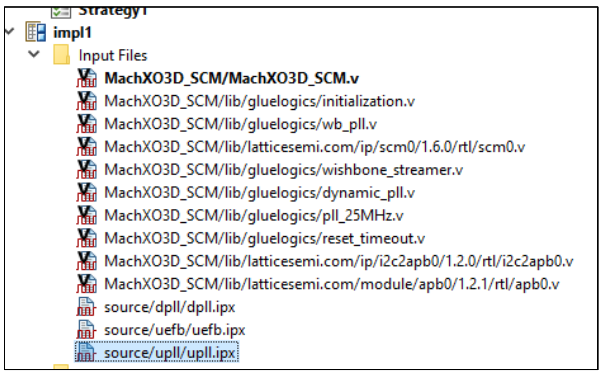

Figure 6.30. UPLL File

2. Under the CLKI block, change the frequency value, then click **Calculate**. Note that if the input frequency is already 25 MHz, it might not be necessary to use this PLL instance. See Removing the 25 MHz Generator section for instructions on how to remove this module.

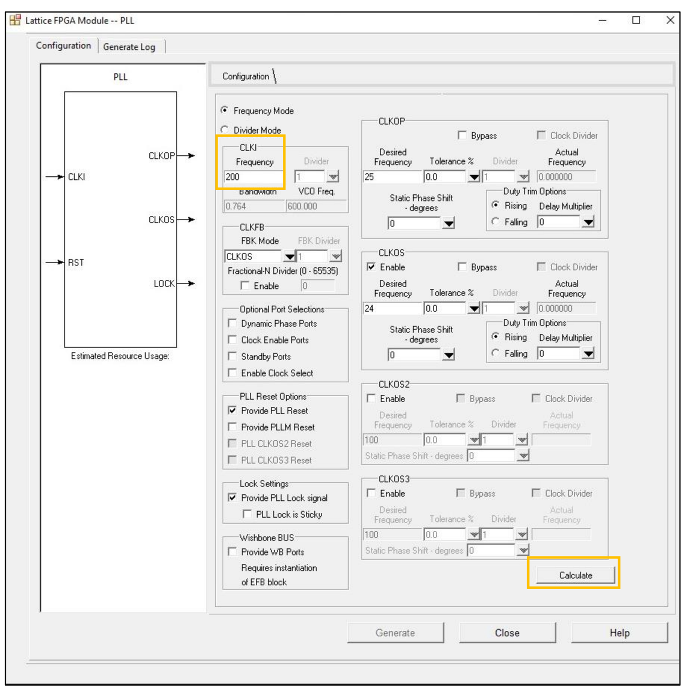

Figure 6.31. PLL GUI Window

3. If an error occurs, check if setting the CLKOS to a factor or multiple of the input clock can work. This is usually due to the PLL not being able to calculate the divider values that will allow it to generate the CLKOP or CLKOS. In the case of the reference design, directly using CLKOP as the feedback clock will not work because the PLL cannot calculate a 25 MHz clock based on a 12 MHz input. This is why it uses CLKOS as the feedback clock, which is set to 24 MHz. For more information on how to calculate the PLL frequencies, refer to the MachXO3 sysCLOCK PLL Design and User Guide (FPGA-TN-02058) or the MachXO3D sysCLOCK PLL Usage Guide (FPGA-TN-02070). In Figure 6.32, an input clock of 35 MHz cannot be used to calculate CLKOP based on the current PLL settings. In Figure 6.33, setting CLKOS (feedback clock) to 70 MHz (a multiple of 35) allowed the IP to calculate the dividers to generate CLKOP.

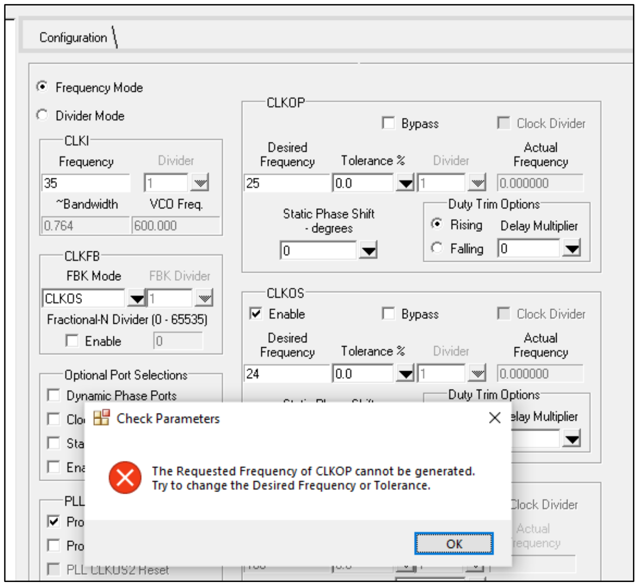

Figure 6.32. PLL Clock Calculation Error

4. Once successfully calculated (indicated by the availability of the Generate button), click **Generate**, then **Close**.

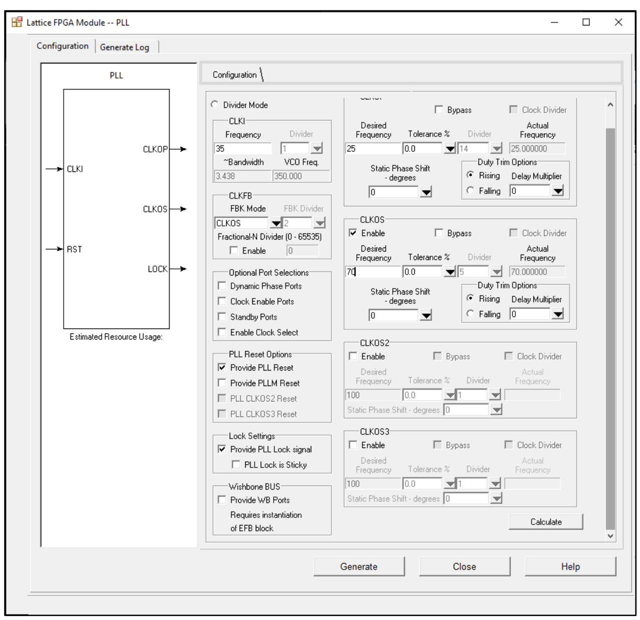

Figure 6.33. PLL Clock Calculation Done

6.2.5. Updating the Pin Assignments

There are two ways to change the pin assignment:

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

6.2.5.1. Updating Using Constraint File

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

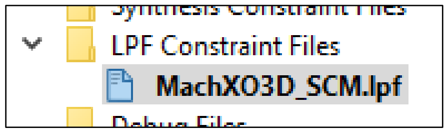

Figure 6.34. Constraint File (*.lpf)

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

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

Figure 6.35. IO 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 6.36. IO Location Example

4. Save the file.

6.2.5.2. Updating Using Spreadsheet View

1. Run synthesis.

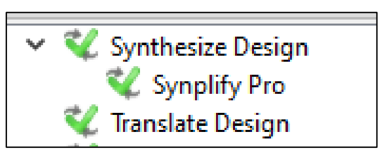

Figure 6.37. Lattice Diamond Synthesis

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

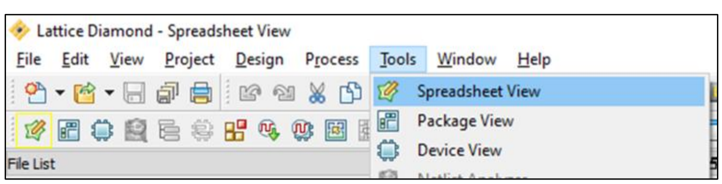

Figure 6.38. Open Spreadsheet View

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

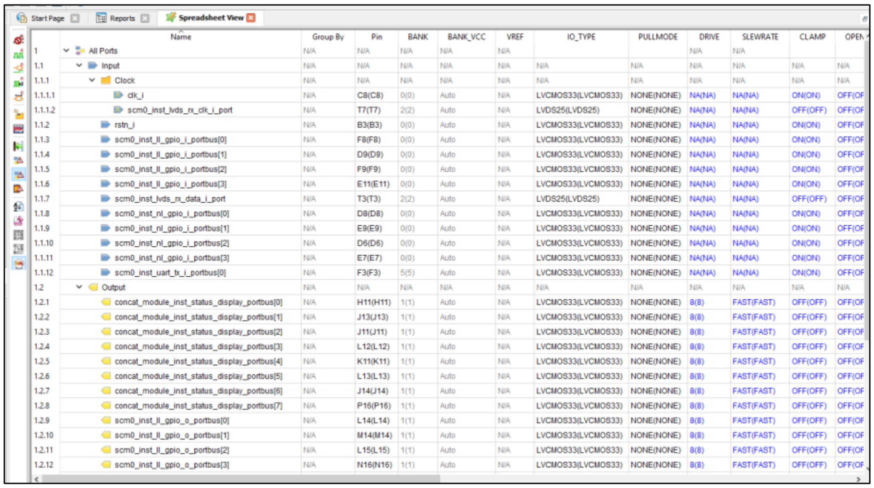

Figure 6.39. Spreadsheet View

4. Save the changes.

6.3. 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 2024.2 User Guide.

6.3.1. 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**.

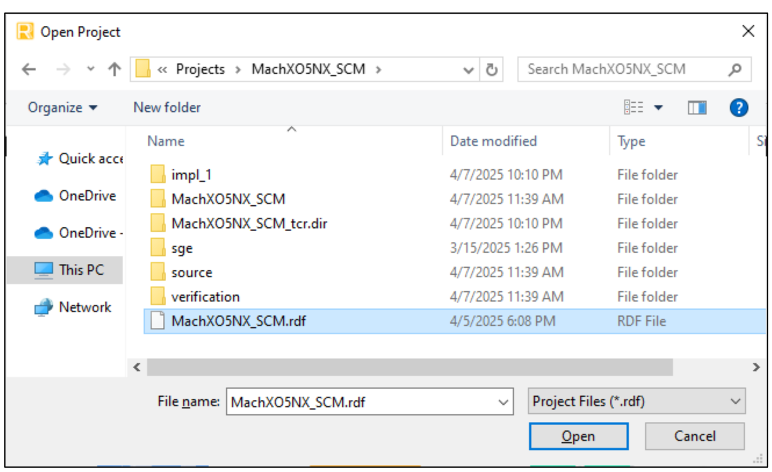

Figure 6.40. Selecting the Lattice Radiant Project

6.3.2. 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,

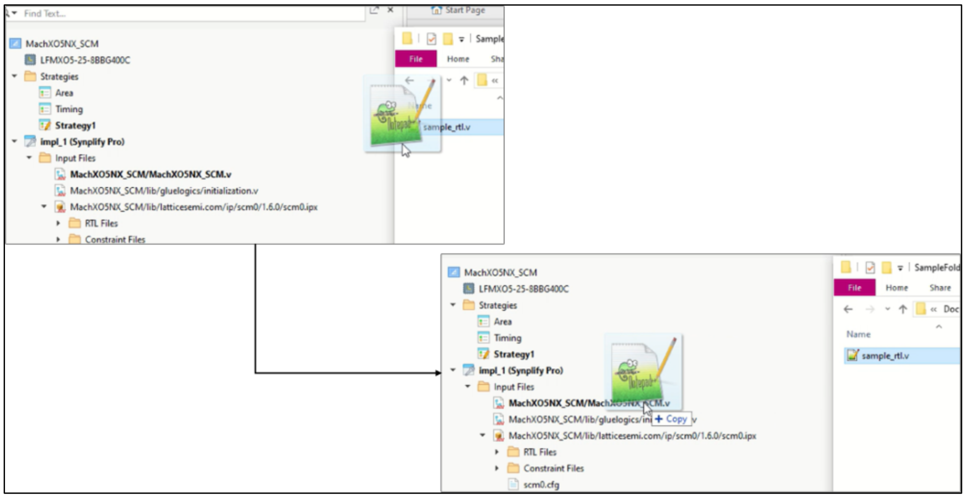

Figure 6.41. Adding a Module via Drag and Drop

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

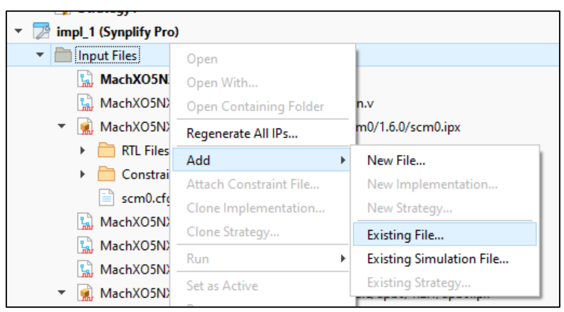

Figure 6.42. Selecting the Lattice Diamond™ Project

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

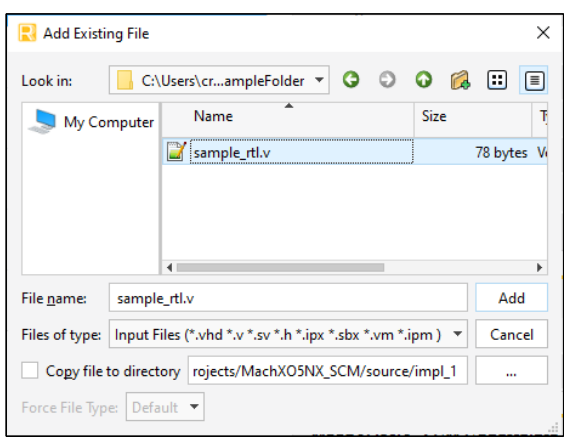

Figure 6.43. Add Existing File Window

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.

6.3.3. Changing the 25 MHz Generator Input

The default requirement for the 25 MHz generator is 125 MHz. These are the steps to change the input frequency required.

1. Double-click the upll_xo5.ipx file.

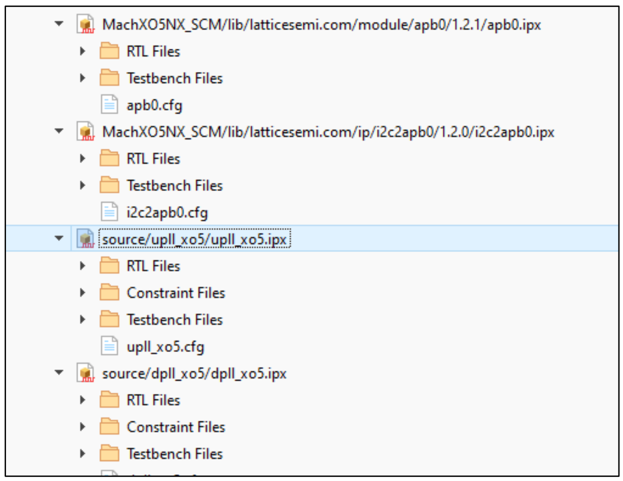

Figure 6.44. UPLL File

2. Under the Reference Clock section, change the CLKI frequency value then click **Calculate**. Note that if the input frequency is already 25 MHz, it might not be necessary to use this PLL instance. See the Removing the 25 MHz Generator section for instructions on how to remove this module.

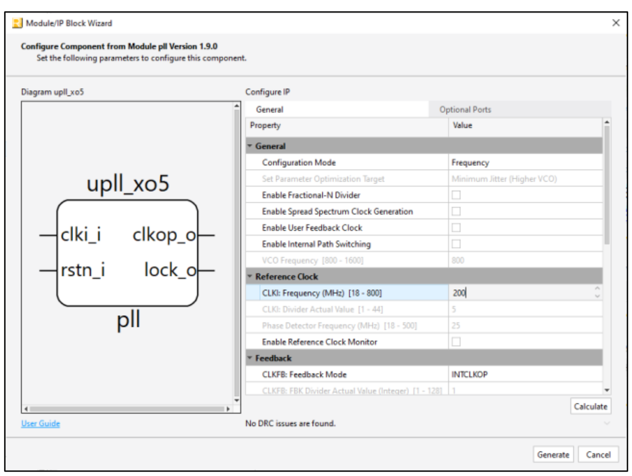

Figure 6.45. PLL GUI Window

3. Once successfully calculated (indicated by the message seen in Figure 6.46), click **Generate**, then **Finish**.

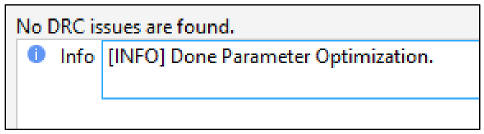

Figure 6.46. PLL Clock Calculation Done

6.3.4. Updating the Pin Assignments

There are two ways to change the pin assignment:

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

6.3.4.1. Updating Using Constraint File

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

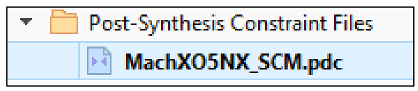

Figure 6.47. Lattice Radiant Constraint File (*.pdc)

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 6.48. Lattice Radiant IO 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 6.49. Lattice Radiant™ IO Location Example

4. Save the file.

6.3.4.2. Updating Using the Device Constraint Editor

1. Run synthesis.

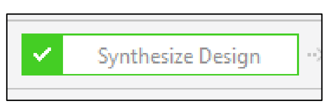

Figure 6.50. Lattice Radiant Synthesis

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

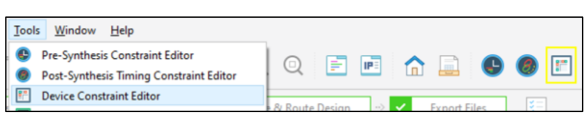

Figure 6.51. Open Device Constraint Editor

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

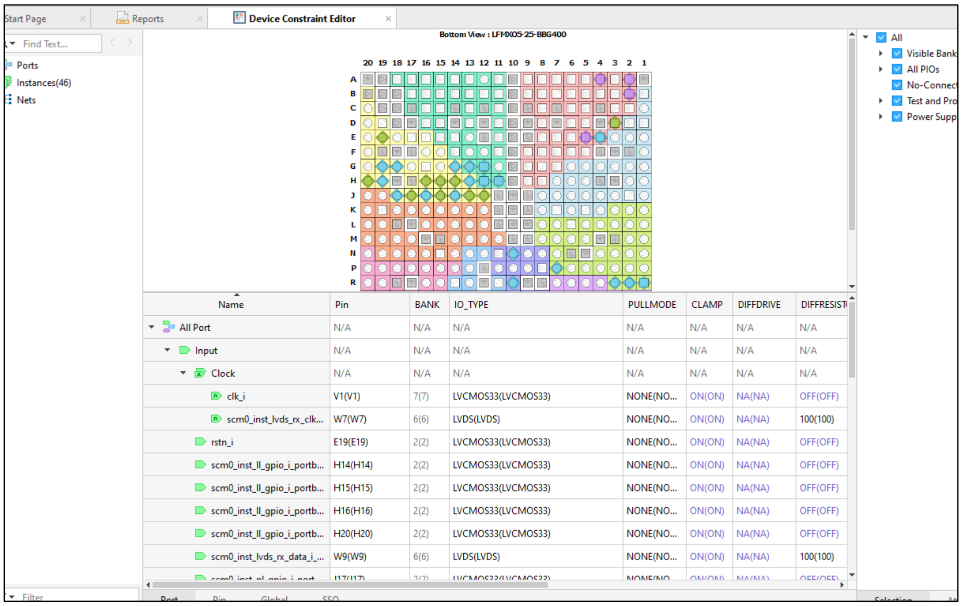

Figure 6.52. Device Constraint Editor

4. Save the changes.

7. Simulating the Reference Design

7.1. 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.

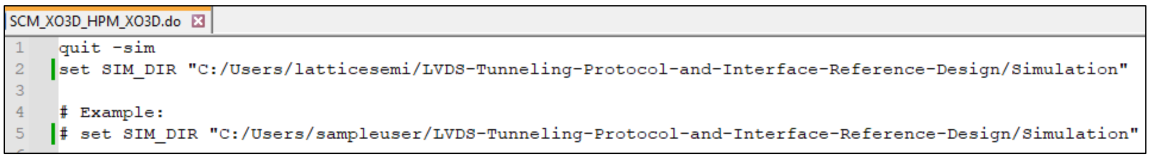

Figure 7.1. Simulation Do File

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** > **Tcl** > **Execute Macro**.

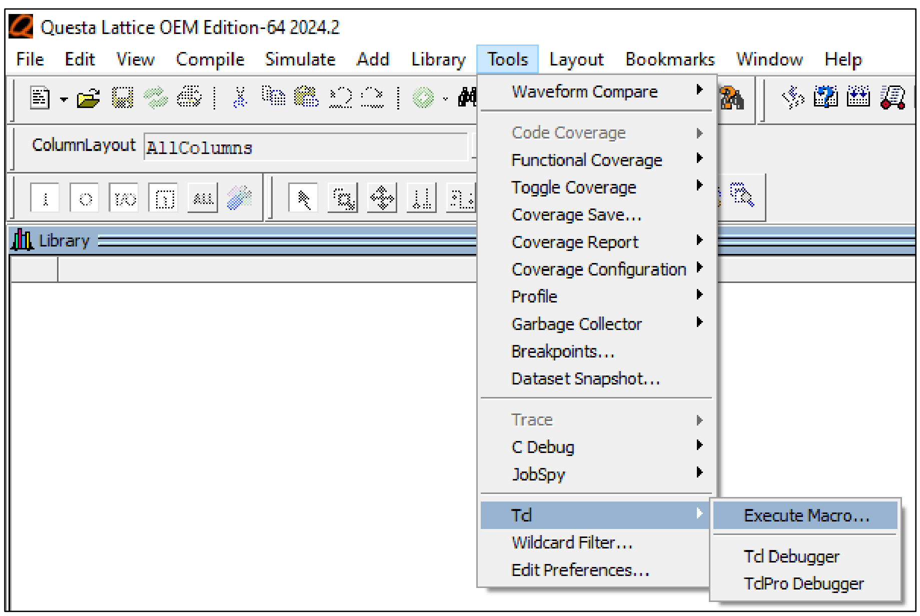

Figure 7.2. Execute Macro

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

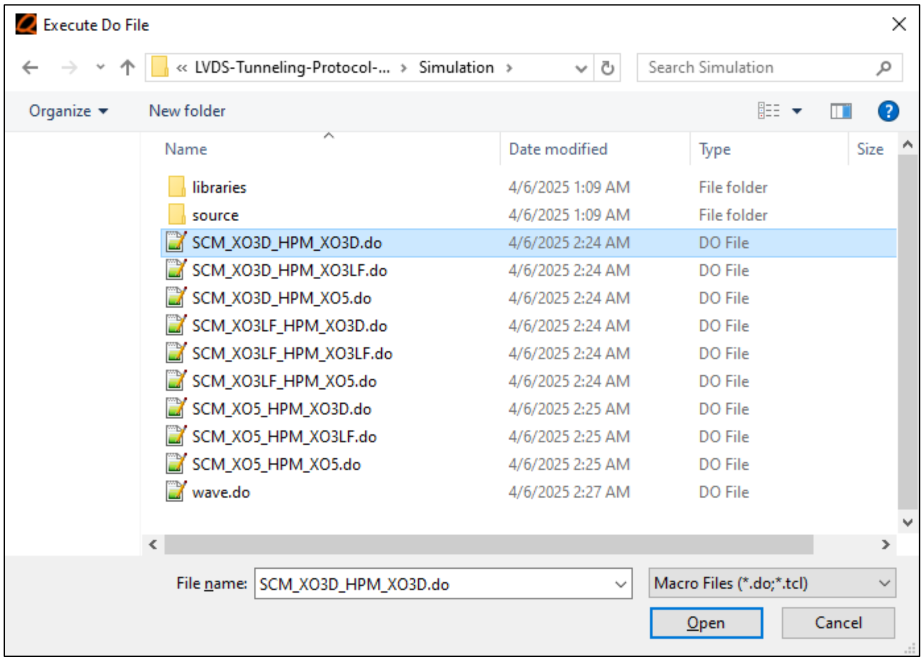

Figure 7.3. Execute Macro

7.2. Simulation Details and Results

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

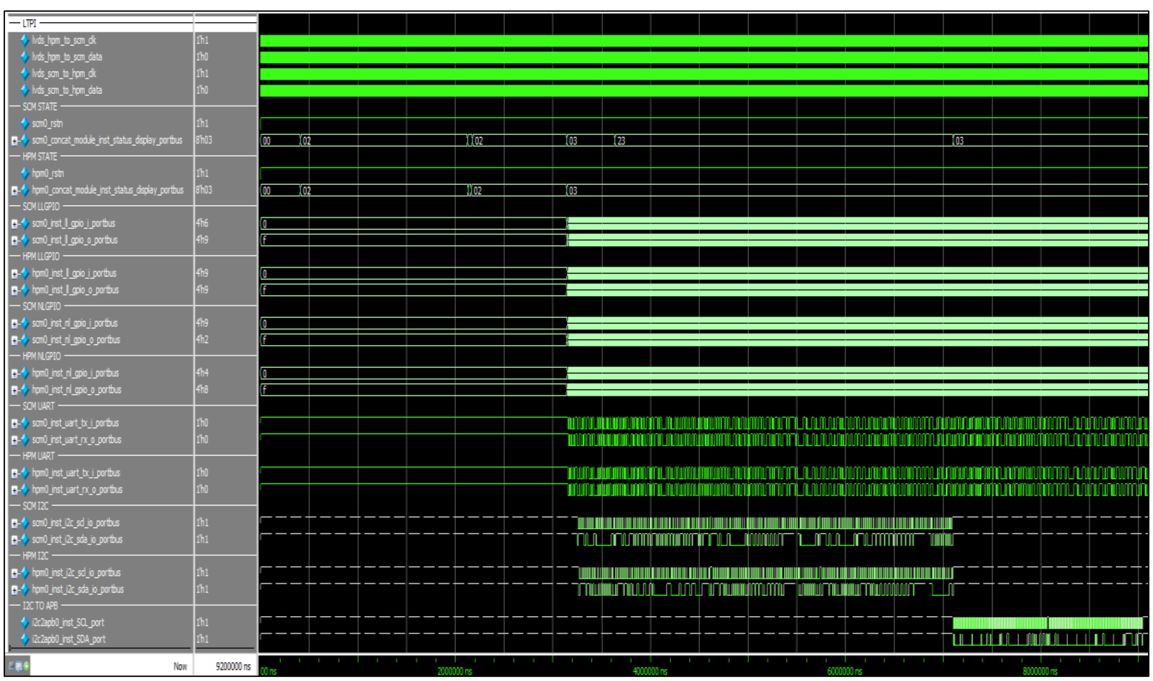

Figure 7.4. Simulation Wave View

7.2.1. Link Training

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

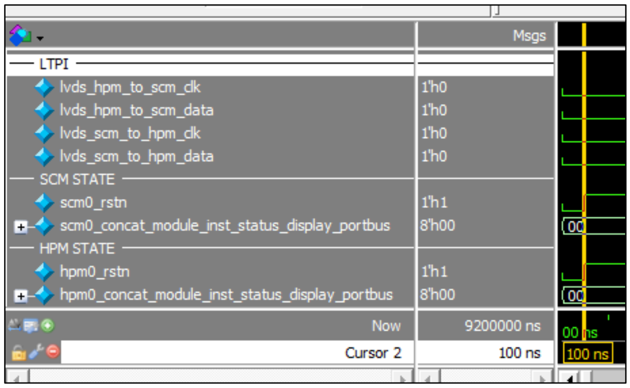

Figure 7.5. Reset Release

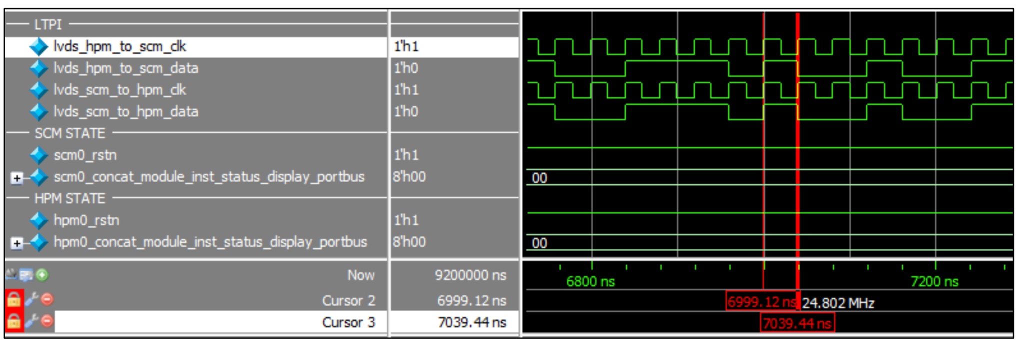

Figure 7.6. LTPI Base Speed

2. When status_display[1] = 0, it indicates that the PHY is not yet aligned.
3. Once aligned (status_display[1] = 1), the system enters either the Link Detect state or the Link Speed state. The transitions between these states are not explicitly shown in the simulation and are only observable via CSR access.

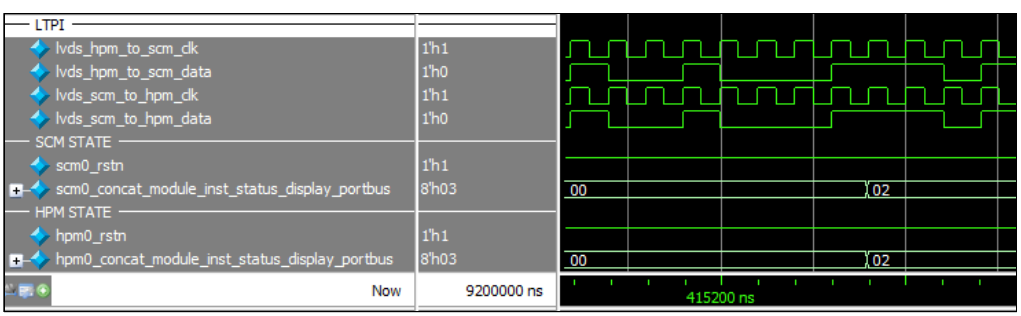

Figure 7.7. PHY Aligned

4. The transition between Base Speed and Target Speed is also observable in the simulation. This occurs when both PHY alignments of SCM and HPM (status_display[1]) go low without encountering a Link Error (status_display[7]). The default Target Speed of the reference design is 100 MHz DDR.

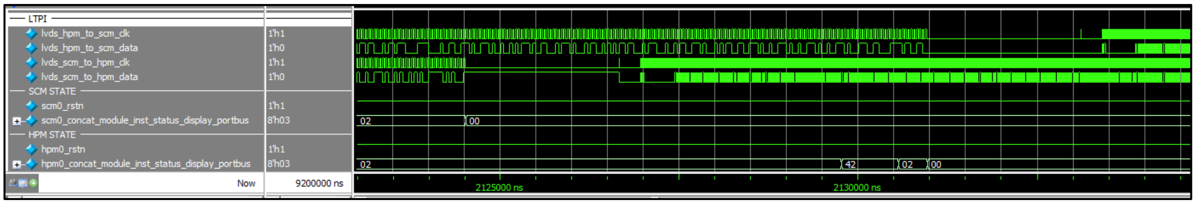

Figure 7.8. Base Speed to Target Speed Transition

5. Both SCM and HPM will realign their PHY using the new clock frequency and will go through Advertise, Configuration (SCM only), Accept (HPM only) and Active states.
6. A message will be printed on the simulator console once the link is established (state_display[0] = 1)

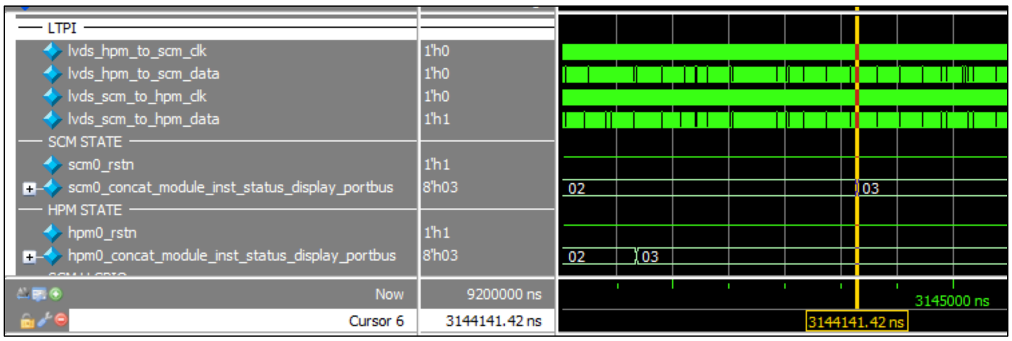

Figure 7.9. Link Training Done

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

Figure 7.10. Link Established Message

7.2.2. 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 0x9.
2. The test bench checks for changes in the Low Latency Output of HPM. If the change matches the set value of 0x9, a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.
3. The Low Latency Input of HPM is set to 0x6.
4. The test bench checks for changes in the Low Latency Output of SCM. If the change matches the set value of 0x6, a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.
5. The Normal Latency Input of SCM is set to 0xA.
6. The test bench checks for changes in the Normal Latency Output of HPM. If the change matches the set value of 0xA, a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.
7. The Normal Latency Input of HPM is set to 0x5.
8. The test bench checks for changes in the Normal Latency Output of SCM. If the change matches the set value of 0x5, a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.

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

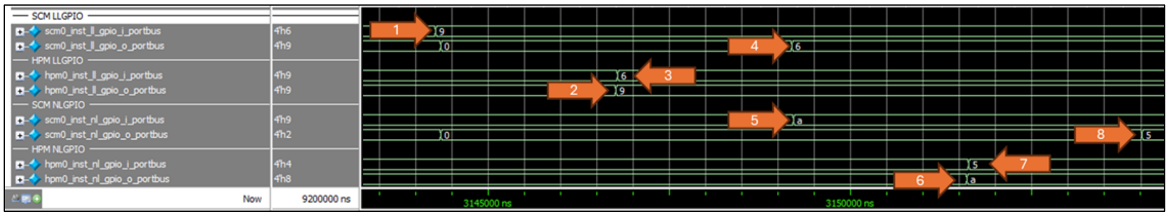

Figure 7.11. GPIO Test Flow

```
2905305ns   : <<<<<GPIO Test - START>>>>>


2905315ns   : SCM to HPM LLGPIO Test - START
2905320ns   : Setting SCM LLGPIO Input to 0x9
2909434ns   : SCM to HPM LLGPIO Test - Passed
2909439ns   : HPM LLGPIO Output received 0x9
2909439ns   : SCM to HPM LLGPIO Test - END


2909449ns   : HPM to SCM LLGPIO Test - START
2909454ns   : Setting SCM LLGPIO Input to 0x6
2911795ns   : HPM to SCM LLGPIO Test - Passed
2911800ns   : SCM LLGPIO Output received 0x6
2911800ns   : HPM to SCM LLGPIO Test - END


2911810ns   : SCM to HPM NLGPIO Test - START
2911815ns   : Setting SCM NLGPIO Input to 0xA
2915885ns   : SCM to HPM NLGPIO Test - Passed
2915890ns   : HPM NLGPIO Output received 0xA
2915890ns   : SCM to HPM NLGPIO Test - END


2915900ns   : HPM to SCM NLGPIO Test - START
2915905ns   : Setting SCM NLGPIO Input to 0x5
2918995ns   : HPM to SCM NLGPIO Test - Passed
2919000ns   : SCM NLGPIO Output received 0x5
2919000ns   : HPM to SCM NLGPIO Test - END


2919010ns   : <<<<<GPIO Test - END>>>>>
```

Figure 7.12. GPIO Test Console Messages

7.2.3. UART Test

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

1. Fixed data (0xA5) is sent through the SCM's Tx port.
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 SCM's Rx port. If the received data matches the transmitted data, a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.

After step 3, a message will indicate that the UART test is complete. The test bench will continue to stimulate the SCM's Tx port by increments of one. However, the SCM's Rx port will no longer be checked by the test bench. This continuous activity helps simulate an active UART channel while the next channel is tested.

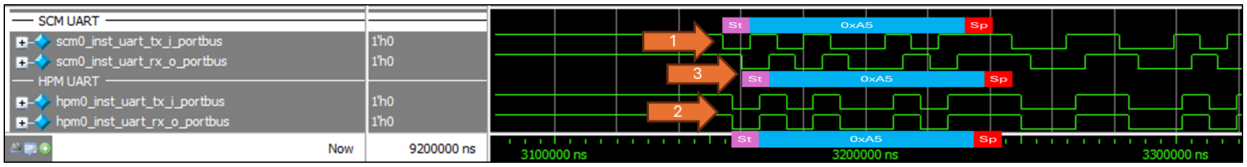

Figure 7.13. UART Test Flow

```
2919020ns  : <<<<<UART Test - START>>>>>


2919030ns  : Sending 0xA5
3014030ns  : UART Test - Passed
3014035ns  : SCM UART RX received 0xA5


3014045ns  : <<<<<UART Test - END>>>>>
```

Figure 7.14. UART Test Console Messages

7.2.4. 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 via 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 only performed 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 (0xAA55FF00).
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 will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.

After step 3, a message will indicate that the I2C test is complete. The I2C channels will no longer be stimulated by the test bench and will remain in a pull-up state.

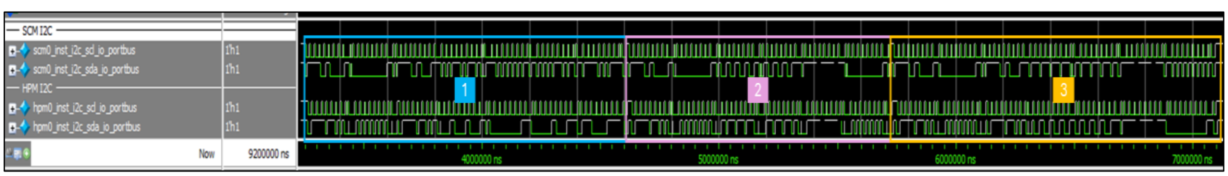

Figure 7.15. I2C Test Flow

```
3014055ns  : <<<<<I2C Test - START>>>>>


3014065ns  : Initial 32-bit Data Read from Address 0x00
4491895ns  : Received Data is 0x24810963
4491900ns  : Write Data 0xAA55FF00 to Address 0x00
5716145ns  : Read Data from Address 0x00
7199095ns  : I2C Test - Passed
7199100ns  : Received Data is 0xAA55FF00


7199110ns  : <<<<<I2C Test - END>>>>>
```

Figure 7.16. I2C Test Console Messages

7.2.5. 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 0x00000400, effectively reading offset 0x00 (ID) of SCM CSR (mapped to 0x400-0x7FF by default).
2. If the read data matches the expected SCM ID (0x0000CA26), a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.
3. The external controller sends a 32-bit read command to address 0x00000000, effectively reading offset 0x00 (ID) of HPM CSR (mapped to 0x000-0x3FF by default).
4. If the read data matches the expected HPM ID (0x0000BF17), a message will be printed, indicating that the test passed. Otherwise, the message will indicate that the test failed.

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

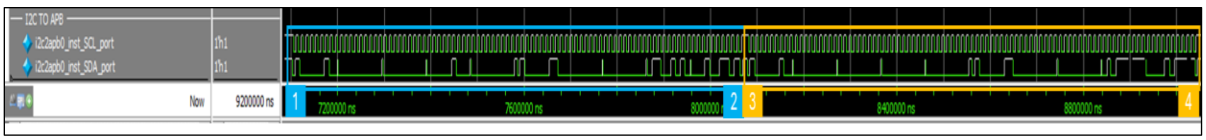

Figure 7.17. CSR Test Flow

```
7199120ns  : <<<<<CSR Access via I2C Test - START>>>>>


7199130ns  : SCM CSR Test - START
7199135ns  : Read Unique ID for SCM (Offset = 0x00000400), expecting 0x0000CA26
8168555ns  : SCM CSR Test - Passed
8168560ns  : Received Data is 0x0000CA26
8168565ns  : SCM CSR Test - END
8168570ns  : HPM CSR Test - START
8168575ns  : Read Unique ID for HPM (Offset = 0x00000000), expecting 0x0000BF17
9144555ns  : HPM CSR Test - Passed
9144560ns  : Received Data is 0x0000BF17
9144565ns  : HPM CSR Test - END


9144575ns  : <<<<<CSR Access via I2C Test - END>>>>>
```

Figure 7.18. CSR Test Console Messages

8. Hardware Implementation

Table 8.1 lists the pin assignments for the target devices. The port names are shortened for brevity.

The assignments are based on the boards used for each device during design evaluation. Table 8.2 lists the boards used for each device.

Connect the LVDS Tx ports of SCM to the LVDS Rx ports of HPM, and vice versa.

A successful link will be indicated by a logic 1 on state_display[0] (Init Done).

Table 8.1. Pin Assignments

Port Name	MachXO3LF/MachXO3D	MachXO5-NX	Notes
LVDS Interface (SCM)			
clk_i	C8	V1	External on-board oscillator
rstn_i	B3	E19	Pushbutton
LVDS Interface (SCM)			
lvds_tx_clk_o_port	B9	R10	—
lvds_tx_data_o_port	A3	N10	—
lvds_rx_clk_i_port	T7	W7	—
lvds_rx_data_i_port	T3	W9	—
LVDS Interface (HPM)			
lvds_tx_clk_o_port	D10	Y12	—
lvds_tx_data_o_port	A5	N11	—
lvds_rx_clk_i_port	T9	Y11	—
lvds_rx_data_i_port	T11	Y14	—
Low Latency GPIO Interface			
ll_gpio_i_portbus[0]	F8	H14	—
ll_gpio_i_portbus[1]	D9	H15	—
ll_gpio_i_portbus[2]	F9	H16	—
ll_gpio_i_portbus[3]	E11	H20	—
ll_gpio_o_portbus[0]	L14	G14	—
ll_gpio_o_portbus[1]	M14	G18	—
ll_gpio_o_portbus[2]	L15	G19	—
ll_gpio_o_portbus[3]	N16	H19	—
Normal Latency GPIO Interface			
nl_gpio_i_portbus[0]	D8	J17	—
nl_gpio_i_portbus[1]	E9	J15	—
nl_gpio_i_portbus[2]	D6	J13	—
nl_gpio_i_portbus[3]	E7	J12	—
nl_gpio_o_portbus[0]	L16	J18	—
nl_gpio_o_portbus[1]	M15	J16	—
nl_gpio_o_portbus[2]	N14	J14	—
nl_gpio_o_portbus[3]	M16	H13	—
UART Interface			
uart_tx_i_portbus	F3	D3	—
uart_rx_o_portbus	D1	E4	—
I2C Interface			
i2c_scl_io_portbus	D3	A4	Requires External Pull-up
i2c_sda_io_portbus	C2	E5	Requires External Pull-up

Port Name	MachXO3LF/MachXO3D	MachXO5-NX	Notes
I2C-to-APB Interface			
i2c2apb0_inst_SCL_port	E3	B2	Requires External Pull-up (SCM Only)
i2c2apb0_inst_SDA_port	E1	A2	Requires External Pull-up (SCM Only)
Protocol Information			
status_display_portbus[0]	H11	R3	Init Done (LED)
status_display_portbus[1]	J13	R2	Rx Aligned (LED)
status_display_portbus[2]	J11	R1	Link Speed Timeout (LED)
status_display_portbus[3]	L12	P7	Configuration Timeout (LED)
status_display_portbus[4]	K11	H12	I2C Timer Timeout (LED)
status_display_portbus[5]	L13	H11	I2C Code Error (LED)
status_display_portbus[6]	N15	G13	CRC Error (LED)
status_display_portbus[7]	P16	G12	Link Error (LED)

Table 8.2. LTPI Clock Source Implementation

Target Device	Board Name	Ordering Part Number
MachXO3	MachXO3LF Starter Kit	LCMXO3LF-6900C-S-EVN
MachXO3D	MachXO3D Breakout Board	LCMXO3D-9400HC-B-EVN
MachXO5-NX	MachXO5-NX Development Board	LFMXO5-25-EVN

8.1. Requirements

8.1.1. DC-SCM 2.0 Requirements

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

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.

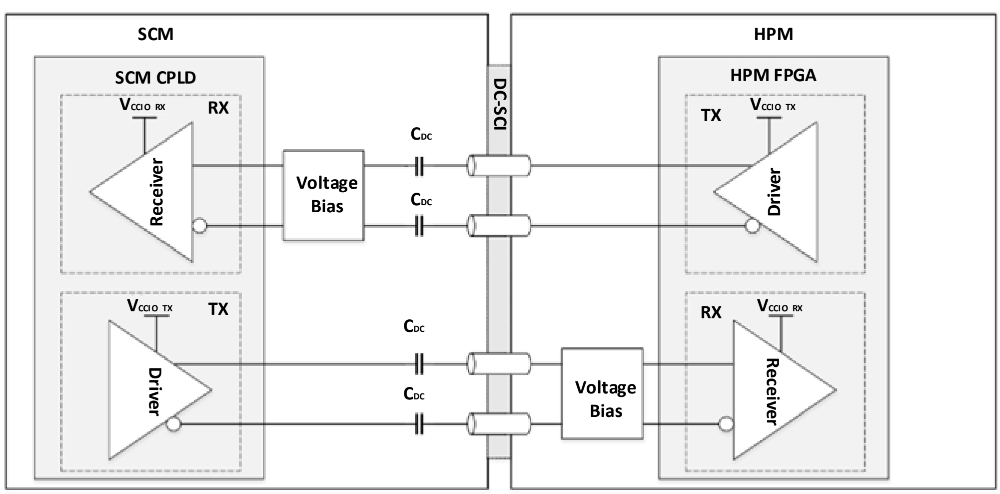

Figure 8.1. LVDS Link

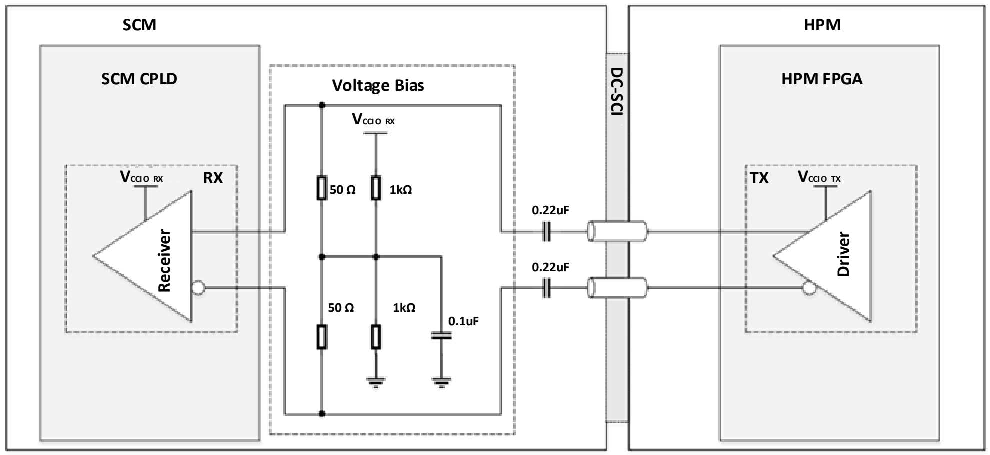

Figure 8.2. LVDS Link

Table 8.3. 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.
$V_{CCIO\ RX/TX}$	2.5 V	3.3 V	LVDS VCCIO Voltage (Refer to MachXO3/MachXO3D/Mach-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).

8.1.2. 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).

DISCLAIMER: This is not a comprehensive list of requirements for each device. Please refer to the supporting documents for each device for more information.

MachXO3 : https://www.latticesemi.com/Products/FPGAandCPLD/MachXO3

MachXO3D : https://www.latticesemi.com/Products/FPGAandCPLD/MachXO3D

MachXO5-NX : https://www.latticesemi.com/Products/FPGAandCPLD/MachXO5-NX

9. Resource Utilization

The resource utilization depends on the DC-SCM LTPI IP configuration. Table 9.1 shows the resource utilization of the reference design. Resource utilization may vary when using additional or fewer channel buses.

Table 9.1. Resource Utilization for Demo Designs

Device	Design	LUT4	Registers	EBR	SLICE
MachXO3LF	SCM	2,358	2,307	6	1,382
	HPM	2,180	2,074	6	1,301
MachXO3D	SCM	2,358	2,307	6	1,382
	HPM	2,180	2,074	6	1,301
MachXO5-NX	SCM	2,893	2,329	3	3,829
	HPM	2,660	2,091	3	3,422

10. Common Issues and Troubleshooting

10.1. 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 10.1).

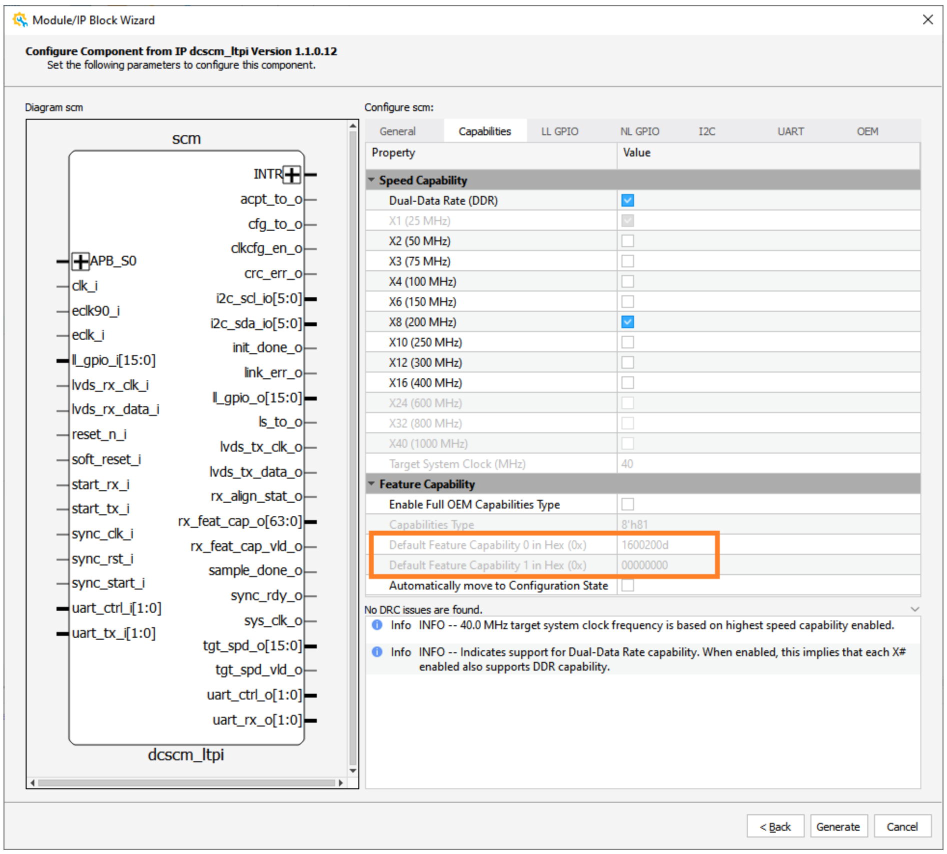

Figure 10.1. Feature Capabilities

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.
3. Clock Mismatch
   - This issue occurs when the external clock frequency does not match the expected clock input frequency of the 25 MHz generator. It can be resolved by updating upll.ipx on Lattice Diamond or upll_xo5.ipx on Lattice Radiant.

10.2. 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 10.2 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 10.2. 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 Specifications
- MachXO3 web page
- MachXO3D web page
- MachXO5-NX web page
- LVDS Tunneling Protocol and Interface Reference Design web page
- Lattice Propel 2024.2 Builder User Guide (FPGA-UG-02219)
- Lattice Propel Design Environment web page
- Lattice Radiant Software 2024.2 User Guide
- Lattice Radiant FPGA design software
- Lattice Diamond 3.14 User Guide
- Lattice Diamond FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

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

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

Revision History

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
	• 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) • 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.
Modifying the	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.

Section	Change Summary
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:
	  • 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.

Section	Change Summary
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.
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